Improving Natural Language Dataset Annotation Quality and Efficiency

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)

vorgelegt von

Jan-Christoph Klie

Tag der Einreichung:	1. Februar 2024
Tag der Disputation:	18. April 2024
Referentinnen:	Prof. Dr. Iryna Gurevych, Darmstadt, Germany Prof. Ph.D. Bonnie Webber, Edinburgh, United Kingdom

Darmstadt 2024

Improving Natural Language Dataset Annotation Quality and Efficiency

Genehmigte Dissertation von Jan-Christoph Klie

Tag der Einreichung: 1. Februar 2024
Tag der Disputation: 18. April 2024

Darmstadt, Technische Universität Darmstadt

Acknowledgements

My time as a doctoral student has been a life-changing experience for me, and I would like to take this opportunity to thank everyone who contributed to making it so memorable.

First of all, I want to thank my Master's Thesis supervisor Theresa Botschen née Martin who took me on despite my only limited NLP experience at that time. Second, I would like to thank Iryna Gurevych for giving me the opportunity to pursue a Ph.D. and for supporting me throughout this journey. Thank you also Richard for being my second supervisor who always helped me out with my many random questions and issues. Exchanging instant messages at night with you always was and still is quite delightful.

My sincere appreciation goes to the DFG for kindly financing my Ph.D. study via the INCEpTION project.

I'm very thankful to all my colleagues from the UKP Lab for the constructive and helpful feedback that I received during various talks and discussions. In particular, I would like to thank my fellow Ph.D. students and PostDocs (random order): Andreas Rücklé, Nils Reimers, Ilia Kuznetsov, Michael Bugert, Edwin Simpson, Gözde Gül Sahin, Benjamin Schiller, Nafise Sadat Moosavi, Max Glockner, Jonas Pfeiffer, Tilman Beck, Ji-Ung Lee, Yevgeniy Puzikov, Thomas Arnold, Ivan Habernal, Luke Bates, Irina Bigoulaeva, Jan Buchmann, Dennis Zyska, Nils Dycke, Max Eichler, Dominic Petrak, Qian Ruan, Sukannya Purkayastha, Yang Gao, Gisela Vallejo, Claudia Schulz, Tristan Miller. My biggest thanks go especially to Kevin Stowe who extensively helped me write my first long paper. Thank you Office Admin, especially Sue Messenger and Elisabeth Niemann for all the support, for the great conversations, and for helping with the bureaucracies of daily life. I also want to thank the students with whom I have had the privilege to work: Wei Ding, Peter Jiang, Lin Li, and Falko Helm. Thank you so much for the inspiring collaborations.

I had the honor to manage a distinguished Sysadmin team with many excellent members; without that second job, UKP and my Ph.D. would have been a tad boring. These are Fabian Herrlich, Sebastian Alles, Johannes Daxenberger, Erik-Lân Do Dinh, Michael Bugert, Naveen Kumar, Ute Winchenbach, Stephan Krämer and the new generation with Max Eichler, Dennis Zyska and Dominic Petrak.

I also want to thank Alex Yang Gao, who recommended me to chair the virtual website team for ACL and EMNLP online virtual conference. Only through this community service, I met Bonnie Webber, who gave me the idea for the annotation error detection paper, which is now a large contribution of this thesis. Thereby I also found a second reviewer of this very thesis. Thank you Bonnie for our fruitful collaborations!

The annotation error detection paper opened the door for me to an internship at Apple in Heidelberg. There I met many outstanding colleagues, especially Rahul, Juan, Lorenz and Marc and my fellow interns Kenneth and Aga. Thank you for the inspirational and supportive mentorship and for making this a wonderful experience.

Thank you for SIG Job that kept me sane during the pandemic and helping me with my work and life stuff; members are Michael Bugert, Ji-Ung Lee and Yevgeniy Puzikov. I hope we will never stop with our meetings.

I would like to thank my family and especially my parents, Karsten and Kirsten, sister Anne Katrin and grandparents Marga, Georg and Ilse for your love and support, and for supporting me even if we did not have much growing up. My achievements were only possible due to your sacrifices and you always being by my side.

Many thanks go to Dr. Haider, Frau Hechler, Frau Türke, Die Chirurgen and Frank who kept me healthy enough to get this thing done.

Thank you Zaijuan for supervising me during my time with you as a research assistant and often inviting me to your place for talks and foods. In Darmstadt, I met many friends as part of going to the Sprachencafé, who are, among others, Johann, Yelan, Jia, Shujie, Mingzhu and many more. Eating and drinking free stuff with you, learning Chinese and German; and going hiking was and is always very enjoyable. I also want to thank my teachers in Kunming, especially my older sister Li Yue.

Thank you Viet, you are the one who brought me to Darmstadt and started this journey.

Lastly, and most importantly, I would like to express my deepest gratitude to my beautiful wife Hong Nhung Le, the love of my life and the closest friend I have. You were always there when things were difficult, I would not have made it without you. Chồng yêu vợ rất nhiều!

Erklärungen laut Promotionsordnung

§8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Klie

Darmstadt, 1. Februar 2024

Jan-Christoph Klie

Wissenschaftlicher Werdegang des Verfassers[1]

Okt 2011 – Sep 2014	Bachelor of Science (B.Sc.) in Angewandter Informatik, Duale Hochschule Baden-Württemberg Mannheim.
Okt 2014 – Okt 2017	Master of Science (M.Sc.) in Informatik, Technische Universität Darmstadt.
Jan 2018 – Apr 2024	Doktorand, Ubiquitous Knowledge Processing (UKP-Lab), Technische Universität Darmstadt.

[1] Gemäß §8 Abs. 1 lit. a der Promotionsordnung der TU Darmstadt

Zusammenfassung

Annotierte Daten sind in vielen wissenschaftlichen Disziplinen unverzichtbar, z. B. in der Verarbeitung natürlicher Sprache, Linguistik, der Spracherwerbsforschung, der Bioinformatik, dem Gesundheitswesen oder den digitalen Geisteswissenschaften. Datensätze werden verwendet, um Modelle mittels maschinellem Lernen zu trainieren und zu evaluieren, um neues Wissen zu generieren und um bestehende Theorien zu erweitern. Insbesondere im Bereich des maschinellen Lernens spielen große, qualitativ hochwertige Datensätze eine entscheidende Rolle, um das Feld voranzubringen und neue Ansätze auszuwerten. Bei der Erstellung dieser essentiellen Datensätze sind zwei Themen von zentraler Bedeutung: Annotationseffizienz und -qualität. In dieser Arbeit werden wir beide Aspekte verbessern.

Annotierte Daten sind von grundlegender Bedeutung und sehr nachgefragt, aber die manuelle Erstellung von Annotationen ist teuer, zeitaufwändig und erfordert oft Experten. Es ist daher sehr wünschenswert, die Annotationskosten zu senken und die Annotationsgeschwindigkeit zu verbessern - zwei wichtige Aspekte der Annotationseffizienz.

In dieser Arbeit schlagen wir daher verschiedene Möglichkeiten zur Verbesserung der Annotationseffizienz vor, darunter Human-in-the-Loop Annotationsvorschläge, interaktives Annotatorentraining und Annotation durch freiwillige Helfer.

Um gut funktionierende Modelle zu trainieren und eine akkurate Auswertung zu ermöglichen, müssen die Daten selbst von höchster Qualität sein. Annotationsfehler können zu schlechten Ergebnissen in der eigentlichen Anwendung führen; Modellvorhersagen können sogar schädlich sein. Wenn fehlerhafte Daten zur Bewertung oder zum Vergleich von Modellarchitekturen, Algorithmen, Trainingssystemen oder anderen Aspekten verwendet werden, kann sich außerdem die relative Reihenfolge der Methoden in Bezug auf die Leistung ändern. Somit können Fehler in annotierten Daten zu falschen Schlussfolgerungen führen. Der Schwerpunkt der meisten Arbeiten im Bereich des maschinellen Lernens liegt auf der Entwicklung neuer Modelle und Methoden; Forschung zur Datenqualität wird dabei oft vernachlässigt. In dieser Arbeit werden zwei Beiträge zur Verbesserung der Annotationsqualität vorgestellt, um Qualitätsprobleme zu reduzieren. Erstens analysieren wir bewährte Verfahren des Annotationsqualitätsmanagements, untersuchen, wie es in der Praxis durchgeführt wird, und leiten daraus Empfehlungen für zukünftige Datensatzersteller ab, wie der Annotationsprozess strukturiert und die Qualität gemanagt werden kann. Zweitens geben wir einen Überblick über den Bereich der automatischen Fehlererkennung bei Annotationen, formalisieren die Aufgabe, implementieren die am häufigsten verwendeten Methoden neu und untersuchen deren Wirksamkeit. Auf der Grundlage umfangreicher Experimente geben wir Einblicke und Empfehlungen dazu, welche Methoden in welchem Kontext verwendet werden sollten.

Abstract

Annotated data is essential in many scientific disciplines, including natural language processing, linguistics, language acquisition research, bioinformatics, healthcare, or the digital humanities. Datasets are used to train and evaluate machine learning models, to deduce new knowledge, and to suggest appropriate revisions to existing theories. Especially in machine learning, large, high-quality datasets play a crucial role in advancing the field and evaluate new approaches. There are two central topics when creating these crucial datasets: annotation efficiency and annotation quality. We improve on both in this thesis.

While annotated data is fundamental and sought after, creating it via manual annotation is expensive, time-consuming, and often requires experts. It is therefore very desirable to reduce costs and improve speed of data annotation, two significant aspects of annotation efficiency. Through this thesis, we hence propose different ways of improving annotation efficiency, including human-in-the-loop label suggestions, interactive annotator training, and community annotation.

To train well-performing models and for their accurate evaluation, the data itself needs to be of the highest quality. Errors in the dataset can lead to degraded downstream task performance, biased or even cause harmful predictions. In addition, when erroneous data is used to evaluate or compare model architectures, algorithms, training regimes, or other scientific contributions, the relative order in performance might change. Thus, dataset errors can cause incorrect conclusions to be drawn. The focus of most machine learning work is on developing new models and methods; data quality is often overlooked. To alleviate quality issues, this thesis presents two contributions to improve annotation quality. First, we analyze best practices of annotation quality management, analyze how it is conducted in practice, and derive recommendations for future dataset creators on how to structure the annotation process and manage quality. Second, we survey the field of automatic annotation error detection, formalize it, re-implement and study the effectiveness of the most commonly used methods. Based on extensive experiments, we provide insights and recommendations concerning which ones should be used in which context.

Contents

Part I

Synopsis

Publications and My Contributions

This thesis is based on six scientific publications to which I contributed as the lead author. They were co-authored together with my advisors Iryna Gurevych and Richard Eckart de Castilho as well as many excellent colleagues and collaborators: Luke Bates, Beto Boullosa, Michael Bugert, Gözde Gül Şahin, Ji-Ung Lee, Nafise Sadat Moosavi, Dominic Petrak, Kevin Stowe, and Bonnie Webber. I am grateful to all my co-authors and their significant contributions to these pleasant as well as successful collaborations. In the following, I describe my own contributions to each publication. Details on our strategy for data handling are given in Appendix A.

Core Publications

Chapter 6 corresponds to the following publication:

> **Jan-Christoph Klie**, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In: *Proceedings of the 27th International Conference on Computational Linguistics (COLING): System Demonstrations*, pages 5–9.

This is the first system demonstration paper describing INCEpTION, our extensible, semantic annotation platform offering intelligent assistance and knowledge management. Richard is the project lead and main developer. I mostly worked on the internal and external recommendation functionality, various features, code review, testing, infrastructure, documentation, user support, student supervision, and dissemination. INCEpTION has been used throughout my research. I wrote the initial draft of the article and performed the subsequent corrections and discussed this work regularly with my advisors, who helped me improve the draft. I presented the work in person at the conference venue.

Chapter 7 corresponds to the following publication:

> **Jan-Christoph Klie**, Richard Eckart de Castilho, Iryna Gurevych. 2023a. Analyzing Dataset Annotation Quality Management in the Wild. In: *arXiv/Under submission.*

I conceived the original research contributions, collected and annotated the data, and performed all implementations, experiments, and analyses. I wrote the initial draft of the article and performed the subsequent corrections. Richard provided the implementation of the PDF editor in INCEpTION, suggested using *Papers with Code* as the means for finding dataset-introducing publications, and helped with his extensive proofreading. I discussed this work regularly with my advisors, who helped me improve the research contributions and draft.

Chapter 8 corresponds to the following publication:

> **Jan-Christoph Klie**, Bonnie Webber, Iryna Gurevych. 2023c. Annotation Error Detection: Analyzing the Past and Present for a More Coherent Future. In: *Computational Linguistics*, 49 (1): 157–198.

I conceived the research ideas after detailed discussions with Bonnie and Iryna. I performed all of the implementation work, planned and conducted all experiments, and performed all of the analyses. I wrote the initial draft of the article and performed the subsequent corrections. I discussed this work regularly with my advisors, who helped me improve the draft.

Chapter 9 corresponds to the following publication:

> **Jan-Christoph Klie**, Richard Eckart de Castilho, Iryna Gurevych. 2020. From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6982–6993.

The idea for this work came from Richard and was initially given as a Bachelor thesis topic to Peter Jiang, whom Richard and I co-supervised. I extended the original idea from using heuristics to learning-to-rank, conceived the research questions, collected the data, planned as well as ran the experiments, devised and conducted the user study, and performed all analyses. I did most of the implementation work; for some parts related to INCEpTION, Richard advised me how to best add the required functionality. I wrote the initial draft of the article and performed the subsequent corrections. I discussed this work regularly with my advisors, who helped me improve the draft.

Chapter 10 corresponds to the following publication:

> Ji-Ung Lee*, **Jan-Christoph Klie***, Iryna Gurevych. 2022. Annotation Curricula to Implicitly Train Non-Expert Annotators. In: *Computational Linguistics*, 48 (2): 343–373.
>
> (*: equal contribution)

This is a joint publication between Ji-Ung and me. We developed the core ideas during several brainstorming sessions. We researched and wrote the introduction, background, definition, and conclusion together. We also made the subsequent corrections together. I collected and selected the datasets for the simulation described in §10.4, implemented, executed, analyzed the experiments, and wrote the section itself. Ji-Ung planned, conducted, supervised, and analyzed the user study described in §10.5 and also wrote the respective section. We discussed this work regularly with our advisors, who helped us improve the draft.

Chapter 11 corresponds to the following publication:

> **Jan-Christoph Klie**, Ji-Ung Lee, Kevin Stowe, Gözde Gül Şahin, Nafise Sadat Moosavi, Luke Bates, Dominic Petrak, Richard Eckart de Castilho, Iryna Gurevych. 2023b. Lessons Learned from a Citizen Science Project for Natural Language Processing. In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL): Main Volume*, pages 6982–6993.

The original research idea was conceived by Kevin, who spearheaded running the user study. I consulted him on how to best use INCEpTION for the user study and helped with planning and conducting it. After Kevin left our group, I stepped in as the project lead and brought this work to publication. I wrote the largest part of the initial draft of the article, did the analysis, and performed most of the subsequent revisions. The other authors helped with conceptualizing the project, selecting the data, recruiting annotators in the various channels, and writing small parts of the paper itself. I regularly discussed this work with the team and my advisors. Our advisors helped sharpen the research questions and supported us in improving the draft.

Other Publications

During my time as a Ph.D. student, I was fortunate to work with great researchers on various topics, some of which did not fit into this thesis. In the interest of completeness, I provide references to these papers:

> Beto Boullosa, Richard Eckart de Castilho, Naveen Kumar, **Jan-Christoph Klie**, Iryna Gurevych. 2018. Integrating Knowledge-Supported Search into the INCEpTION Annotation Platform. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 127–132.

> Teresa Botschen, Iryna Gurevych, **Jan-Christoph Klie**, Hatem Mousselly-Sergieh, Stefan Roth. 2018. Multimodal Frame Identification with Multilingual Evaluation. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long Papers)*, pages 1481–1491.

> Richard Eckart de Castilho, Nancy Ide, Jin-Dong Kim, **Jan-Christoph Klie**, Keith Suderman. 2019. A Multi-Platform Annotation Ecosystem for Domain Adaptation. In: *Proceedings of the 13th Linguistic Annotation Workshop*, pages 189-194.

> Richard Eckart de Castilho, Nancy Ide, Jin-Dong Kim, **Jan-Christoph Klie**, Keith Suderman. 2019. Towards cross-platform interoperability for machine-assisted text annotation. In: *Genomics & Informatics*, 17(2):e19.

Chapter 1

Introduction

Annotated data is an essential component in many scientific disciplines, including natural language processing (NLP) (Gururangan et al., 2020; Peters et al., 2019), linguistics (Haselbach et al., 2012), language acquisition research (Behrens, 2008), bioinformatics (Zeng et al., 2015), healthcare (Suster et al., 2017), and the digital humanities (Schreibman et al., 2004). Annotation is defined as enriching data with additional information, for example, assigning labels to texts so that they can be more easily processed by machines (Chapter 2). Datasets are used to train and evaluate machine learning models, to deduce new knowledge, and to suggest appropriate revisions to existing theories. Especially in machine learning, large, high-quality datasets play a crucial role in advancing the field (Sun et al., 2017; Sambasivan et al., 2021).

To train well-performing and accurate models as well as for their evaluation, the data itself needs to be of the best-possible quality. For instance, large language models often require high-quality annotated data, be it for (instruction) finetuning or their evaluation (Chen et al., 2023; Zhang et al., 2023; Zhou et al., 2023). Errors in the dataset can lead to degraded downstream task performance or even cause harmful predictions. As an example, models are already used in recruitment (Maheshwary and Misra, 2018; Luo et al., 2019) or legal cases (Rodrigues, 2020). There, they can cause real-world harm like discrimination or wrong rulings. When used to evaluate or compare model architectures, algorithms, training regimes, or other scientific contributions, the performance ranking might change (Reiss et al., 2020). This might cause incorrect conclusions to be drawn.

The focus of most machine learning work is on developing new models and methods; data quality is often overlooked and researched only seldomly (Sambasivan et al., 2021). Recent work has shown that even commonly used datasets contain annotation errors (Northcutt et al., 2021b). To improve the situation, we present two contributions that can help improving data quality. Our first contribution is analyzing best practices of annotation quality management, analyzing how it is conducted in the wild, and deriving recommendations for future dataset creators on how to best structure the annotation process and manage quality. The second contribution concerns automatic annotation error detection. We survey the field, formalize the task, re-implement and compare the most well-known methods, and show which of them should be used in which context.

While annotated data is fundamental to many fields of science and highly sought after, creating it via manual annotation is expensive, time-consuming, and often requires experts (Monarch, 2021). It is hence very desirable to reduce costs and improve annotation speed of data annotation, two major aspects of *annotation efficiency*. Through this thesis, we therefore explore several different ways of improving annotation efficiency. These are interactive annotation suggestions, leveraging citizen science for NLP dataset creation and training annotators during annotation via annotation curricula.

The goal of dataset creation here is not only to train machine learning models. Annotated datasets can and are also important for other applications, for example, devising linguistic theories or investigating information extraction methods. Therefore, we are interested in methods that are domain-agnostic and widely applicable.

To summarize, this thesis is guided by the following research questions:

- How can we efficiently and effectively improve annotation quality?
- How can we improve annotation speed and reduce annotation costs while retaining annotation quality?

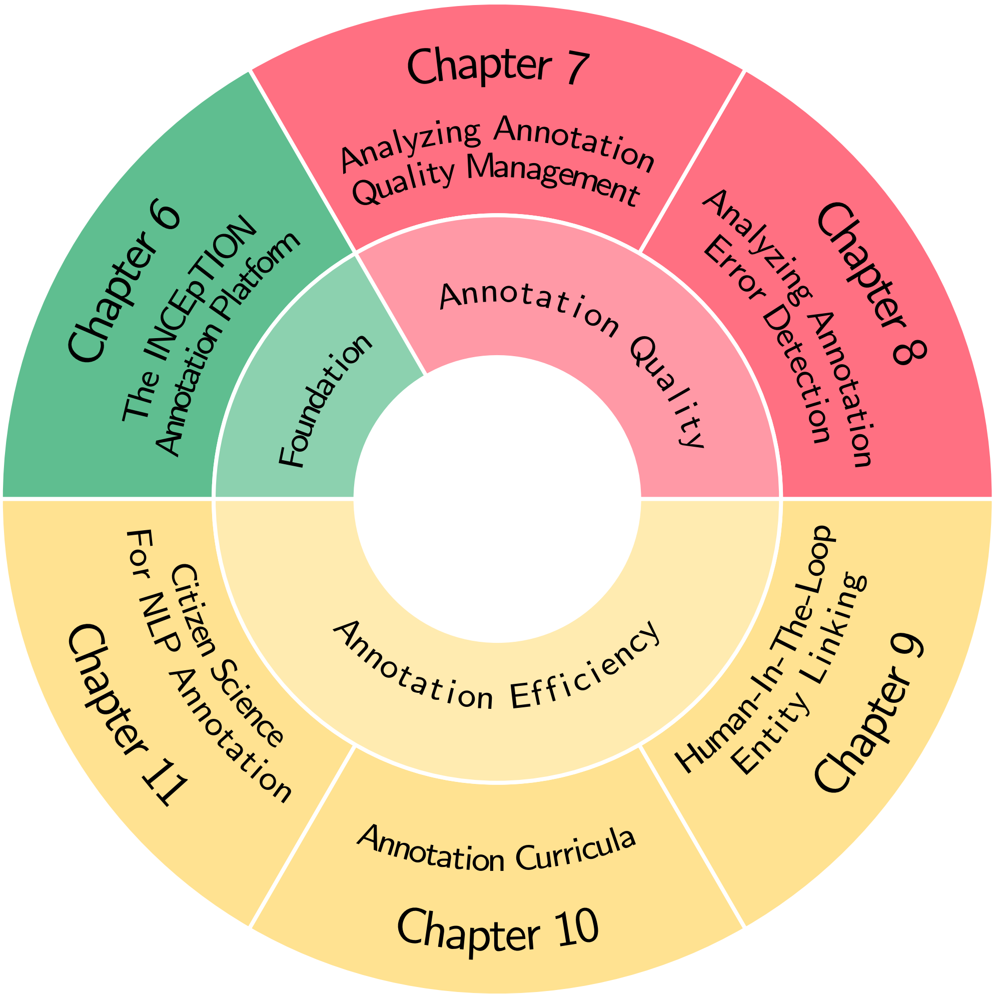

Chapter 7
Analyzing Annotation Quality Management
Chapter 6
The INCEpTION Annotation Platform
Chapter 8
Analyzing Annotation Error Detection
Foundation
Annotation Quality
Annotation Efficiency
Chapter 11
Citizen Science For NLP Annotation
Chapter 9
Human-In-The-Loop Entity Linking
Annotation Curricula
Chapter 10

Figure 1.1: Overview of how the publications making up this thesis are related to annotation quality and annotation efficiency. INCEpTION is used throughout the thesis as the foundation of most publications, either for annotating itself or as the platform hosting the novel methods devised as part of this thesis.

1.1 Thesis Outline

In the following, we give a brief overview of the structure and contents of this thesis. The synopsis summarizes the topic of annotation for natural language dataset creation, followed by the two core aspects: annotation quality and annotation efficiency.

Chapter 2 - Annotation Process Annotated data is an essential ingredient for many fields of science, especially machine learning. This chapter summarizes the annotation process, focusing on annotation for natural language processing.

Chapter 3 - Annotation Quality To train well-performing and precise machine learning models and for their evaluation, the annotated data must be of high quality. We give an overview of annotation quality, why it is important, and how it can be improved. The most relevant methods to manage and improve annotation quality are presented.

Chapter 4 - Annotation Error Detection We discuss automatic annotation error detection, a part of annotation quality, in detail. Annotation error detection describes algorithms that can automatically flag or score instances with regard to their correctness. These can save costs, as manually searching and correcting errors is expensive and time-consuming.

Chapter 5 - Annotation Efficiency Datasets are usually manually annotated, which is expensive and difficult while often requiring experts. Therefore, making the annotation processes more efficient and supporting annotators throughout is advantageous. We discuss different approaches for improving annotation efficiency, especially with regard to reducing time and costs.

Then, the publications that make up this dissertation are listed; they are in the same order as the publication record given in Publications and My Contributions. Figure 1.1 illustrates how they fit into the big picture of annotation quality and efficiency.

Chapter 6 - The INCEpTION Annotation Platform This dissertation was written as part of the INCEpTION project. The project aimed to build a configurable, web-based platform for annotating text documents at span, relation, and document levels. Leveraging its extensibility, INCEpTION was used as the foundation for most of the following publications.

Chapter 7 - Analyzing Annotation Quality Management High-quality annotated data is imperative for training and evaluating machine learning models. However, recent work has shown that even widely used benchmark datasets still contain non-negligible amounts of annotation errors. We hence summarize best practices and analyze a large corpus of publications introduction new datasets with regard to their annotation quality management.

Chapter 8 - Analyzing Annotation Error Detection To reduce costs for finding annotation errors, many different algorithms for automatic annotation error detection have been devised over the years. However, they rarely compare their methods to previous work or on the same datasets, making evaluation and comparisons of methods difficult. To improve the situation, we properly define the task, re-implement the most popular methods and evaluate them on various tasks and datasets.

Chapter 9 - Human-In-The-Loop Entity Linking Entity Linking, that is, disambiguating entity mentions in a text against knowledge bases, is an essential tool in a considerable number of fields like the digital humanities or biomedical sciences. It is a complex and often tedious annotation task, especially for low-resource domains with noisy texts. Hence, we propose a new kind of annotation support using recommenders that suggest potential concepts and adaptive candidate ranking. In a simulation and a user study, we show a significant reduction in annotation time.

Chapter 10 - Annotation Curricula Dataset creation projects often require annotators to familiarize themselves with the task, its annotation scheme, and the data domain on the fly. This can be overwhelming, mentally taxing, and induce errors in the resulting annotations, especially in scenarios where domain expertise is not required. To alleviate these issues, we propose annotation curricula, an approach to implicitly train annotators. The goal is to gradually introduce annotators into the task by ordering annotation tasks according to a learning curriculum, for example, by perceived difficulty. In a simulation and a user study, we show that annotation time can be significantly reduced compared to a random ordering without negatively impacting annotation quality.

Chapter 11 - Citizen Science For NLP Annotation Citizen science describes the participation and collaboration of volunteers from the general public with researchers to conduct science; it is often used for environmental data collection and analysis. By asking the community to re-annotate parts of an already existing, crowdsourced dataset, we show that citizen science can, under certain circumstances, be a viable way for collecting annotations for other NLP tasks.

We conclude in Chapter 12 by summarizing the main research contributions of this thesis and considering future research directions.

Chapter 2

Natural Language Dataset Creation

Having large, high-quality labeled datasets available is essential for developing, training, evaluating, and deploying reliable machine learning models (Sun et al., 2017; Bender and Friedman, 2018; Peters et al., 2019; Gururangan et al., 2020; Sambasivan et al., 2021). They are also used in many other fields, for example, bioinformatics (Kim et al., 2003), healthcare (Suster et al., 2017), or the digital humanities (Nantke and Schlupkothen, 2020). A special type of dataset is a *corpus*, which — in the context of linguistic research — is a set of texts collected for a particular purpose according to certain criteria relevant to that purpose. We use both terms interchangeably to refer to a collection of labeled instances.

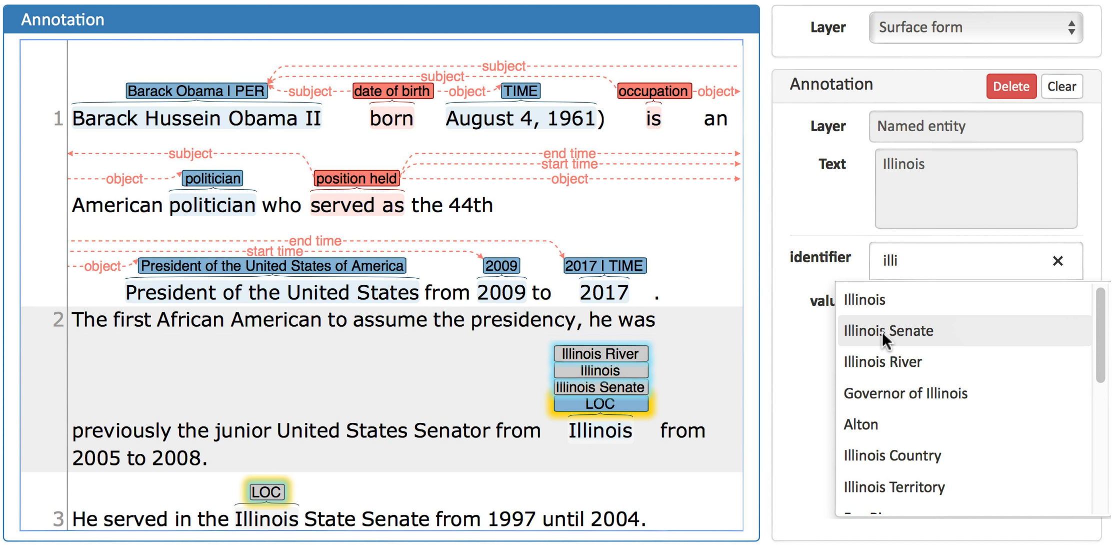
In the context of natural language processing (NLP), similarly to Shmueli et al. (2021), we find three different categories of tasks annotators are asked to perform when creating such datasets. These are *annotation*, *production* and *evaluation*:

Annotation (also called labeling throughout this work) describes the activity of enriching data like text, images, audio, or video with additional information. The goal is often that the new, structured information can then be better processed by computers, e.g., for training machine learning models (Pustejovsky and Stubbs, 2013). In the context of this thesis, we think of annotation as assigning labels to text. Annotations can be made of different levels of granularity. With regard to text annotation, the focus of this work, the unit of annotation can be for instance tokens, spans, sentences, paragraphs or documents.

Production means that the annotators create the data themselves, e.g., writing new texts, summarizing, or paraphrasing. This is commonly done when creating datasets for tasks like question answering or natural language inference.

Evaluation subsumes activities like comparing, ranking, or scoring data items for quality or other metrics.

This dissertation primarily focuses on annotation, which is one of the most important tasks when creating datasets of supervised training of machine learning models. Therefore, this this chapter discusses annotation in more detail. Nevertheless, many points in this thesis also apply to the tasks of production or evaluation.



1 Barack Hussein Obama II born August 4, 1961) is an American politician who served as the 44th President of the United States from 2009 to 2017 .

2 The first African American to assume the presidency, he was previously the junior United States Senator from Illinois from 2005 to 2008.

3 He served in the Illinois State Senate from 1997 until 2004.

Layer: Surface form

Annotation

Layer: Named entity

Text: Illinois

identifier: illi ×

valu
Illinois
Illinois Senate
Illinois River
Governor of Illinois
Alton
Illinois Country
Illinois Territory

Figure 2.1: Example annotation project for named entity recognition, entity linking, and relation labeling using the INCEpTION annotation tool. Annotations in grey are automatically created suggestions (§5.1.2).

2.1 Annotation

As the name implies, annotated datasets are created by annotating data, i.e., assigning labels to pieces of text (Monarch, 2021).

We call the smallest unit of data that is annotated an *instance*; a collection of annotated instances is called an annotated dataset. What makes up an instance depends on the task. For example, text can be annotated at the document level for document classification, paragraph level for argumentation mining, sentence level for sentiment analysis, span level for named entity recognition (see Figure 2.1), word level for part-of-speech tagging, or character level for morphological annotation.

Annotations are made with regard to an *annotation schema*, which describes the kind of annotations that can be made, for instance, which categories can be assigned. For text documents, this can, for example, be for annotating sentiment, i.e., whether the content is positive, negative, or neutral. For spans, it can be the kind of named entity, for example, person, location or organization. *Annotation guidelines* describe how to annotate, e.g., which categories to assign for which instances in which context.

Annotations are made in annotation editors, which are the user interfaces that display the data to annotate and provide labeling tools. Annotation editors can also enable user management, configure the annotation schema, and integrate additional functionalities like collaboration, quality management, or adjudication. Commonly used editors for text annotation are GATE Teamware (Wilby et al., 2023), brat (Stenetorp et al., 2012), Doccano (Nakayama et al., 2018), or WebAnno (Eckart de Castilho et al., 2016). This dissertation was developed as part of the INCEpTION project (Klie et al., 2018); INCEpTION was used throughout the publications that make up this thesis. Compared to other annotation tools, INCEpTION is a annotation platform that incorporates all the tasks related to the annotation process into a joint web-based platform. It was designed be extensible from the start and supports many different annotation functionalities out of the box. Its extensibility allowed us to integrate the newly developed methods presented in this work. An example view of INCEpTION is shown in Figure 2.1.

2.2 Annotators

When high-quality datasets are to be created, then the annotations are usually made by human *annotators*, which are given instances (here, pieces of text) for labeling. Hence, dataset annotation often involves a non-negligible amount of manual labor to create annotated datasets (Snow et al., 2008). Creating large, annotated datasets can take many months or years and cost tens or hundreds of thousands of dollars (Francis and Kucera, 1979; Hovy et al., 2006). While there are also ways that fully automatically create annotations without manual labeling (e.g., Ratner et al., 2017; Smirnova and Cudré-Mauroux, 2019), the resulting annotations are often of lower quality than those using human annotators (Mintz et al., 2009) and out of scope for this thesis.

Depending on the kind of task, its difficulty, and the available budget, different types of annotators can be hired to work on a dataset creation project. In case specific domain knowledge is needed, for instance, for annotating linguistic, legal, biomedical, or financial phenomena, expert annotators might be required. As an example, early, "classic" corpora for linguistics like the Penn Treebank corpus (Marcus et al., 1993) were usually annotated by experts (Chamberlain et al., 2013).

Snow et al. (2008) and Callison-Burch (2009) have shown that many annotation tasks can be phrased as so-called *micro-tasks*. These tasks that do not need context can be completed in seconds or minutes and require only a little skill or experience (Finnerty et al., 2013). Micro-tasks can be farmed out to so-called crowdworkers, which are (often anonymous) freelancers that accept work via online crowdsourcing platforms like Amazon Mechanical Turk or Appen. The annotation costs per task are often in the range of cents (Whiting et al., 2019). Because micro-tasks are simple and cheap and crowdworkers do not need special qualifications, crowdsourcing can be used to create very large datasets, which would be infeasible or impossible with expert annotation.

For several reasons, annotations created via crowdsourcing oftentimes have quality issues (Northcutt et al., 2021b). Compared to experts, crowdworkers have less experience and qualifications for annotating. They are also usually fiscally motivated and want to finish as many tasks as possible in the shortest amount of time. This can result in them not taking special care or even actively spamming to quickly finish tasks (Hovy et al., 2013). Recently, crowdworkers themselves have been observed using machine learning in the form of large language models to do the annotation work for them (Veselovsky et al., 2023). This is an issue, as the data is often used to evaluate these very models, thus poisoning the created dataset. For these reasons, quality management is crucial when creating high-quality datasets with crowdsourcing (Callison-Burch and Dredze, 2010). Often, many labels per instance are collected to alleviate most quality issues in a step called *adjudication* (see §3.2.4).

Crowdsourcing can have several potential ethical issues that need to be considered (Schlagwein et al., 2019). Crowdsourcing has been criticized for undermining workplace regulations, akin to digital sweatshops. Frequently, crowdworkers barely make the (US) minimum wage, are exposed to global arbitrage, or are required to take on low-paid tasks to qualify for better work (Fort et al., 2011; Kummerfeld, 2021).

But overall, if the annotation task is solvable by crowdworkers, then crowdsourcing is often an efficient way to create datasets (Snow et al., 2008; Hovy et al., 2014).

Contractors are a middle-ground between crowdworkers and experts; they might have experience conducting annotation tasks and a background in the domain but are not necessarily experts in their field (Peer et al., 2017). These can be, for instance, student workers or qualified freelancers. Recently, more and more datasets are annotated by contractors that are hired by platforms like Upwork or Prolific (Chen et al., 2021). Compared to crowdsourcing, usually, only a few contractors are hired for a project that in turn annotate more. Hence, they can be individually trained and given feedback. More complex tasks can be annotated that way that might not be suitable for crowdsourcing. Hiring contractors can be cheaper than crowdsourcing as fewer workers and fewer repetitions are needed. A disadvantage is that hiring such contractors itself might be difficult; crowdsourcing platforms can be easily tapped for annotators, whereas finding skilled contractors requires time and effort.

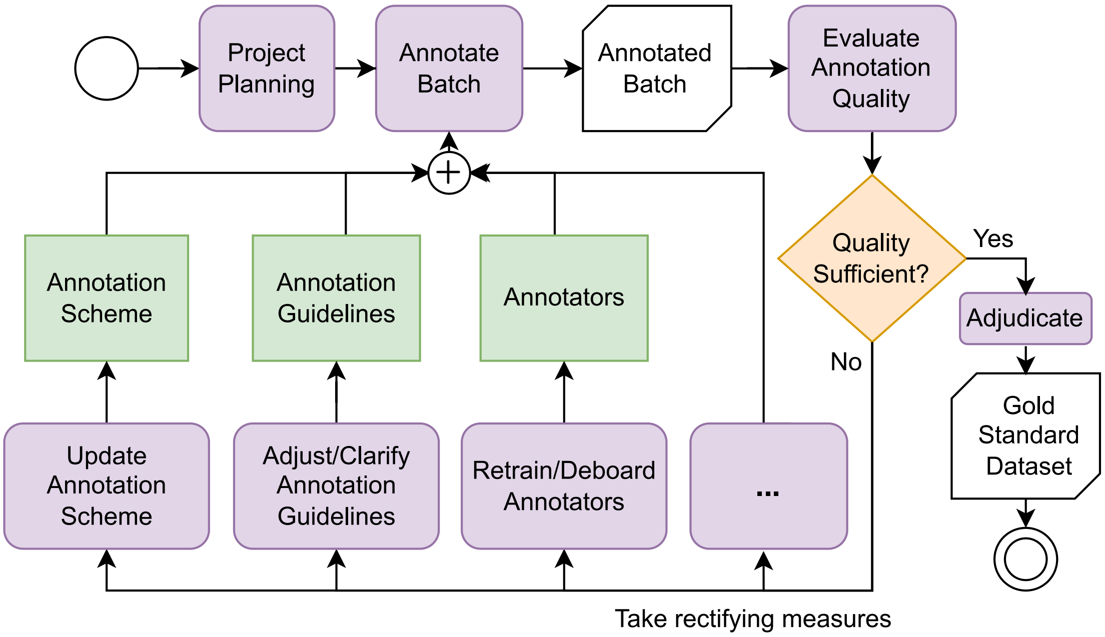


Figure 2.2: Agile dataset creation. After a batch of data is annotated, it is evaluated. If the quality is sufficient, it can be adjudicated. If not, several corrective measures can be taken, e.g., correcting the annotations in an additional step, annotator training, or adjusting the annotation scheme or guidelines. This process similarly applies to text production workflows where usually no adjudication occurs.

2.3 Annotation Process

Dataset creation is a quite involved process consisting of many different steps. To better understand its parts, we give an overview of the activities that make up the typical, recommended annotation project in the following section. It is also depicted in Figure 2.2.

An annotation project usually starts with a *planning phase*. It can encompass important preliminaries as setting the goal of data collection, making initial choices for data and annotators, setting a budget, desired quality level or reviewing the literature for similar datasets or relevant annotation practices. Ideally, these choices are documented and become part of the dataset documentation once the dataset gets released.

In many cases, the annotation scheme is developed during the course of an annotation study and is a living document. As annotators only get familiar with the task during the annotation process and can make errors, the task setup and annotations must be repeatedly refined. Therefore, it is best practice to structure an annotation campaign as a sequence of cycles with iterative quality improvement actions (Hovy and Lavid, 2010; Pustejovsky and Stubbs, 2013; Monarch, 2021). This approach is also called *agile corpus creation* (Alex et al., 2010). In each cycle, only a slice of the data is annotated: a batch. After a batch has been annotated, its quality is evaluated (§3.2.2), and quality-improving/rectifying measures are taken if needed (§3.2.3). These cycles repeat until an acceptable quality level for a sufficient number of batches has been reached. Frequently, multiple annotation per instance are collected to increase reliability and assure quality. These need to be adjudicated at the end to create gold instances with just one curated gold label (§3.2.4).

Chapter 3

Annotation Quality Management

Annotated data is an essential ingredient to train and evaluate machine learning models. These models are then used to compare aspects like network architectures, training regiments, or hyperparameter configurations (Reimers and Gurevych, 2017). Machine learning models are also deployed in production and used to make business decisions or even directly interact with humans (Ameisen, 2020; Monarch, 2021). Therefore, it is crucially important that the underlying data is of the best-possible quality. Datasets that are of the highest quality, are trustworthy and accepted as adhering to the annotation schema and guidelines are also called *gold-standard* (Wissler et al., 2014).

Recently, conversational agents and search engines based on large language models trained via instruction tuning have been widely adopted in science and society (Ouyang et al., 2022; Wei et al., 2022). It is hence imperative that datasets used for fine-tuning are factually correct and contain as few biases as possible for the resulting models to be accurate, trustworthy and not to cause misinformation or harm. Also, when fine-tuning large language models, it has been observed that fewer instances of higher quality yield better results than more instances of lower quality (Chen et al., 2023; Zhang et al., 2023; Zhou et al., 2023).

In machine learning, it is often taken for granted that gold-standard datasets have no or only very few errors. Recent work, however, has shown that even datasets that are widely used to train and evaluate state-of-the-art models contain non-negligible proportions of questionable labels (Northcutt et al., 2021b). For instance, the CONLL-2003 (Tjong Kim Sang and De Meulder, 2003) test split has an estimated 6.1% wrongly labeled instances (Reiss et al., 2020; Wang et al., 2019), IMAGENET 5.8% (Vasudevan et al., 2022; Northcutt et al., 2021b) and TACRED 23.9% incorrect instances (Stoica et al., 2021). GOEMOTIONS (Demszky et al., 2020) is estimated to contain even up to 30% wrong labels.[1] Using these datasets for machine learning can —among other issues— lead to inaccurate estimates of model performance (Manning, 2011; Reiss et al., 2020; Vasudevan et al., 2022), generalization failure due to data bias (McCoy et al., 2019), or decreased task performance (Nettleton et al., 2010; Stoica et al., 2021; Vădineanu et al., 2022) When deployed in production, deployed models might even cause downstream harm to users (Bolukbasi et al., 2016; Hildebrandt, 2019; Cheng et al., 2022), especially in critical applications like medical or legal settings.

[1] `https://archive.ph/jQbNM`

In order to alleviate these issues and to create high-quality datasets, proper quality management needs to be applied throughout the annotation process. Quality management encompasses, among others, proper data selection, choice of annotators and training, creating and improving annotation schema and guidelines as well as annotator agreement, data validation and error rate estimation (Hovy and Lavid, 2010; Alex et al., 2010; Pustejovsky and Stubbs, 2013; Monarch, 2021).

As the literature concerning annotation quality management is quite scattered throughout many books and scientific publications, we first give a brief summary on annotation quality management for data annotation. Based on this, we then analyze how annotation quality management is handled in practice by inspecting a large set of publications that introduce new datasets (Chapter 7).

3.1 Quality Aspects

Before describing quality management methods, we first define what annotation quality actually subsumes. Following Krippendorff (1980); Neuendorf (2016), we suggest targeting at least the following quality aspects:

Stability An annotation process is stable if there is no drift over time in its output. Instability can, for example, occur due to carelessness, distractions, tiredness, or even learning through practice.

Reproducibility An annotation process is reproducible if different annotators under varying conditions can still deliver the same results.

Accuracy Annotations are accurate if they adhere to the guidelines and the desired outcome.

Unbiasedness This describes the extent to which the annotations are free of systematic, nonrandom errors (bias).

Stability, *reproducibility*, and *accuracy* are also subsumed under the term *reliability* in content analysis (Krippendorff, 1980). *Consistency* is related to *stability* and *reproducibility*. *Reliability* thus measures the differences that occur when repeatedly annotating the same instances; it is empirical (Hardt and Recht, 2022). It is required to infer *validity*, that is, to show that the annotations capture the underlying phenomenon targeted (Artstein and Poesio, 2008), but not sufficient. *Validity* is latent and cannot be directly measured. Therefore, proxy metrics targeting reliability, for example agreement, need to be used instead.

Note that dataset quality is not only limited to label accuracy but also encompasses aspects such as the quality of the underlying text, annotation scheme, and social or data bias.

3.2 Methods

This section presents the most relevant and frequently used quality management methods for data annotation. It is a condensed summary of §7.3. This list is derived from best

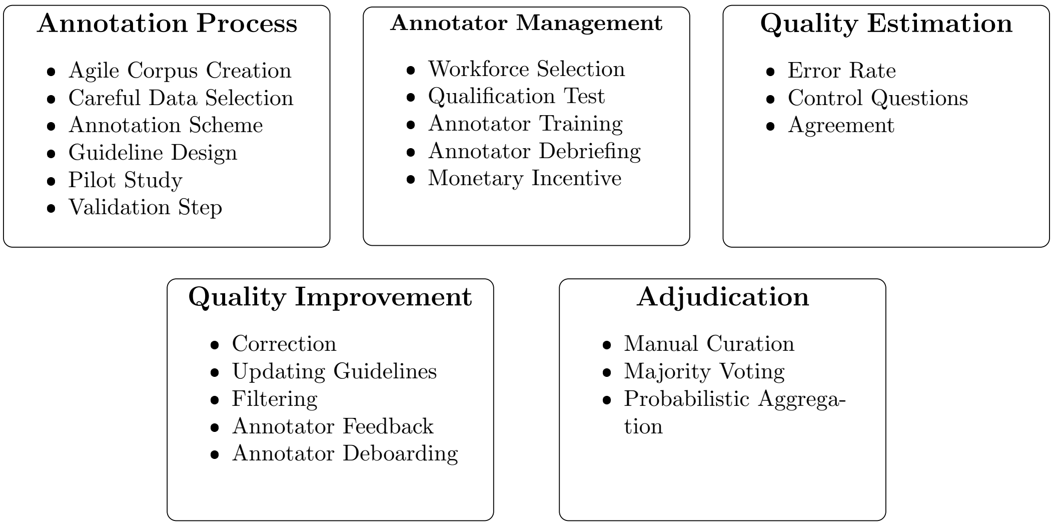


Figure 3.1: Quality Management methods discussed in this work. We categorize methods into annotation process, annotator management, quality estimation, quality improvement, and adjudication. The annotation process is described in §2.3.

practices in previous works and the methods found while surveying the dataset papers (Chapter 7). An overview of the discussed methods is also given in Figure 3.1.

3.2.1 Annotator Management

Annotation projects stand or fall with the quality of the annotators; an annotation project is often an exercise in people management (Monarch, 2021). It is crucial to treat annotators fairly and respectfully throughout the process. The following gives a high-level overview of the different aspects of annotator management that influence annotation quality.

Annotator Selection The background of the workers who annotate considerably impacts annotation time, cost, and quality. Here, we differentiate between domain experts, contractors with annotation but not necessarily domain experience, crowdworkers, and volunteers (see also §2.2). The more experienced an annotator is, the higher the resulting quality of their annotation, but also their cost. We discuss the impact of workforce selection on annotation time and cost in more depth in §5.1.1.

Qualification Filter As a common way to filter out annotators that might produce low-quality work, many crowdsourcing tools offer setting requirements for the worker, such as a certain percentage of accepted tasks or a number of already completed tasks. Kummerfeld (2021) analyses the impact of these measures on quality and discusses the ethical aspects of requiring a minimum number of tasks, as it forces lower-paying work to reach these requirements.

Qualification Test A more elaborate way to restrict annotators from participating in a task is to use qualification tests (Kummerfeld, 2021). Before an interested annotator can participate in the annotation process itself, they need to work on a small set of qualification tasks. If their performance is acceptable, then annotators are allowed to work on the real annotations.

Annotator Training Before involving new annotators in a campaign, it is often beneficial to train them in the annotation task at hand, to go through the guidelines with them and make sure that everything is clear (Neuendorf (2016, p. 133); Sabou et al. (2014)). Training is also vital for annotation stability, as early in the process, annotators are often not sure and unfamiliar with the annotation process. This changes with more time spent annotating, rendering earlier annotations potentially inconsistent with later ones.

Annotator Debriefing During and after the run of an annotation project, it is often useful to ask one's annotators for feedback about the annotation project (Neuendorf, 2016, p. 134). This input can then, for instance, be used to improve the guidelines, update the annotation scheme, or alleviate issues that became only apparent while annotating.

Monetary Incentive Giving annotators additional monetary compensation in addition to their base pay might be an option (Harris, 2011; Ho et al., 2015). This can be, for instance, based on their performance on control questions or after feedback rounds have shown that they reach the target for a bonus. Another way is to pay annotators more for sticking to a task (Parrish et al., 2021).

3.2.2 Annotation Quality Estimation

After annotations have been created, their quality should be estimated and compared to the desired quality level. In case it is insufficient, counter-measures should be taken to improve it.

Manual Inspection In order to judge the quality of an instance dichotomously as correct or incorrect, annotators (usually, they are different from the annotators who created the instances in the first place) or project managers can manually inspect and grade them. Inspection can either be done on a subset of instances or as a complete validation step. After the dataset has been completely annotated, its error rate can be estimated and reported because even datasets that are considered gold often still contain errors (Northcutt et al., 2021b).

Control Instances To gauge the performance of annotators, control instances can be injected into the annotation process for which the answer is known (Callison-Burch and Dredze, 2010). Another way is to compare a single annotator's submissions to the others'; the performance estimate is then the deviation from the majority vote (Hsueh et al., 2009) or the agreement (Monarch, 2021). For example, the resulting estimates then can be used to assign more involved tasks like manual adjudication to well-performing annotators, retrain annotators if they annotated too many instances incorrectly, send batches created by underperforming annotators back for re-annotation or remove annotators from the workforce.

Agreement A common way to quantify the reliability of annotations and annotators is to compute their inter-annotator agreement (Ebel, 1951; Krippendorff, 1980, 2004). For natural language processing, it has been increasingly adopted after Carletta (1996) introduced agreement, coming from the field of content analysis, as an alternative to previously used ad-hoc measures.

One of the most common agreement measures is percent agreement (Klie et al., 2023a). It is computed as the percentage of coded units on which two annotators agreed. Percentage agreement suffers, however, from several issues (Krippendorff, 1980, 2004; Artstein and Poesio, 2008). First, it yields skewed results for imbalanced datasets, similar to accuracy when evaluating classification. Second, it does not take into account when annotators assign the same label by chance, for example, in case they just randomly guess or spam. Third, percent agreement is influenced by the size of the label set and thus is difficult to compare across annotation schemes.

In order to remedy the issues of percent agreement, Cohen (1960) proposes a chance-corrected coefficient to measure the agreement between two annotators called Cohen's κ. It is the quotient of chance agreement and observed agreement. Fleiss (1971) extend Scott's π (Scott, 1955) to multiple annotators (Fleiss' κ).[2] Cohen's and Fleiss' κ both require that all instances are annotated by the same number of annotators; no entries may be missing. In addition, Fleiss' κ assumes that annotators for each instance are sampled randomly, it is not suitable for settings where all annotators annotate all instances (Fleiss et al., 2003). Also, annotations need to be categorical. κ are one of the most commonly used agreement measures for text annotation.

A different way to estimate agreement has been proposed by Krippendorff (1980). It is based on the quotient of observed and chance disagreement. Compared to Fleiss' κ, Krippendorff's α is more powerful and versatile: it can deal with missing annotations, supports more than two annotations per instance, and can be generalized to even handle categorical, ordinal, hierarchical, or continuous data (Hayes and Krippendorff, 2007). For instance, span labeling tasks like named entity recognition or relation extraction can be evaluated using a coefficient of the Krippendorff's unitized α (α_u) family Krippendorff et al. (2016).[3] Unitizing means that annotators first divide the instances into smaller units and only then assign labels (Lombard et al., 2002, Chapter 4). In the context of named entity annotation, unitizing, for example, can be marking spans that contain entities. Hence, Krippendorff's α can also be applied to any task with a one-to-many relation between instances and annotations.

[2] Fleiss' κ is not an extension of Cohen's κ, as it assumes similarly to Scott's π that the labeling distributions are the same for each annotator, which Cohen's κ does not (Artstein and Poesio, 2008).

[3] The α_u family currently consists of four different coefficients Krippendorff et al. (2016). They differ in how and whether 'gaps' (unannotated units) are take into consideration, whether labels or only units are used, or whether only a subset of labels are used when computing agreement. α_{cu} is the most applicable choice of the four that ignores gaps and takes label values into account.

Evaluation tasks (Chapter 2) consist of assigning scores to instances on a numerical, continuous, or discrete rating scale or a Likert (Likert, 1932) scale. These tasks are, among others, annotating sentiment (Socher et al., 2013), emotions (Demszky et al., 2020), or semantic textual similarity (Cer et al., 2017). Correlation measures like Pearson's r, Spearman's ϱ, or Kendall's τ are often used to compute agreement in these settings. Nevertheless, using correlation coefficients as an agreement measure is controversial, as they measure covariation, not agreement, i.e., they measure whether variables move together, but not whether they are actually similar (van Stralen et al., 2012; Ranganathan et al., 2017; Edwards et al., 2021). A better alternative to the aforementioned correlation coefficients is using Intraclass Correlation (ICC) (Fisher, 1925), which is explicitly designed to measure agreement.

Especially for sequence labeling tasks like named entity recognition, classification metrics like *accuracy*, *precision*, *recall*, and F_1 are often used between two annotators to compute agreement (Brandsen et al., 2020). This comes with several issues. First, they are only applicable as pairwise agreement; having more annotators would require averaging, which might lose information. Second, they are not chance-corrected (Powers, 2011). Third, using precision and recall for computing agreement also has the downside of not being symmetric. Given two lists of labels a and b, the precision value for p of a and b turns into the recall when swapping its arguments: $\text{precision}(a, b) = \text{recall}(b, a)$. Being symmetric is essential for agreement metrics, as one annotator should not be preferred over another. This differs from classification metrics, where one input is from the gold data, and the other is usually from model predictions.

For a more in-depth treatment of agreement and how to apply it, we refer the interested reader to the excellent works of Krippendorff (1980); Lombard et al. (2002); Artstein and Poesio (2008); Neuendorf (2016); Monarch (2021).

3.2.3 Quality Improvement

In case quality estimation has shown that the annotation quality is insufficient, rectifying measures need to be taken to improve it.

Manual Correction If the quality in a batch of annotations is too low, then it can be given back to the annotators for further improvement (Monarch, 2021). Also, it can be routed to different, more experienced annotators to resolve issues in case instances are found to be too difficult for the original annotators (Yang et al., 2019).

Updating Guidelines It can happen that the annotation guidelines are not covering certain phenomena in the underlying text, are ambiguous or difficult to understand. Then, it might be appropriate to go back to the annotation schema or guidelines and improve them (Pustejovsky and Stubbs, 2013). Updating the guidelines may require discarding previously created annotations or at least reviewing and updating them.

Data Filtering Sometimes, certain instances are too ambiguous and annotators strongly disagree on a single, correct label. Or annotations are of low quality and should be removed. A simple solution is to filter out these instances and not process them further. The filtering can for instance be based on expert judgement or if there is no majority agreement. Sometimes, it might also be useful to measure the time it takes annotators to process instances and filter out annotations which have improbably low annotation times.

Filtering instances has the potential disadvantage of reducing diversity, which should be taken into account. When removing difficult instances and using the resulting dataset for evaluating machine learning models, its performance might be overestimated. Recent work also emphasizes that disagreement is inherent to natural language (Aroyo and Welty, 2015) and can for instance be used to create a hard dataset split or even directly learn from them (Checco et al., 2017; Uma et al., 2021).

Annotator Training through Feedback After a batch has been completed, experts can manually inspect the data and give annotators feedback on it. Thereby, common errors can be pointed out and aspects to improve can be discussed. More detailed and extensive feedback might be more feasible for smaller annotator pools, e.g., contractors or expert annotators.

Annotator Deboarding If certain annotators repeatedly deliver low quality work, it might be desirable to remove them from the annotator team. One way to find these annotators is via annotation noise (Hsueh et al., 2009), that is, the deviation of each annotator from the majority. Another is manual inspection by the dataset creators or more seasoned annotators. Spammers can also be detected during adjudication (§3.2.4), for instance by using MACE (Hovy et al., 2013). After deboarding annotators, it is recommended that their annotations are marked to be redone. Even though some platforms like Amazon Mechanical Turk make it possible to withhold payment, they should still be payed for the work already done unless there is compelling evidence for excessive fraudulent behavior.

Automatic Annotation Error Detection (and Correction) Instead of having human annotators manually inspect instances and search for errors, automatic approaches can be used. For some error types, it is possible to write checks that automatically find issues and sometimes even correct them (Květoň and Oliva, 2002; Qian et al., 2021). These checks can be simple rules which define wrong combinations of surface form and label and are derived from the data. For noisy text like Twitter data or crawled forum texts, spell checking might improve the underlying text before it is given to annotators. A more involved approach is annotation error detection, which leverages machine learning models to automatically find error candidates, which can then be given to annotators for manual inspection and an eventual correction (e.g., Dickinson and Meurers, 2003b; Northcutt et al., 2021a; Klie et al., 2023c). Annotation error detection is discussed in detail in Chapter 4.

3.2.4 Adjudication

In order to increase overall annotation reliability, commonly, more than one label per instance is collected (Bontcheva et al., 2014). These then usually need to be *adjudicated*, that is, finding a consensus, to create the final dataset, which has a single label per instance (Hovy and Lavid, 2010). For reproducibility, it is suggested to publish not only the adjudicated corpus but also raw annotations by the respective annotators. Learning from individual labels is also an option, especially in tasks with significant ambiguity and disagreement (Uma et al., 2021); then, no adjudication is used. The most common adjudication methods are described in the following.

Manual Adjudication To create a gold corpus, skilled annotators, often domain experts, manually inspect and curate each instance to a single label. While slow and expensive, this approach can yield high-quality data because ties can be broken and errors corrected during this inspection procedure. Curation can be sped up with automatic tooling, for instance, by automatically merging instances for which there is no disagreement or where the disagreement is below a certain threshold.

Majority Voting When using majority voting, given an instance rated by multiple annotators, its resulting label is the one that has been chosen most often. Instances without majority label can either be discarded or given to an additional annotator to break the tie. These are often experts but can also be (experienced) crowdworkers or contractors. In some works, supermajority voting is used, meaning that more than 50% of annotators need to agree, e.g., at most one differing label is allowed or even an unanimous vote is required. Lease (2011) notes that using majority voting might drown out valid minority voices and can reduce diversity, which should be taken into account.

Probabilistic Aggregation In majority voting, it is assumed that all annotators are equally reliable as well as skilled and errors are made uniformly random. This is not always the case in real annotation settings, especially in crowdsourcing. Annotators can be better or worse in certain aspects, might be biased, spamming, or even adversarial (Passonneau and Carpenter, 2014). To alleviate these issues, Dawid and Skene (1979) propose a probabilistic graphical model that associates a confusion matrix over label classes for each annotator, thereby modeling their proficiency and bias. The resulting aggregation is then based on weighing labels with the respective annotator's expertise for this label. An alternative formulation called *MACE* that also models spammers is given by Hovy et al. (2013).

It has been shown that using more sophisticated aggregation techniques can yield higher-quality gold standards (Passonneau and Carpenter, 2014; Paun et al., 2018; Simpson and Gurevych, 2019), but majority voting is often a strong baseline. The aforementioned works also discuss probabilistic aggregation in more detail.

3.3 Contributions

Quality management is an essential part of creating high-quality annotated datasets. Disseminating and analyzing quality management is especially relevant in the context of an increasing number of datasets being created and released. These datasets can then exhibit the aforementioned dangers of low-quality data collection. To better understand how annotation quality management is conducted in practice, as part of our publication

Analyzing Dataset Annotation Quality Management in the Wild (Chapter 7), we survey the relevant literature and annotate a large corpus of dataset-introducing publications for their quality management. From these, we derive recommendations for future dataset creators concerning which quality management exists and in which situations they can and should be applied. In summary, our contributions concerning annotation quality management are the following:

- Even though there exists an extensive body of work that discusses quality management in theory (see §3.2), we found that this knowledge is difficult to find and to consult. It is scattered across many different sources and usually treated as part of the general annotation process, hence often lacking depth. Therefore, we first survey the literature and summarize best practices regarding quality management for annotating datasets.
- To better understand how quality management is actually performed in practice, based on *Papers With Code*[4], we collect a large set of publications (591, of which 314 report human annotation or validation) that introduced new text dataset and annotate their quality management. Based on these annotations, we analyze how often and how well the different quality management methods were used. We find that a majority of the annotated publications apply good or very good quality management. However, we deem the effort of 30% of the works as only subpar. Our analysis also shows common errors, especially when using inter-annotator agreement and computing annotation error rates.
- We provide a list of recommendations (§7.6) that can be used by future dataset creators to improve the quality of their datasets and to avoid common pitfalls, for instance, how to best structure the annotation process, how to properly use agreement and how to compute the sample sizes when manually inspecting instances for their correctness.
- To foster further investigation into quality management for data annotation, we also release the code to collect and analyze the dataset and our annotations. Our dataset can also be used as a reference to find papers that use specific quality management methods, thereby serving as an example of how to apply them.

[4] `https://paperswithcode.com/datasets`

Chapter 4

Annotation Error Detection

One of the most impactful ways to estimate and improve annotation quality is to find erroneous or inconsistently labeled instances by manual inspection and then either correct them or filter them out (e.g., Barnes et al., 2019; Reiss et al., 2020; Northcutt et al., 2021b; Chen et al., 2022; Kreutzer et al., 2022). While effective, it is costly and time-consuming because it requires manual efforts and often employs expert labor. Therefore, many methods for automatic annotation error detection (AED) have been devised over the years. These methods enable dataset creators and machine learning practitioners to narrow down the instances that need manual inspection and —if necessary— correction (Dickinson, 2005). This reduces the overall work needed to find and fix annotation errors (see, e.g., Reiss et al., 2020).

AED has been used to discover that widely used benchmark datasets contain errors and inconsistencies (Northcutt et al., 2021b). Around 2% of the samples (sometimes even more than 5%) found to be incorrectly annotated in datasets like PENN TREEBANK (Dickinson and Meurers, 2003a), sentiment analysis datasets like SST, AMAZON REVIEWS, or IMDB (Barnes et al., 2019; Northcutt et al., 2021b), CONLL-2003 (Wang et al., 2019; Reiss et al., 2020), or relation extraction in TACRED (Alt et al., 2020; Stoica et al., 2021). AED has likewise been used to find inconsistently annotated instances, e.g., for part-of-speech (POS) annotation (Dickinson and Meurers, 2003a). Additionally, it has been shown that errors in automatically annotated (silver) corpora can also be found and fixed with the help of AED (Rehbein, 2014; Ménard and Mougeot, 2019).

In the following, we define the task of annotation error detection (AED) and present the most commonly used methods and metrics for their evaluation.

4.1 Task definition

Given an adjudicated dataset with one label per annotated instance, the goal of AED is to find instances labeled incorrectly or inconsistently. These instances can then be given to human annotators for manual inspection or used in annotation error correction methods. The definition of "instance" depends on the task and defines the granularity on which errors or inconsistencies are detected. For instance, when using AED for text classification, instances can be sentences; in POS tagging, instances can be tokens, and in named entity recognition, instances can be spans.

AED is typically used after a new dataset has been annotated and adjudicated. It is assumed that no already cleaned data or other data with the same annotation scheme is available.

We consider an instance labeled *incorrectly* if, according to the annotation guidelines, there is a unique, true label, but it is different from the current label of the instance. As an example for named entity recognition, `‘Obama’` is a PER and not an ORG. An instance is labeled *inconsistently* if its label implies that it belongs to one type, where it actually belongs to another, e.g., if `‘Manchester’` is used to refer to its football team, its label should be ORG and not LOC. An instance that is neither incorrect nor inconsistent is *correct*. Annotations can also be ambiguous; that is, at least two different labels are valid given the context. For example, in the sentence `‘They were visiting relatives’`, `‘visiting’` can either be a verb or an adjective. We discuss the impact of ambiguity on AED in §8.3.1.

Flaggers vs. scorers We divide automatic methods for AED into two categories which we dub *flaggers* and *scorers*. Flagging refers to methods that cast a binary judgment of whether the label assigned to an instance is correct or incorrect. Scoring methods estimate how likely it is that an annotation is incorrect. These correspond to classification and ranking from machine learning and information retrieval.

While flaggers are explicit about whether they consider an annotation incorrect, they do not indicate the likelihood of that decision. On the other hand, while scorers provide a likelihood, they require a threshold value to decide when an annotation is considered an error — for example, instances with a score above 80%. Those would then be given to human evaluators. Scorers can also be used in settings similar to active learning for error correction (Vlachos, 2006; Lin et al., 2016).

This distinction between flaggers and scorers regarding AED has not been made in previous work. However, it is critical to understand why different metrics need to be used when evaluating flaggers compared to scorers (see §8.5). Flaggers need to be evaluated with classification metrics like precision, recall, and F1 score. Scorers, however, require evaluation with ranking metrics like precision@k, recall@k or average precision.

4.2 Methods

Over the past three decades, many different methods have been developed for AED. Here, we group them by how they detect annotation errors and briefly describe each. It is a condensed summary of §8.3.2. Table 4.1 shows all methods discussed in this thesis (in alphabetical order).

4.2.1 Variation-based

Methods based on the variation principle leverage the observation that similar surface forms are often annotated with only one or, at most, a few distinct labels. If an instance

Abbr	Method Name	Proposed by
Flagger methods		
CL	Confident Learning	Northcutt et al. (2021a)
CS	Curriculum Spotter	Amiri et al. (2018)
DE	Diverse Ensemble	Loftsson (2009)
IRT	Item Response Theory	Rodriguez et al. (2021)
LA	Label Aggregation	Amiri et al. (2018)
LS	Leitner Spotter	Amiri et al. (2018)
PE	Projection Ensemble	Reiss et al. (2020)
RE	Retag	van Halteren (2000)
VN	Variation N-Grams	Dickinson et al. (2003a)
Scorer methods		
BC	Borda Count	Larson et al. (2020)
CU	Classification Uncertainty	Hendrycks et al. (2017)
DM	Data Map Confidence	Swayamdipta et al. (2020)
DU	Dropout Uncertainty	Amiri et al. (2018)
KNN	k-Nearest Neighbor Entropy	Grivas et al. (2020)
LE	Label Entropy	Hollenstein et al. (2016)
MD	Mean Distance	Larson et al. (2019)
PM	Prediction Margin	Dligach et al. (2011)
WD	Weighted Discrepancy	Hollenstein et al. (2016)

Table 4.1: Annotation error detection methods discussed in this thesis.

is annotated with a different, rarer label, it may be an annotation error or an inconsistency. Variation-based methods are relatively easy to implement and can be used in settings in which it is difficult to train a machine learning model, such as low-resource scenarios or tasks that are difficult to train models for, e.g., detecting lexical semantic units (Hollenstein et al., 2016). The main disadvantage of variation-based methods is that they need similar surface forms to perform well, which is not the case in settings like text classification or datasets with diverse instances.

Variation n-grams The most frequently used method of this kind is variation n-grams, which has been initially developed for POS tagging (Dickinson and Meurers, 2003a) and later extended to discontinuous constituents (Dickinson and Meurers, 2005), predicate-argument structures (Dickinson and Lee, 2008), dependency parsing (Boyd et al., 2008), and slot filling (Larson et al., 2020). For each instance, n-gram contexts of different sizes are collected and compared. It is considered incorrect if the label for an instance disagrees with labels from other instances with the same n-gram context.

Label Entropy and Weighted discrepancy Hollenstein et al. (2016) derive metrics from the surface form and label counts, which are then used as scorers. These are the entropy over the label count distribution per surface form or the weighted difference between the most and least frequent labels. They apply their methods to find possible annotation errors in datasets for multi-word expressions and super-sense tagging, which are then reviewed manually for tokens that are actual errors.

4.2.2 Model-Based

Probabilistic classifiers trained on the to-be-corrected dataset can be used to find annotation errors. Models in this context are usually trained via cross-validation (CV), and the respective holdout set is used to detect errors. The complete dataset is analyzed after all folds have been used as holdouts. Several ways have been devised for model-based AED, which are described below.

Re-tagging A simple way to use a trained model for AED is to use model predictions directly; when the predicted labels are different from the manually assigned ones, instances are flagged as annotation errors (van Halteren, 2000). Larson et al. (2020) apply this using a conditional random field (CRF) tagger to find errors in crowdsourced slot-filling annotations. Similarly, Amiri et al. (2018) use *Retag* for text classification. Yaghoub-Zadeh-Fard et al. (2019) train machine learning models to classify whether paraphrases contain errors and, if they do, what kind of error it is. To reduce the need to annotate instances repeatedly and adjudicate to achieve higher quality, Dligach and Palmer (2011) train a model on the labels given by an initial annotator. If the model disagrees with the instance's labeling, it is flagged for re-annotation. For cleaning dependency annotations in a Hindi treebank, Ambati et al. (2011) train a logistic regression classifier. If the prediction does not agree with the original annotation and the model confidence is above a predefined threshold, then the annotation is considered incorrect. *CrossWeigh* (Wang et al., 2019) is similar to *Retag* with repeated CV. During CV, *entity disjoint filtering* is used to force more model errors: instances are flagged as erroneous if the probability of their correct label falls below the respective threshold.

Classification Uncertainty Probabilistic classification models assign probabilities that are typically higher for correctly labeled instances compared to erroneous ones (Hendrycks and Gimpel, 2017). Therefore, the class probabilities of noisy labels can be used to score these for being an annotation error. Using model uncertainty is identical to using the network loss — which was, for example, used by Amiri et al. (2018) — because the cross-entropy function used to compute the loss is monotonic.

Prediction Margin Inspired by active learning (Settles, 2012), *prediction margin* (Dligach and Palmer, 2011) uses the probabilities of the two highest-scoring labels for an instance. The resulting score is their difference. The intuition behind this is that samples with smaller margins are more likely to be an annotation error since the smaller the decision margin is, the more unsure the model is. For multi-class classification, this could be generalized by using entropy.

Confident Learning This method estimates the joint distribution of noisy and true labels (Northcutt et al., 2021a). A threshold (the average self-confidence) is then learned, and instances whose computed probability of having the correct label is below the respective threshold are flagged as erroneous.

Dropout Uncertainty Amiri et al. (2018) use Monte Carlo dropout (Gal and Ghahramani, 2016) to estimate the uncertainty of an underlying model. There are different acquisition methods to compute uncertainty from the stochastic passes. A summary can be found in Shelmanov et al. (2021). The work of Amiri et al. (2018) uses the probability variance averaged over classes.

Label Aggregation Given a set of predictions obtained via Monte Carlo dropout, Amiri et al. (2018) use MACE (Hovy et al., 2013), an aggregation technique from crowdsourcing to adjudicate the resulting repeated predictions.

4.2.3 Training Dynamics

Methods based on training dynamics use information derived from how a model behaves during training and how predictions change throughout training.

Curriculum and Leitner Spotter Amiri et al. (2018) train a model via curriculum learning, where the network trains on easier instances during earlier epochs and is then gradually introduced to harder instances. Instances then are ranked by how difficult they are to learn for the model during training. They also adapt the ideas of the Zettelkasten (Ahrens, 2017) and Leitner queue networks (Leitner, 1974) to model training. There, difficult instances are presented more often during training than easier ones. The assumption behind both of these methods is that instances that are perceived as harder or misclassified more frequently are more often annotation errors than easier ones. These two methods require that the instances can be scheduled independently. This requirement does not hold for tasks like sequence labeling, as the model trains on complete sentences, not individual tokens or spans.

Data Map Confidence Swayamdipta et al. (2020) use the class probability for each instance's gold label across epochs to measure confidence. Low confidence correlates well with an item with an incorrect label in their experiments.

4.2.4 Vector Space Proximity

Approaches of this kind leverage dense embeddings of tokens, spans, and texts into a vector space and use their distribution therein. The distance of an instance to semantically similar instances is expected to be smaller than the distance to semantically different ones. Embeddings are typically obtained by using BERT-type models (Devlin et al., 2019) for tokens and spans or S-BERT (Reimers and Gurevych, 2019) for sentences.

Mean distance Larson et al. (2019) compute the centroid of each class by averaging vector embeddings of the respective instances. Items are then scored by the distance between their embedding vector and their centroid. The underlying assumption is that semantically similar items should have the same label and be close together (and thereby close to the centroid) in the vector space.

k-Nearest-Neighbor Entropy In the context of named entity recognition in clinical reports, Grivas et al. (2020) leverage the work of Khandelwal et al. (2020) regarding nearest-neighbor language models to find mislabeled named entities. First, all instances are embedded into a vector space. Then, each instance's k nearest neighbors according to their Euclidean distance are retrieved. Their distances to the instance embedding vector are then used to compute a distribution over labels by applying softmax. An instance's score is then the entropy of its distance distribution; if large, it indicates uncertainty, hinting at being mislabeled.

4.2.5 Ensembling

Ensemble methods combine the scores or predictions of several individual flaggers or scorers to obtain better a performance than each method would on its own.

Diverse Ensemble Instead of using a single prediction like *Retag* does, the predictions of several architecturally different models are aggregated. For instance, if the majority disagrees with the label, it will likely be an annotation error. Alt et al. (2020) use an ensemble of 49 different models to find annotation errors in the TACRED relation extraction corpus. In their setup, instances are ranked by how often a model suggests a label different from the original. Barnes et al. (2019) use three models to analyze error types on several sentiment analysis datasets; they flag instances where all models disagree with the gold label. Loftsson (2009); Angle et al. (2018) use an ensemble of different taggers to correct POS tags.

Projection Ensemble In order to correct the CONLL-2003 named entity corpus, Reiss et al. (2020) train 17 logistic regression models on different Gaussian projections of BERT embeddings. The aggregated predictions that disagreed with the dataset were then corrected by hand.

Item Response Theory Lord et al. (1968) developed *Item Response Theory* as a mathematical framework to model relationships between measured responses of test subjects (e.g., answers to questions in an exam) for an underlying, latent trait (e.g., the overall grasp on the subject that is tested). It can also be used to estimate the discriminative power of an item, i.e., how well the response to a question can be used to distinguish between subjects of different abilities. In the context of AED, test subjects are trained models; the observations are the predictions on the dataset, and the latent trait is task performance. Rodriguez et al. (2021) have shown that items that negatively discriminate — i.e., where a better response indicates being less skilled — correlate with annotation errors.

Borda Count Similarly to combining several flaggers into an ensemble, rankings obtained from different scorers can also be combined. For that, Dwork et al. (2001) propose to leverage Borda counts (de Borda, 1781), a voting scheme that assigns points based on their ranking. For each scorer, given scores for N instances, the instance that is ranked the highest is given N points, the second-highest $N-1$ and so on (Szpiro, 2010). The points assigned by different scorers are then summed up for each instance and form the aggregated ranking. Larson et al. (2019) use this to combine scores for runs of *Mean Distance* with different embeddings and show that this improves overall performance compared to only using individual scores.

4.2.6 Rule-based

Several works leverage rules that describe which annotations are valid and which are not. For example, to find errors in POS annotated corpora, Květoň and Oliva (2002) developed a set of conditions that tags have to fulfill in order to be valid, especially n-grams that are impossible based on the underlying lexical or morphological information of their respective surface forms. Rule-based approaches for AED can be very effective but are hand-tailored to the respective dataset, its domain, language, and task.

4.3 Contributions

While annotation error detection (AED) methods have been applied successfully in the past, several issues hinder their widespread use. First, there is no agreed-upon task definition and formalization, which causes many different ways to evaluate and apply AED methods. Second, new approaches for AED are often only evaluated on newly introduced datasets that are proprietary or not otherwise available (e.g., Dligach and Palmer, 2011; Amiri et al., 2018; Larson et al., 2019). Third, for most AED methods, there exists no publicly available implementation. Also, they rarely compare newly introduced methods to previous works or baselines. These issues make comparisons of AED methods very difficult. As a result, it is often unclear how well AED works in practice, especially which AED methods should be applied to which kind of data and underlying tasks. To alleviate these issues, as part of our publication Annotation Error Detection: Analyzing the Past and Present for a More Coherent Future (Chapter 8), we define a unified evaluation setup for AED, conduct a large-scale analysis of 18 AED methods, and apply them to 9 datasets for text classification, token labeling, and span labeling. The research reported in Chapter 8 addresses the aforementioned issues by providing the following contributions:

Evaluation methodology To unify its findings and establish comparability, we first define the task of annotation error detection (AED) and a standardized evaluation setup, including an improvement for evaluating span labeling in this context.

Easy to use reference implementations We survey past work from the last 25 years and implement the 18 most common and generally applicable AED methods. We publish our implementation in a Python package called NESSIE[1] which is easy to use, thoroughly tested, and extensible to new methods and tasks. Our package makes it significantly easier for researchers and practitioners to get started with AED.

[1] https://github.com/UKPLab/nessie

Benchmarking datasets We provide a common ground for benchmarking these approaches by collecting and generating 9 datasets for text classification, token labeling, as well as span labeling. We also publish the collected datasets to facilitate easy comparison and reproducibility.

Evaluation and analysis Using our implementation, we investigate several fundamental research questions regarding AED. We specifically focus on exploring how to achieve the best AED performance for each task and dataset, taking model calibration, usage of cross-validation, and model selection into account. Based on our results, we provide recipes and recommend how to best use AED in practice.

Takeaways Overall, the methods that worked best are *Classification Uncertainty*, *Confident Learning*, *Curriculum Spotter*, *Datamap Confidence*, *Diverse Ensemble*, *Label Aggregation*, *Leitner Spotter*, *Projection Ensemble*, and *Retag*; more complicated methods are not necessarily better. Model-based methods should be trained via cross-validation, otherwise the recall of downstream methods is heavily degraded (while the precision improves). Calibration can improve these model-based annotation error detection methods, but more research is needed to determine when exactly it can be useful.

Chapter 5

Annotation Efficiency

The creation of human-annotated datasets is often very time-consuming, expensive, and difficult (Ringger et al., 2008; Pustejovsky and Stubbs, 2013; Monarch, 2021). Annotation projects can take from months to even years. While often not explicitly reported, annotation costs per instance can range from cents to dollars. At first thought, this does not sound like much, but manually annotated dataset sizes typically range in the thousands or ten thousands of instances, and are often even be larger. Thereby, costs can quickly add up. For instance, He et al. (2018) report that for creating DUCHINSE, a machine reading comprehension dataset, it took 51,408 person-hours by about 800 crowdworkers and 52 experts to label 200,000 questions and write 400,000 answers. FitzGerald et al. (2018) collect annotations for Question-Answer driven Semantic Role Labeling and give a cost of \$43,647.33 for 265,000 questions. Ning et al. (2020) report a price of around \$15,000 while developing a dataset for reading comprehension benchmarking (30 700 instances); Dua et al. (2019) spent \$60,000 for a similar dataset (96,567 instances). NLVR2 (Suhr et al., 2019), a dataset for visual reasoning required \$19,132.99 in annotation costs. The Stanford Natural Language Inference (SNLI, 570k instances) Corpus (Bowman et al., 2015) cost around \$60,000 to create. Note that costs can vary greatly, depending on the dataset size, the number of labels per instance, the annotator's expertise and salary, or whether the annotation is outsourced or done in-house.

As annotated data is often an essential ingredient of training and evaluating machine learning models, it is very desirable to reduce annotation time as well as overall annotation costs while sustaining a target quality level. We subsume algorithms and methods to alleviate the data annotation with this under the term *annotation efficiency*.

Annotation efficiency can benefit machine learning practitioners in two ways. First, the same number of instances can be annotated for less effort. Second, with the same effort, larger datasets can be created.

5.1 Methods

The following section reviews the most commonly used methods to increase annotation efficiency. We only list methods that are applicable to projects that rely on manual annotations as part of their dataset creation, as we are interested in ways to reduce costs and improve quality when human annotators are involved. We exclude, for instance, methods to create datasets solely by indirect or self-supervision, whose improvements we leave for future work.

5.1.1 Task Setup

The overall task setup is comprised of the choice of annotation editor, task design and structure, as well as choice of annotators. These choices can strongly influence the annotation efficiency.

Annotation Editor The annotation editor —i.e., the user interface element annotators use to make annotations— is an essential ingredient towards a successful and efficient annotation process (Cerezo et al., 2021). Also, it determines and limits what and how it can be annotated.

Annotation editors can implement many features that support the annotators, thus increasing annotation efficiency. They can be designed to minimize the effort needed to annotate. This, for instance, can be done by reducing the cognitive load by presenting a streamlined user interface, reducing clicks needed, or being responsive to user inputs (Dandapat et al., 2009; Kummerfeld, 2019). Further speed-up can be achieved by providing keyboard shortcuts or macros for commonly executed actions like selecting labels or jumping to the next instance to annotate (Mikulová et al., 2022). More advanced annotation efficiency features like label suggestions via pre-annotation or recommenders (see §5.1.2) also require the editor to support these features.

Many annotation projects start with developing a new annotation editor (Chamberlain et al., 2013), which binds many resources. Reinventing the wheel also comes with the danger of repeating mistakes that have already been solved in other projects. For these reasons, using existing, general-purpose annotation tools like INCEpTION (Klie et al., 2018) is usually recommended if applicable.

Task Design Oftentimes, it can be more efficient to split a complex task into several, more manageable phases (Verhagen, 2010), which are then annotated separately. For example, for relation extraction, the first step can be marking spans, then labeling their relation later. When annotating treebanks, this can be annotating the different layers sequentially instead of annotating all layers at the same time (Zeldes, 2017). Kulkarni et al. (2012) even propose a system where crowdworkers themselves can split their assignments into subtasks and later consolidate them. Splitting can ease tasks and thus reduce costs but also might remove context, thereby potentially reducing quality.

Annotator Selection As previously mentioned in §2.2 and §3.2.1, the choice of annotators can significantly impact annotation time, cost, and quality. Which kind of annotators to employ depends, among others, on the task difficulty, availability, target language, and whether particular expertise is needed. If the annotation task is solvable by crowdworkers, this is often an efficient way to annotate (Snow et al., 2008). However, using crowdworkers usually also involves repeated annotations per instance that need to be adjudicated (see §3.2.4), which increases costs (Hovy et al., 2014). Trained contractors can be an alternative to hiring crowdsourcing or domain experts (Chen et al., 2021) and, in many cases, even be cheaper, as fewer repetitions and less cleanup are needed. While it is desirable to reduce costs, it is essential to pay and treat annotators well and thereby build a pool of trusted annotations. Minimizing the annotation costs via saving on the salary of annotators comes with ethical issues (Fort et al., 2011; Kummerfeld, 2021) that need to be taken into account.

It is also possible to leverage annotators of diverse backgrounds, e.g., give more straightforward instances to crowdworkers and let experts validate. For biomedical relation extraction, Yang et al. (2019) have shown that estimating annotation difficulty and routing more difficult instances to experts yields better results than solely relying on crowdsourcing. Alonso and Romeo (2014) let crowdworkers annotate in the first phase and only route instances with disagreement to experts afterward, thus reducing costs.

5.1.2 Label Suggestions

A line of research that explicitly aims to increase annotation efficiency are label suggestions. Label suggestions are automatically generated, potential annotations that are presented to human annotators during annotation. These low-cost, quick-to-create but unreliable annotations are then given to human annotators for subsequent review and, if necessary, correction. Many automatic annotations are correct and therefore require no manual correction. Accepting suggestions instead of creating new annotations can also be mentally less taxing. Therefore, when using label suggestions, potentially, annotation time can be saved. Annotations can for instance be generated by dictionaries/gazetteers (e.g., Savary et al., 2010; Simon et al., 2015), pre-trained machine learning models (e.g., Schulz et al., 2019; Beck et al., 2021) or rules (e.g., Ratner et al., 2017; Névéol et al., 2011; Mikulová et al., 2022).

Several aspects must be considered when using label suggestions for an annotation project. First, they need to be helpful to annotators; that is, have sufficient precision and not be distracting (Greinacher and Horn, 2018). Second, it needs to be taken into account whether annotators can reliably differentiate between correct or incorrect annotations, which is not always the case, especially in more complex or domain-specific annotation tasks (Fort and Sagot, 2010). Third, label suggestions might introduce model bias into the resulting dataset, for example, if annotators accept the suggestions as is to reduce their effort. Bias here means whether label suggestions increase the discrepancy between annotations and a final, aggregated gold dataset (Lingren et al., 2014).

We differentiate between two different kinds of label suggestions based on whether they are added before annotation starts (pre-annotation) or while annotating (recommenders). Works that used either kind are summarized together with their impact on annotation efficiency in Table 5.1.

Pre-Annotation This describes the practice of automatically annotating a dataset with silver quality, i.e., potentially erroneous, annotations before giving them to annotators for correction. For pre-annotation, label suggestions are only generated before the annotation project takes place and are static afterward; annotators can either leave them as correct, delete them, or correct them. Many datasets have been created while using pre-annotation, of which we mention several in this section. More works that used pre-annotations paired with their impact on annotation quality and efficiency are given in Table 5.1.

A famous example of using pre-annotations is the creation of PENN TREEBANK (Marcus et al., 1993), which leveraged a POS tagger trained on the BROWN CORPUS (Francis and Kucera, 1979). Compared to manual tagging, pre-annotation halved the annotation time and the annotator disagreement rate. Fort and Sagot (2010) further analyze their setup by manually re-annotating parts of the PENN TREEBANK using different taggers and annotation configurations. They confirm the initial findings of Marcus et al. (1993) and further note that even pre-annotations generated by not-so-accurate taggers already provide a reduction in annotation time. In addition, they conclude that the better a model is, the better the resulting inter-annotator agreement after correction and the lower the overall annotation time. Their findings agree with the observations of Dandapat et al. (2009), who analyzed a similar setting for POS labeling in Bangla and Hindi. For frame-semantic argument structure annotation, Rehbein et al. (2009) observe an increase in annotation quality but no impact on annotation time when using pre-annotation compared to annotation without.

Concerning how annotators react to pre-annotations, Dandapat et al. (2009) interviewed annotators after letting them annotate documents with and without pre-annotations; they reported that they felt less distracted and could focus more on the task without pre-annotations. Rosset et al. (2013) found that annotators of all experience levels liked pre-annotations, even in some cases where the evaluation showed for some annotators that they had no impact on time or quality.

Recommenders An extension of the static pre-annotations are *recommenders* (Ganchev et al., 2007). These provide annotation suggestions that are generated during annotation. Suggestions created that way are displayed to the user alongside already-made annotations. The user may accept a suggestion, which turns the suggestion into a proper annotation, which can then be further edited if desired. The user may also reject the suggestion to hide it and prevent the recommender from suggesting it again. The difference between pre-annotation and recommenders is that no recommender suggestion is accepted without validation, as it requires an additional action to accept. In contrast, pre-annotations can be left as is to create new annotations which can cause more missed errors. Also, the default action for incorrect suggestions does not require any work for users, while deleting annotations for pre-annotation does. Moreover, instead of only presenting the best option, it is easier for recommenders to display top-k suggestions, e.g., based on model confidence, thereby increasing the chance that a correct recommendation is given.

Recommenders can be either static or interactive. Static ones use, for instance, pre-trained models or dictionaries that are created ahead of time; they do not incorporate feedback during annotation. Interactive recommenders can learn from user feedback and improve over time, e.g., by re-training on newly made annotations.

> The advantage of interactive recommenders is that even if no in-domain data exists, a model can be trained on the annotations made so far. Also, issues caused by concept drifts or domain differences can be reduced as models are trained on the same domain they are asked to predict. Interactive recommenders are especially interesting when there is insufficient training data to train a good-enough performing model for pre-annotation. Then, the recommender model can be trained once enough annotations are made, and it can be interactively improved throughout the annotation project. Even new classes can be added to the recommender while an annotation project is underway, which would be difficult to achieve with pre-annotations. Models can also be trained globally, that is, on annotations of all users, or models can be personalized. In the latter case, this means that a model used as a recommender for an annotator is only ever trained on that individual's annotations.
>
> A disadvantage of interactive recommenders is that re-training on incoming annotations can be expensive based on the underlying model architecture. Probabilistic machine learning models like conditional random fields, logistic regression, or gradient-boosted machines can often be made fast enough to re-train after every annotation is made (Klie et al., 2020; Lee et al., 2022). However, re-training for neural models can take minutes or hours and often requires a GPU. Therefore, updates are usually staggered and only made after a certain number of new annotations are made (Schulz et al., 2019; Beck et al., 2021). Alternatively, models can be used that support online or continuous learning, thereby only requiring training time related to the number of new annotations made (e.g., Yimam et al., 2016).
>
> Several works have leveraged recommenders for their annotation project. Brants et al. (2002) are an early user of interactive recommenders to create the TIGER Treebank; they did not compare using recommenders to annotating without them. Yimam et al. (2016) use interactive recommenders for biomedical entity and relation annotation. Schulz et al. (2019) find that even for the task of segmentation and classification of epistemic activities in diagnostic reasoning texts, which is difficult even for experts, the use of recommenders can reduce annotation time and increase agreement. Beck et al. (2021) investigate annotation suggestions for opinion mining in German Covid-19 social media texts. They find that while overall, annotation suggestions substantially improved agreement and quality, interactive recommenders did not yield improvements over static ones. Felt et al. (2014) report statistically significant improvements of annotation time and quality if recommender quality is above 70% accuracy.
>
> Most works report a reduction in annotation time and observe no or only minor introduction of bias compared to annotating without recommender support. More examples are given in Table 5.1.

To summarize, for the practical use of label suggestions, annotation time is usually reduced (see Table 5.1). However, if the suggestion quality is too low, annotation time can even increase (Ogren et al., 2008). In some cases, consistency and quality also improves (e.g., Rehbein et al., 2009; Mikulová et al., 2022).

Concerning bias, Fort et al. (2009) report that in their experiments for annotating named entities, annotators often tended to leave pre-annotations as is, even if they were wrong. Dandapat et al. (2009) also found that pre-annotations introduce bias; the more experienced the annotator, the less pronounced the influence is. In contrast to that, most works that analyzed bias introduced by label suggestions found no bias when using pre-annotations for named entity annotation (e.g., Rosset et al., 2013; Lingren et al., 2014; Schulz et al., 2019). While finding no particular bias in their experiments, Beck et al. (2021) note that interactive recommenders should be used with care in non-expert annotation settings due to the dangers of amplifying biases during re-training.

By	What	Domain	Annotators	Q	T	A	B
Pre-annotation							
Marcus et al. (1993)	Treebank	English newswire	Expert	↗	↘	↗	✗
Chiou et al. (2001)	Treebank	Chinese newswire	Expert	→	↘	·	✓
Xue et al. (2002)	POS tagging	Contemporary Chinese	?	→	↘	·	·
Tanaka et al. (2005)	Treebank	Japanese dictionary entries	Novice	·	↘	·	·
Chou et al. (2006)	PropBank	MEDLINE abstracts	?	·	↘	·	·
Ogren et al. (2008)	Named entities	Clinical notes	Expert	·	↗	↗	✓
Dandapat et al. (2009)	POS tagging	Bangla/Hindi sentences	Novice	·	↘	→	✓
			Expert	·	↘	↗	✗
Rehbein et al. (2009)	Frame labeling	FrameNet 1.3	Expert	↗	→	·	✗
Fort and Sagot (2010)	POS tagging	English newswire	Expert	↗	↘	↗	✓
Meurs et al. (2011)	Text mining	PubMed	Expert	·	↘	·	·
Skjærholt (2011)	POS+Morph. Analysis	Old Latin	Novice	↗	↘	·	·
			Expert	↗	→	·	·
Rosset et al. (2013)	Named entities	Contemporary French	Novice+Expert	↗	↘	↗	·
Felt et al. (2014)	Morphological analysis	Classical Syriac	Expert	↗	↘	·	·
Lingren et al. (2014)	Named entities	Clinical Trial Reports	Expert	→	↘	→	✗
Eckhoff et al. (2016)	Dependency parsing	Old East Slavic texts	Novice+Expert	→	↘	·	·
Lu et al. (2016)	Named entities	English&Chinese newswire	Expert	→	↘	·	·
Recommender							
Ganchev et al. (2007)	Named entities	MEDLINE abstracts	Expert	·	↘	·	·
Ulinski et al. (2016)	Dependency parsing	Video descriptions	Novice+Expert	↘	↘	·	·
Yimam et al. (2016)	Entity linking	MEDLINE abstract	Expert	·	↘	·	·
Greinacher and Horn (2018)	Named entities	German newswire	Novice	↗	↘	·	·
Schulz et al. (2019)	Epistemic activities	Diagnostic reasoning	Expert	↗	↘	·	✗
Klie et al. (2020)	Entity linking	Old English	Expert	·	↘	·	·
Beck et al. (2021)	Opinion mining	German Tweets	Novice	↗	·	↗	✗

Table 5.1: Overview of works levering label suggestions and its influence on **Q**uality, Annotation **T**ime, **A**greement, **B**ias. We differentiate between *increase* ↗, *decrease* ↘ and *no change* → of the respective metric. In case bias was analyzed, if some bias observed, we indicate it as ✓, if not observed, as ✗. If an aspect was not analyzed by a work, then we put " · ".

5.1.3 Efficient Data Selection

To better focus project resources on the annotation of instances that are most relevant, interesting, or beneficial, several methods have been developed to select which data to annotate, in which order, and whom to assign instances for annotation.

Active Learning Active learning is a widely adopted approach to select instances for annotation that are most beneficial to the performance of downstream trained machine learning models (Settles, 2012). From an annotated seed dataset, the target machine learning model is trained. Then, from a large pool of unlabeled data, instances are sampled by ranking them according to heuristics, e.g., uncertainty sampling (Lewis and Catlett, 1994). There, instances are scored highest for which the model is most uncertain when asked to classify them. These are then given for labeling to human annotators. The model is then trained on the just annotated instances and is again used to rank instances. This training, selection, and annotation loop is repeated until the model reaches sufficient task performance or the annotation budget has been used up. Active learning has been shown to reduce the number of annotated instances needed to reach a quality threshold compared to random sampling, thereby saving costs and time (Fang et al., 2014).

Active learning is model-centric, aiming to obtain a well-performing machine-learning model. This goal can negatively impact users, as the model potentially selects instances that are uninteresting or irrelevant to users, too hard or too easy (Lee et al., 2020). Lowell et al. (2019) find that the resulting dataset is tied to the model; training a different model type on the same data often does not improve performance compared to random sampling. Another issue can be that models tend to acquire instances that are then hard to learn or even require external knowledge, thereby degrading the resulting dataset quality (Karamcheti et al., 2021). Also, active learning does not guarantee that the resulting dataset is representative; MacKay (1992); Dasgupta and Hsu (2008) show that corpora annotated that way often do not follow the population distribution. Biased datasets have many issues (Bender and Friedman, 2018); therefore, caution must be exercised when using active learning in the settings targeted by this thesis.

Adversarial Dataset Construction A different approach to data selection but in the same vein as active learning is *adversarial dataset construction*, which is most often used for creating challenging datasets, especially suitable for benchmarks, requiring text production, e.g., question answering. Instances are vetted by a machine learning model trained in the loop on the annotated data. When users submit annotations, they are given to the model to predict; if the model classifies them correctly, they are discarded or given back to annotators to be altered. Examples of datasets created this way are, among others, DROP (Dua et al., 2019) or ADVERSARIALNLI (Nie et al., 2020). However, having a roadblock before submission might be exhausting for annotators as they need to iterate on annotations instead of being allowed to go on. A model serving as the gatekeeper helps increase efficiency because annotators are nudged to create difficult annotations. Compared to the alternative of best-effort guidelines to enforce the creation of tricky instances and manual filtering, adversarial dataset construction can save time and effort.

Annotation Curricula It is not only important what to annotate, but it can also have an impact at which point in the annotation process which annotations are given out. An annotation process is stable if its output does not drift over time (see §3.1). Instability can, for example, occur due to carelessness, distractions, tiredness, or even learning through practice. In order to stabilize the annotation process and provide implicit on-the-annotation-job training, in Lee, Klie et al. (2022), we propose *Annotation Curricula*, which brings the ideas and methods of curriculum learning (Bengio et al., 2009) to the annotation process. A learning curriculum orders exercises to match a learner's proficiency level (Krahnke and Krashen, 1983), i.e., not stressing them too little or too much, thereby giving a targeted, optimal learning stimulus. This is also called the zone of proximal development (Vygotsky, 1980). Adapted to annotation, instances are ordered by a heuristic, for example, annotation time as a proxy for difficulty. Thereby, annotators can implicitly get used to the task and deal with edge cases as well as difficult instances later once they are more experienced. In simulation and a user study, annotation curricula have shown statistically significant reductions in annotation time while preserving annotation quality.

5.1.4 Community Annotation

To reduce costs when creating new datasets, several works have relied on the community's generosity to annotate for free (Uzuner et al., 2010). In addition to the reduction in cost, it also has the potential upside of attracting intrinsically (not only fiscally) motivated volunteers. These are often skilled in the task and can provide high-quality annotations, thus potentially combining the advantages of expert annotations and crowdsourcing.

However, relying on unpaid annotators also entails several issues. First, attracting volunteers can be difficult and effortful; it is not guaranteed that a sufficient number of annotators will participate. Second, there are ethical considerations that need to be taken into account when working with volunteers for unpaid work (Resnik et al., 2015; Rasmussen and Cooper, 2019). First, it is vital to consider an ethical deployment that does not compromise the participants' trust. This ensures that participants are not exploited for "free labor"—in contrast to approaches like reCAPTCHA (von Ahn et al., 2008), where humans are asked to solve a task in order to gain access to services. Whereas CAPTCHAs were initially intended to block malicious bots, they are becoming increasingly problematic due to their deployment and use by monopolizing companies, which raises ethical concerns (Avanesi and Teurlings, 2022). Second, given increasing concerns regarding the ownership and use of collected data (Arrieta-Ibarra et al., 2018), one should grant participants full rights to access, change, delete, and share their personal data (Jones and Tonetti, 2020). Third, community annotation is almost exclusively applicable for creating research datasets and much less for industry applications.

There are two flavors of community annotation that we consider in this thesis, both of which we briefly describe in the following.

Games with a Purpose A fun way to collect annotations from volunteers is *games with a purpose*, i.e., devising a game in which participants annotate data (Chamberlain et al., 2008; Venhuizen et al., 2013). It has been shown that if a task lends itself to being gamified, it can attract a wide audience of participants and be used to create large-scale datasets (von Ahn, 2006). Several works propose games for different purposes and languages, for instance, anaphora annotation (PhraseDetectives, Poesio et al. 2013), dependency syntax annotation (Zombilingo, Fort et al. 2014), or collecting idioms (Eryiğit et al., 2022). Lyding et al. (2022) investigate games with a purpose in the context of (second) language learning to crowdsource annotations from learners and simultaneously teachers. Another example is Substituto, a turn-based, teacher-moderated game for learning verb-particle constructions (Araneta et al., 2020).

Citizen Science Citizen Science broadly describes the participation and collaboration of the general public (the citizens) with researchers to conduct science (Haklay et al., 2021). It is a popular alternative approach for dataset collection efforts and has been successfully applied in cases of, for example, weather observation (Leeper et al., 2015), counting butterflies (Holmes, 1991) or birds (National Audubon Society, 2020), classifying images of galaxies (Lintott et al., 2008) or monitoring water quality (Addy et al., 2010). Compared to crowdsourcing, citizen science participants are volunteers who do not work for monetary gain. Instead, they are often motivated intrinsically. They may have a personal interest in positively impacting the environment (West et al., 2021), or in altruism (Rotman et al., 2012). Intrinsic motivation also has the potential of resulting in higher-quality annotations compared to crowdsourcing. For instance, Lee et al. (2022) find in their evaluation study with volunteers that their participants may have been willing to take more time annotating for the sake of higher annotation accuracy. Tsueng et al. (2016) directly compare crowdsourcing with citizen science and show that volunteers can achieve similar performance in mining medical entities in scientific texts.

Newly emerging technologies and platforms further allow researchers to conduct increasingly innovative citizen science projects, such as the prediction of influenza-like outbreaks (Lee et al., 2021) or the classification of animals from the Serengeti National Park (Swanson et al., 2015). For projects dealing with natural language, *LanguageARC* is a citizen science platform for developing language resources (Fiumara et al., 2020). One work using LanguageARC is by Fort et al. (2022), who collected resources to evaluate bias in language models.

5.2 Contributions

Dataset creation is difficult, expensive, and time-consuming. Therefore, reducing annotation time and supporting annotators throughout the process is highly beneficial and desirable. This thesis contributes to improving annotation efficiency by providing the following contributions:

- As part of Klie et al. (2018) (Chapter 6), we implemented recommender support for the INCEpTION annotation platform. INCEpTION being a general purpose annotation platform that is easily extensible allowed a smooth and simple integration. There are two types of recommenders supported: internal and external. Internal recommenders are directly integrated into the platform by implementing a Java interface. External recommenders use a simple, HTTP-based protocol for which we provide a Python library.[1] Recommender support in INCEpTION has been used extensively in other works, for example, by Schulz et al. (2019) or Beck et al. (2021).

[1] https://github.com/inception-project/inception-external-recommender

In addition, the Python library CASSIS[2] has been developed to convert from and to the INCEpTION internal data format, for instance, to facilitate pre-annotations.

- In Klie et al. (2020) (Chapter 9), we propose a novel, interactive approach to entity linking for low-resourced domains. For this, we combine recommenders that suggest potential concepts and adaptive candidate ranking, thereby speeding up the overall annotation process and making it less tedious for users. To validate our approach, we conducted a user study. There, annotation speed improves by up to 35% compared to annotating without interactive support; users report that they strongly prefer our system.
- In Lee, Klie et al. (2022) (Chapter 10), as part of a joint work with Ji-Ung Lee, we propose *annotation curricula* to implicitly teach the annotation task by smartly ordering instances. We together developed the idea and formalization. The individual contribution of this thesis toward annotation curricula is using machine learning models to estimate annotation time as a proxy for difficulty. We show in simulation that models can be trained interactively to estimate annotation time as a proxy for difficulty based on the annotation time for annotations made so far. In a user study, Ji-Ung showed that annotation curricula can significantly reduce annotation time compared to a random ordering.
- In Klie et al. (2023b) (Chapter 11), we provide a systematic study on citizen science for annotating natural language processing (NLP) datasets. For this, we re-annotate parts of the PERSPECTRUM (Chen et al., 2019) dataset using citizen science and compare these to the original, crowdsourced annotations. We provide guidelines and recommendations on how to best conduct a citizen science project for NLP annotation and discuss critical legal and ethical aspects. Our results show that using citizen science for such annotation projects can result in high-quality annotations but that attracting and motivating people is critical for its success, especially in the long term. We thus conclude that citizen science projects have the potential to be applied to NLP annotation if they are conceptualized well but are best suited for creating smaller datasets.

[2] https://github.com/dkpro/dkpro-cassis

Part II

Publications

Chapter 6

The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation

The INCEpTION Platform:
Machine-Assisted and Knowledge-Oriented Interactive Annotation

Jan-Christoph Klie **Michael Bugert** **Beto Boullosa**
Richard Eckart de Castilho **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt, Germany
`https://www.ukp.tu-darmstadt.de`

Abstract

We introduce INCEpTION, a new annotation platform for tasks including interactive and semantic annotation (e.g., concept linking, fact linking, knowledge base population, semantic frame annotation). These tasks are very time consuming and demanding for annotators, especially when knowledge bases are used. We address these issues by developing an annotation platform that incorporates machine learning capabilities which actively assist and guide annotators. The platform is both generic and modular. It targets a range of research domains in need of semantic annotation, such as digital humanities, bioinformatics, or linguistics. INCEpTION is publicly available as open-source software.[1]

1 Introduction

Due to the success of natural language processing (NLP), there is a large interest to apply NLP methods in a wide range of new application domains, for instance to scale textual data analysis or to explore textual data. This requires being able to quickly bootstrap new annotated corpora in these domains. As target users, we consider for instance data scientists who train and evaluate machine learning algorithms as well as researchers who to wish to cross-reference and disambiguate text collections for better exploration and discovery. Furthermore, every application domain uses specific semantics and vocabularies which need to be modeled, making entity linking one of the most important annotation tasks. Thus, we identify three requirements that annotation tools must meet in order to address today's demands:

Annotation assistance. Creating annotated corpora is challenging and requires experts who are highly familiar with the annotation schemes in order to reach high inter-annotator agreement as well as high quality annotations. For semantic annotations, it is even more difficult: tasks such as entity and fact linking are very time intensive and often require an in-depth familiarity with the inventory of the resource. To improve the efficiency of these tasks, it is necessary to create an environment in which the computer can learn from the human and use this knowledge to support the human annotator.

Knowledge management. Semantic resources for new domains typically do not exist from the start. Instead, they are constructed and expanded as part of the annotation task. Thus, while some annotation tools already support entity linking against existing large-scale general knowledge bases (KB) such as Wikidata or DBPedia, it is also necessary that domain specific knowledge can be collected and modeled directly in the annotation tool.

Customizability and extensibility. Every annotation project has specific requirements that go beyond the basic task requirements, e.g. due to the data formats, knowledge resources, or text genres involved. Therefore, it is important that the tool can be customized, extended, and adapted to novel tasks.

INCEpTION addresses these requirements in several ways. To improve the efficiency of (semantic) annotation tasks, so-called *recommenders* are implemented which provide users with suggestions for possible labels. To navigate the annotation suggestions, an *active learning mode* can be enabled which

[1] `https://inception-project.github.io` ; software is licensed under the Apache License 2.0

Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pages 5–9
Santa Fe, New Mexico, USA, August 20-26, 2018.

guides the annotator in an efficient and effective manner. *Knowledge management* is fully integrated; knowledge bases can be created and edited, *entity and fact linking* is supported. The *modular architecture* of INCEpTION enables users to augment their instance with custom machine learning algorithms, data formats, knowledge bases, annotation types, visualizations and more.

2 Related Work

In recent years, several knowledge management and annotation tools have been developed, but none of them offer an integrated environment addressing all of the mentioned requirements.

Several tools, e.g. GATE Teamware (Bontcheva et al., 2013) implement support for automatically pre-annotating text. These are then corrected by the annotator in the next step. In contrast to that, INCEpTION allows recommenders to give suggestions at any time during the annotation process and learns from the user interactions (new annotations, rejections, etc.).

WebAnno (Yimam et al., 2014) integrates an automation mode in which the system can learn from annotations made by the user and provide suggestions. However, retraining has to be triggered manually by an administrator. Also, it uses a non-modular backend that provides only one machine learning algorithm and does not support active learning. WebAnno presents the document to be annotated and the recommended annotations separately in a split-screen mode which makes it tedious to relate recommendations to already existing annotations.

The general approach described by Emanuele Pianta and Zanoli (2008) who integrate an active learning process with an existing annotation tool and the ability to call out to different machine learning backends for recommendations as well as *Prodi.gy*[2] are similar to our approach. However, they focus strongly on the active learning aspect and force the user to follow the lead of the active learning module, restricting the user's workflow. In INCEpTION, the active learning algorithm highlights a particular recommendation to be judged by the user, but does not prevent the user from making other annotations.

The web-based tool AlvisAE (Papazian et al., 2012) supports both linking entity mentions to a knowledge base and editing knowledge bases (with limitations), but it does not support recommendations or active learning. Knowtator (Ogren, 2006) is another instance of a desktop application which ships as an annotation plugin for an ontology building tool. However, single-user tools like the ones above do not meet today's demand for collaboration-oriented annotation tools.

3 INCEpTION – System Overview

INCEpTION offers a number of functionalities expected from a generic annotation platform: a versatile and yet intuitive user interface, flexible configuration of the annotation schema, the ability to run multiple annotation projects concurrently for multiple users and workflow-support with annotation and adjudication stages, etc. With respect to these basic functionalities, we build on our previous work in the context of WebAnno (Yimam et al., 2014) and UIMA (Ferrucci et al., 2009), and therefore refer the interested readers to these projects.

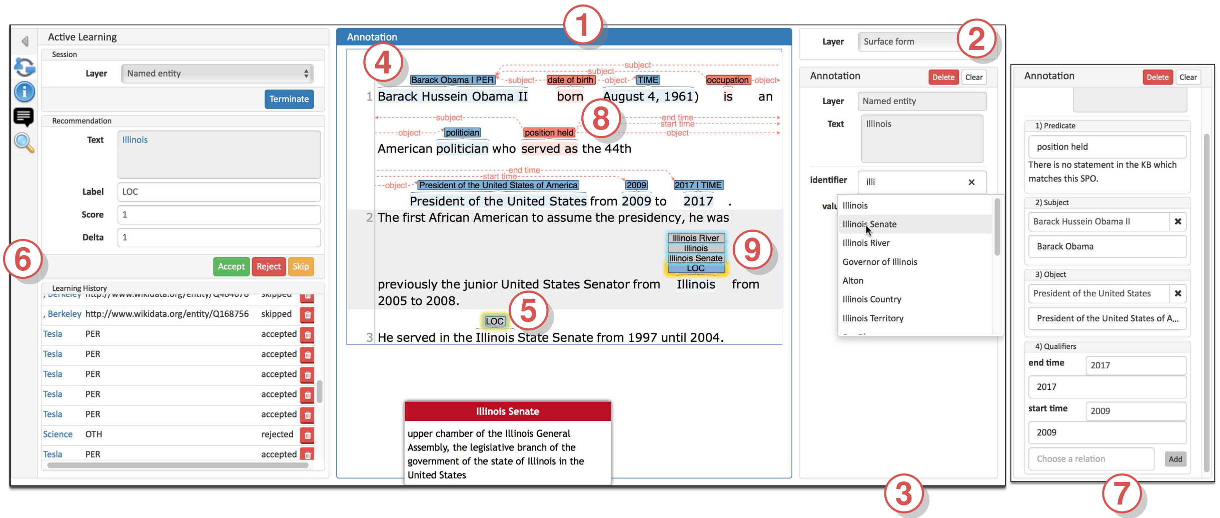
In this paper, we focus specifically on INCEpTION's unique features, in particular on annotation assistance via *recommenders* and *active learning*, the *knowledge management capabilities* and its options for *customizations and extensions*.

Annotation User Interface The annotation scheme used by INCEpTION organizes annotations into layers which define the set of attributes that an annotation may carry. Users can define an arbitrary number of layers that are each either spans or relations between spans. Each layer can have an arbitrary number of features which can be strings, numbers booleans, concept references, or references to other annotations.

The annotation user interface (Figure 1) displays the document text in the central part ①. Marking a span of text here creates a new annotation on the layer that is selected in the right sidebar ② (e.g. named entity). Span annotations are displayed as bubbles above the text.

When an annotation is created or an existing annotation is selected, its features are shown in the right sidebar and can be edited there ③④. Depending on the feature type, a specialized editor is shown. For

[2] `https://prodi.gy`



Active Learning

Session

Layer: Named entity

Recommendation

Text: Illinois

Label: LOC

Score: 1

Delta: 1

Learning History

, Berkeley	http://www.wikidata.org/entity/Q484078	skipped	
, Berkeley	http://www.wikidata.org/entity/Q168756	skipped	
Tesla	PER		accepted
Tesla	PER		accepted
Tesla	PER		accepted
Tesla	PER		accepted
Tesla	PER		accepted
Science	OTH		rejected
Tesla	PER		accepted

Annotation

1 Barack Hussein Obama II born August 4, 1961) is an American politician who served as the 44th President of the United States from 2009 to 2017 .

2 The first African American to assume the presidency, he was previously the junior United States Senator from Illinois from 2005 to 2008.

3 He served in the Illinois State Senate from 1997 until 2004.

Illinois Senate

upper chamber of the Illinois General Assembly, the legislative branch of the government of the state of Illinois in the United States

Layer: Surface form

Annotation

Layer: Named entity

Text: Illinois

identifier: illi

valu

Illinois
Illinois Senate
Illinois River
Governor of Illinois
Alton
Illinois Country
Illinois Territory

Annotation

1) Predicate

position held

There is no statement in the KB which matches this SPO.

2) Subject

Barack Hussein Obama II

Barack Obama

3) Object

President of the United States

President of the United States of A...

4) Qualifiers

end time: 2017

2017

start time: 2009

2009

Choose a relation

Figure 1: INCEpTION annotation editor: ① annotation area, ② annotation layer selection, ③ entity linking feature editor, ④ named entity linked to Wikidata, ⑤ entity mention suggestion, ⑥ active learning sidebar, ⑦ fact linking editor, ⑧ annotated fact, ⑨ entity linking recommendations.

instance, the editor to assign concepts from a knowledge base is an auto-complete input field which shows entities from the knowledge base that match the users' input. The left sidebar provides access to further functionalities, in particular to the active learning mode.

Recommenders To improve annotation efficiency, INCEpTION offers *recommenders*. These are algorithms that make use of machine learning and/or knowledge resources to provide annotation suggestions; they are displayed to the user alongside already made annotations in a different color ⑤. The user may accept a suggestion by clicking on it. This turns the suggestion into a proper annotation which can then be further edited if desired. The user may also reject the suggestion by double-clicking on it.

The recommender subsystem is designed to continuously monitor the users' actions, to update/retrain the recommendation models, and to provide always up-to-date suggestions. Multiple recommenders can be used simultaneously, e.g. high-precision/low-recall recommenders (e.g. using a dynamic dictionary) which are useful during early annotation stages, and context sensitive recall-oriented classifiers (e.g. sequence classifiers) for later stages. To avoid classifiers providing too many wrong suggestions during bootstrapping, a quality threshold can be configured per recommender.

INCEpTION supports two types of recommenders: internal and external. Internal recommenders are directly integrated into the platform by implementing a Java interface, while external recommenders use a simple, HTTP-based protocol to exchange UIMA CAS XMI (a XML representation of UIMA annotations). External recommenders allow users to leverage already existing and pre-trained machine learning models or libraries from other programming languages.

Active learning The goal of active learning (AL) is to quickly reach a good quality of annotation suggestions by soliciting feedback from the user that is expected to be most informative to the underlying machine learning algorithm. Presently, we use the uncertainty sampling strategy (Lewis and Gale, 1994) to drive the AL as it only requires that the recommenders produce a confidence score for each suggestion. The AL mode ⑥ works for one layer at a time to avoid confusion. After the layer has been selected, the system highlights the suggestion it seeks input for in the annotation area and displays its details in the AL sidebar. The user can then accept, reject or skip the suggestion. Skipped suggestions are presented again to the user when there are no more suggestions to accept or reject. The choices are stored in the learning history where the user can review and undo them if necessary. When the AL mode is enabled, the user can still deviate from its guidance and arbitrarily create and modify annotations. All changes made through the AL sidebar or in the main editor are immediately picked up by the recommenders causing the suggestions as well as the AL guidance to be updated.

Knowledge Management For knowledge management, INCEpTION supports RDF-based knowledge bases. While internal KBs can be used for small domain-specific knowledge, large external (remote) knowledge bases can be accessed via SPARQL. A flexible configuration mechanism is used to support different knowledge representations, such as Wikidata, DBPedia, OWL, CIDOC-CRM, SKOS, etc. INCEpTION has notions of classes, instances, properties and qualifiers (for KBs using reification). However, it does not aspire to offer full support for advanced features of schemes such as OWL.

Knowledge bases enable the user to perform knowledge-driven annotations, e.g. annotating mentions of knowledge base entities in documents (entity linking ③) or creating new knowledge bases by annotating subjects, predicates and objects in text (fact linking ⑦⑧). Users can also explore and edit the knowledge base contents within INCEpTION.

To facilitate the entity linking process, INCEpTION can optionally take into account the context of the entity mention in order to provide the user with a ranked list of potential candidates. The same approach is used to drive an entity linking recommender which displays high-ranking candidates as annotation suggestions ⑨ in the annotation area where the user can accept them with a single click.

Customizability and extensibility There are two approaches to customize and extend INCEpTION:

Internal extensions. The dependency injection and event mechanisms of Spring Boot[3] are used to internally modularize INCEpTION. Extension points make it possible to register new types of annotation properties, new editors or new internal recommenders. Modules can coordinate their tasks with each other through events. As an example, the main annotation area issues an event when an annotation has been created or changed. The recommenders and the AL mode react to this event in order to update themselves. Functionality can thereby not only be added but also removed to create custom branded versions of INCEpTION. The event-driven modular approach also enables the system to comprehensively log user and system actions. This data can for instance be used by annotation project managers in order to evaluate the performance of their annotators.

External extensions. Currently supported are external recommenders and knowledge bases. Benefits of using external services include increased stability (failing services do not crash the entire platform), scalability (deploy resource-hungry services on different machines) and the free choice of programming language (e.g. most deep learning frameworks are not implemented in Java).

Additionally, INCEpTION uses (de-facto) standards such as UIMA for annotations and RDF, OWL and SPARQL for knowledge bases to achieve a high level of interoperability with existing tools and resources.

4 Use cases

To ensure that INCEpTION remains generic, we collaborate with multiple use cases:

FAMULUS. Schulz et al. (2017) use INCEpTION to annotate medical case study reports with argument components. These are used to train a machine learning model which evaluates the diagnostic competence of aspiring doctors. A pre-trained deep learning model is integrated as an external recommender and is used during annotation. The annotators that use INCEpTION in conjunction with the recommenders report the usefulness and improvement in annotation speed and quality.

EDoHa. Stahlhut et al. (2018) have created a hypothesis validation tool using INCEpTION. It features a hypothesis/evidence editor which allows users to create hypotheses and link evidence in the form of text paragraphs to it.

Knowledge-driven rntity ranking. In order to support users during entity linking, the re-ranking approach described in (Sorokin and Gurevych, 2018) was adapted and integrated into INCEpTION. It is used as a recommender and in the auto-suggestion box for the named entity layer.

As part of the collaborations with the above use-cases, INCEpTION logs the users' actions in order to investigate for instance which assistive features (i.e. recommenders) work best for the respective tasks, whether they introduce a bias in the annotator's results, and how to improve the user interface for an improved user experience.

[3] Spring Boot: `https://projects.spring.io/spring-boot/`

5 Conclusion and Future Work

In this paper, we have presented INCEpTION which –to the best of our knowledge– is the first modular annotation platform which seamlessly incorporates recommendations, active learning, entity linking and knowledge management. Our approach provides a number of advantages over current state-of-the-art annotation tools. Recommenders giving suggestions on-line allow users to annotate texts more quickly and accurately. External recommenders can be added to leverage already existing machine learning models and bootstrap the annotation for new domains. Knowledge management is directly integrated which allows entity- and fact linking together with building the knowledge base on the fly. The modular approach used by INCEpTION provides users with the possibility to tailor the platform according to their needs, for instance by adding new machine learning algorithms, annotation editors or knowledge bases.

INCEpTION is publicly available as open source-software. We welcome early adopters and encourage feedback for a continued alignment of the platform with the needs of the community. Several collaborations are on the way to develop and improve features that are useful to researchers and annotators, e.g. corpus search, recommenders that check the plausibility of annotations or fully custom user interfaces.

Acknowledgements

We thank Wei Ding, Peter Jiang and Marcel de Boer and Naveen Kumar for their valuable contributions and Teresa Botschen and Yevgeniy Puzikov for their helpful comments. This work was supported by the German Research Foundation under grant No. EC 503/1-1 and GU 798/21-1 (INCEpTION).

References

Kalina Bontcheva, Hamish Cunningham, Ian Roberts, Angus Roberts, Valentin Tablan, Niraj Aswani, and Genevieve Gorrell. 2013. GATE Teamware: A Web-based Collaborative Text Annotation Framework. *Language Resources and Evaluation*, 47(4):1007–1029.

Christian Girardi Emanuele Pianta and Roberto Zanoli. 2008. The TextPro Tool Suite. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 2603–2607.

David Ferrucci, Adam Lally, Karin Verspoor, and Eric Nyberg. 2009. Unstructured information management architecture (UIMA) version 1.0. OASIS Standard.

David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12.

Philip V. Ogren. 2006. Knowtator: A Protégé plug-in for annotated corpus construction. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Demonstrations*, pages 273–275.

Frédéric Papazian, Robert Bossy, and Claire Nédellec. 2012. AlvisAE: a collaborative Web text annotation editor for knowledge acquisition. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 149–152. Association for Computational Linguistics.

Claudia Schulz, Michael Sailer, Jan Kiesewetter, Christian M. Meyer, Iryna Gurevych, Frank Fischer, and Martin R. Fischer. 2017. Fallsimulationen und automatisches adaptives Feedback mittels Künstlicher Intelligenz in digitalen Lernumgebungen. *e-teaching.org Themenspecial "Was macht Lernen mit digitalen Medien erfolgreich?"*, pages 1–14.

Daniil Sorokin and Iryna Gurevych. 2018. Mixing context granularities for improved entity linking on question answering data across entity categories. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 65–75.

Chris Stahlhut, Christian Stab, and Iryna Gurevych. 2018. Pilot experiments of hypothesis validation through evidence detection for historians. In *Proceedings of Design of Experimental Search & Information REtrieval Systems (DESIRES)*. (in press).

Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96.

Chapter 7

Analyzing Dataset Annotation Quality Management in the Wild

Analyzing Dataset Annotation Quality Management in the Wild

Jan-Christoph Klie*
Ubiquitous Knowledge Processing Lab
Department of Computer Science and
Hessian Center for AI (hessian.AI)
www.ukp.tu-darmstadt.de

Richard Eckart de Castilho
UKP Lab

Iryna Gurevych
UKP Lab

Data quality is crucial for training accurate, unbiased, and trustworthy machine learning models as well as for their correct evaluation. Recent works, however, have shown that even popular datasets used to train and evaluate state-of-the-art models contain a non-negligible amount of erroneous annotations, biases, or artifacts. While practices and guidelines regarding dataset creation projects exist, to our knowledge, large-scale analysis has yet to be performed on how quality management is conducted when creating natural language datasets and whether these recommendations are followed. Therefore, we first survey and summarize recommended quality management practices for dataset creation as described in the literature and provide suggestions for applying them. Then, we compile a corpus of 591 *scientific publications introducing text datasets and annotate it for quality-related aspects, such as annotator management, agreement, adjudication, or data validation. Using these annotations, we then analyze how quality management is conducted in practice. A majority of the annotated publications apply good or excellent quality management. However, we deem the effort of 30% of the works as only subpar. Our analysis also shows common errors, especially when using inter-annotator agreement and computing annotation error rates.*

1. Introduction

Having large, high-quality annotated datasets available is essential for developing, training, evaluating, and deploying reliable machine learning models (Sun et al. 2017; Bender and Friedman 2018; Peters, Ruder, and Smith 2019; Gururangan et al. 2020; Sambasivan et al. 2021). Annotated datasets are also frequently used in linguistics (Haselbach et al. 2012), language acquisition research (Behrens 2008), bioinformatics (Zeng et al. 2015), healthcare (Suster, Tulkens, and Daelemans 2017), and the digital humanities (Schreibman, Siemens, and Unsworth 2004). Concerning machine learning, recent work has shown, however, that even datasets widely used to train and evaluate state-of-the-art models contain non-negligible proportions of questionable labels. For instance, the CoNLL-2003 (Tjong Kim Sang and De Meulder 2003, named entity recognition) test split has an estimated 6.1% wrongly labeled instances (Reiss et al. 2020; Wang

[*] Corresponding author

et al. 2019), ImageNet 5.8% (Vasudevan et al. 2022; Northcutt, Athalye, and Mueller 2021, image classification) and TACRED 23.9% incorrect instances (Stoica, Platanios, and Poczos 2021, relation extraction). GoEmotions (Demszky et al. 2020, sentiment classification) is estimated to contain even up to 30% wrong labels.[1] Using these datasets for machine learning can —among other issues— lead to inaccurate estimates of model performance (Reiss et al. 2020; Vasudevan et al. 2022), generalization failure due to data bias (McCoy, Pavlick, and Linzen 2019), or decreased task performance (Stoica, Platanios, and Poczos 2021; Vădineanu et al. 2022).

Recently, conversational agents and search engines based on large language models trained via instruction tuning have been widely adopted in science and society (Ouyang et al. 2022; Wei et al. 2022). Hence, datasets used for fine-tuning must be factually correct and contain as few biases as possible for the resulting models to be accurate and trustworthy and not to cause misinformation or harm. Benchmark datasets to evaluate their performance and rankings also need to be as accurate as possible to allow fair comparisons.

Proper quality management must be conducted throughout the dataset creation process (as depicted in § 3.1) to produce high-quality datasets. Dataset quality is not only limited to label accuracy but also encompasses aspects such as the quality of the underlying text, annotation scheme, adhering to established practices, or standards for a task, and social or data bias. Quality management encompasses, among others, proper data selection, choice of annotators and training, creating and improving annotation schemes and guidelines as well as annotator agreement, data validation, and error rate estimation (Hovy and Lavid 2010; Alex et al. 2010; Pustejovsky and Stubbs 2013; Monarch 2021). Even though there exists an extensive body of work that discusses quality management in theory (see § 2), we observe that this knowledge is difficult to find and to consult, as it is scattered across many different sources and usually treated as part of the general annotation process, hence often lacking depth. Also, to the best of our knowledge, no work as of yet has analyzed whether and how these recommendations are applied in practice. Disseminating and analyzing quality management is especially relevant in the context of a growing number of datasets being created and released, which can exhibit the aforementioned discussed dangers of low-quality data collection.

To better understand how quality management is actually performed in practice, we first survey the literature to summarize good practices regarding quality management for dataset creation. Based on *Papers With Code*[2], we then collect and annotate a large set of publications (591, of which 314 report human annotation or validation) that introduce new text datasets, and analyze how often and how well the different quality management methods are used. We also analyze the coverage of *Papers With Code* with regard to the ACL anthology, LDC corpora and shared tasks to validate the representativeness of our collected dataset. Finally, we summarize our findings and provide suggestions that dataset creators can use to consult and improve their annotation process. To the best of our knowledge, this newly annotated dataset and analysis of annotation good practices is the most extensive and detailed to date. We answer the following research questions:

RQ 1 What are good practices for data annotation quality management as described in the literature and derived from actual annotation projects?

RQ 2 Compared to the previously collected good practices, which methods are actually used in practice?

[1] https://archive.ph/jQbNM

[2] https://paperswithcode.com/datasets

RQ 3 Overall, how thorough is annotation quality management conducted in practice?

Our analysis shows that while many datasets are created according to good practices, several widespread issues exist. When using inter-annotator agreement, there is a frequent lack of actual interpretation of the agreement values. Also, sample sizes tend to be too low to make statistically sound conclusions when computing agreement and estimating the annotation error rate. Good practices suggested by the literature, like annotator training, pilot studies, or an iterative annotation process, are only mentioned rarely. Another interesting finding is that most of the time, adjudication is performed via majority voting; we found only three datasets that reported using probabilistic aggregation. Overall, we find a lack of proper reporting of how the annotation process was planned and executed, who annotated, as well as which quality management methods were used. These issues make it more difficult to gauge the quality of datasets and can hinder reproducibility. In summary, our contributions are:

- We survey the literature and compile an extensive summary of quality management methods.
- We analyze how quality management is done in **practice** compared to the good practices we found and recommend and point out common mistakes.
- Based on our findings, we provide a list of recommendations that can be used by future dataset creators to improve the quality of their datasets and to avoid common pitfalls.

In order to foster further investigation into quality management for data annotation, we will also release our code[3] to collect and analyze the dataset as well as our annotations [4]. Our dataset can also be used as a reference to find papers that use specific quality management methods and serve as an example of how to apply them.

2. Background

In the following section, we discuss the most relevant works dealing with the dataset creation process in general and its quality management in particular. By quality management, we understand the overall process and measures taken to reach and maintain a desirable level of quality. The quality management measures we found are described in detail in § 3.

Dataset Creation. Dataset creation subsumes several activities, which can be coarsely divided into three categories: *annotation*, *production* or *evaluation* (Shmueli et al. 2021). Different quality management methods are applicable or should be used depending on the task. Annotation or labeling means enriching data with additional information, e.g., tags for text classification. Production encompasses activities like writing the text for question answering, paraphrasing, or summarizing. Evaluation means using humans to compare or assess properties like quality of previously labeled or produced instances. These can be manually or automatically created. While also touching on text production, this article primarily discusses annotation quality management. We call participants in a dataset creation process still annotators, even if they only perform production.

[3] https://github.com/UKPLab/qanno ; GPL v3

[4] https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/3939 ; CC BY-NC 4.0

Dataset Creation Good Practices. Several books and articles have been written discussing dataset creation, especially concerning the annotation process itself. For instance, Ide and Pustejovsky (2017) collected descriptions for a wide range of different annotation projects. Pustejovsky and Stubbs (2013) describe the annotation process targeted towards training a machine learning model. However, both focus mainly on setting up the respective annotation projects, collecting data, as well as developing the annotation scheme and guidelines. Quality management is mentioned, but – except for inter-annotator agreement – not discussed in depth. Hovy and Lavid (2010) define good practices concerning conducting linguistic annotation projects. They emphasize the importance of proper annotator selection and training and how to evaluate the resulting dataset quality using agreement. Monarch (2021) discusses quality management for data annotation in the greatest detail. Their focus is predominantly on how to evaluate the quality of annotated data, from simple agreement measures over comparison with gold data to annotator-specific performance. Wynne (2005) describe good practices when creating linguistic corpora but only mention quality as important, not how to assure it. Similarly, Roh, Heo, and Whang (2021) survey the different ways to collect data, for instance, via annotation, distant or self-supervision, but only bring up quality management in a short paragraph.

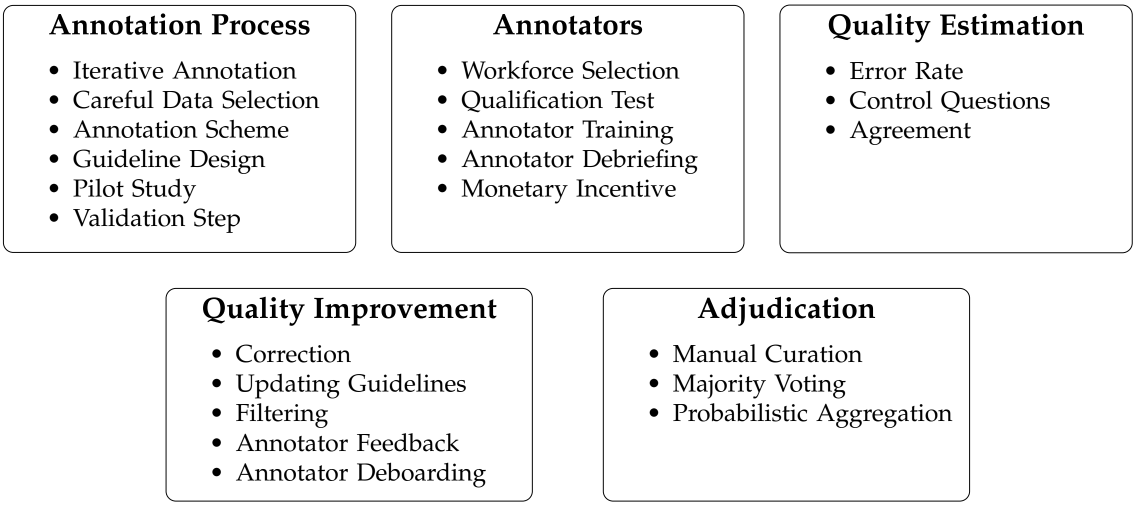
Several large-scale projects were conducted to develop standards and recommendations for creating language resources. These projects are, among others, the *Expert Advisory Group on Language Engineering Standards* (EAGLES) funded by the European Union (launched in 1993) or ISO/TC 37/SC 4, a technical subcommittee within the International Organization for Standardization. The resulting standards are either relatively challenging to find or require payment. While searching, we did not find explicit mentions of quality management or related recommendations.

Quality Management in Crowdsourcing. Many works have shown that crowdworkers can annotate or create datasets with similar quality compared to experts (Snow et al. 2008; Hovy, Plank, and Søgaard 2014). Proper quality management is especially important in crowdsourcing, where the risk of unreliable workers is usually higher (Hovy et al. 2013). An early work describing basic quality control measures to use with Amazon's Mechanical Turk is given by Callison-Burch and Dredze (2010). These include having multiple annotators for each instance or using control instances to estimate annotator quality. Daniel et al. (2019) define a taxonomy of quality for crowdsourcing and extensively describe related quality control measures. Their survey focuses on annotator management and how it is implemented in annotation tools. Unlike our study, they do not analyze if and how quality control measures are used in practice as reported by dataset-introducing scientific publications. Lease (2011) note that the annotation platform and tools can automate quality management in crowdsourcing to a certain degree, but manual inspection is still needed.

Annotation Process Analysis. Sabou et al. (2014) analyze 13 datasets created by crowdsourcing concerning how they were collected and derive good practices from this analysis. Amidei, Piwek, and Willis (2019) analyze inter-annotator agreement in the context of natural language generation evaluation and annotate 135 publications for this. Compared to these works, we go beyond analyzing only crowdsourced datasets, have a more detailed annotation scheme, annotate as well as analyze far more publications, and summarize quality management measures and good practices in greater detail.

Dataset documentation checklists. In the past, it has been found that datasets were often not adequately documented and were just published as-is. Therefore, several works proposed checklists and templates that should be published alongside the dataset to remedy this issue. These are, among others, *datasheets for datasets* (Gebru et al. 2021), *dataset nutrition labels* (Holland et al. 2018), *data statements for NLP* (Bender and Friedman 2018), *accountability frameworks* (Hutchinson et al. 2021), or *data cards* (Pushkarna, Zaldivar, and Kjartansson 2022). Similarly, more and more machine learning and natural language processing (NLP) conferences have adopted and are adopting reproducibility checklists for machine learning model training. The focus of these checklists is mostly on bias, annotator background, intended use, general data statistics, data description, data origin, or preprocessing. Kottner et al. (2011) propose a checklist that can be used when using agreement values, which is a good start but very specific to only a single aspect of quality management. It is designed for clinical trials and might require adaptation for use in NLP. We did not find any checklist explicitly targeted towards overall quality management.

To summarize, while a large body of work generally discusses the dataset creation process, the parts discussing quality management are relatively scarce, quite scattered in the literature, and not easy to find. Therefore, we summarize the literature and provide an easily referenceable set of good practices and recommendations for the dataset-creation practitioner. We additionally annotate a large set of dataset-introducing papers for their quality management and conduct an extensive empirical evaluation of how it is applied in practice. To the best of our knowledge, our analysis of quality management in textual dataset publications is currently the largest and the first, while not limited to a particular area like crowdsourcing.



Annotation Process

- Iterative Annotation
- Careful Data Selection
- Annotation Scheme
- Guideline Design
- Pilot Study
- Validation Step

Annotators

- Workforce Selection
- Qualification Test
- Annotator Training
- Annotator Debriefing
- Monetary Incentive

Quality Estimation

- Error Rate
- Control Questions
- Agreement

Quality Improvement

- Correction
- Updating Guidelines
- Filtering
- Annotator Feedback
- Annotator Deboarding

Adjudication

- Manual Curation
- Majority Voting
- Probabilistic Aggregation

Figure 1: Quality Management methods discussed in this work. We categorize methods into annotation process, annotator management, quality estimation, quality improvement, and adjudication.

3. Dataset Creation Quality Management

To answer our first research question, in the following, we present the most relevant and frequently used quality management methods for dataset creation. This list is derived from good practices stated in previous works (§ 2) and the methods we found while surveying the dataset papers (§ 4) themselves. We consider the following methods *good* practices for two reasons. They are disseminated in well-acclaimed books or have been adopted by the community and are thus commonly used and tested in practice. We thus believe that the methods discussed in the following are well-suited for managing quality. It has to be mentioned, however, that only a few works have thoroughly investigated the exact impact of these methods on aspects like quality, time savings, or agreement (see also § 5 and § 8).

Another important point to consider is to see quality management as a means towards a goal and not as a goal in itself. Depending on the goal, for instance creating datasets with low bias, high quality, or diversity, some methods might be preferred over others. The choice of methods should thus be based on the purpose and usage goals.

Also, when applying the ensuing methods in practice, their use can be expensive. Therefore, extensive quality management needs to be balanced against the annotation costs itself when working on a limited budget; a healthy compromise between the two needs to be found.

We propose a taxonomy that puts the methods into five groups related to the *annotation process*, *annotator management*, *quality estimation*, *quality improvement*, and *adjudication*. While only briefly outlining the techniques here, we refer the interested reader to each method for a more in-depth description. An overview of the discussed methods is given in Fig. 1.

We differentiate between two types of tasks for dataset creation (see § 2), namely *annotation* (e.g., named entities or text classification) and *text production* (e.g., writing questions and answers for question answering, paraphrasing, sentence simplification). This distinction is important because specific quality management methods may work for one but not the other. For example, inter-annotator agreement and adjudication are usually not applicable to text production tasks. Both expert annotation and crowdsourcing are considered.

Our survey primarily focuses on annotation, especially label errors, but we also discuss annotation consistency, biases, and how to mitigate them. Regarding label errors, while it is sometimes impossible to assign a single, true label due to inherent ambiguity, especially in natural language processing, deciding whether a label is incorrect is often much more straightforward.

Before describing quality management methods, we first define what dataset quality subsumes. Following Krippendorff (1980); Neuendorf (2016), we suggest targeting at least the following quality aspects[5]:

Stability A dataset creation process is stable if its output does not drift over time. Drift here means that similar phenomena are annotated similarly independent of whether they are annotated earlier or later throughout the process. Instability can, for instance, occur due to carelessness, distractions or tiredness, change in annotation guidelines, or even learning through practice.

[5] Note that dataset creation projects that run over a very long time and which might be subject to external effects, such as general advances in the field or societal changes, may need other definitions for these categories or incorporate specific approaches to deal with such external effects.

Reproducibility A dataset creation process is reproducible if different annotators can still deliver the same results given the same project documentation regarding process, guidelines and scheme.

Accuracy Annotations and texts created during the process are accurate if they adhere to the guidelines and the desired outcome.

Unbiasedness This describes the extent to which the created artifacts are free of systematic, nonrandom errors (bias).

Stability, *reproducibility*, and *accuracy* are also subsumed under the term *reliability* in content analysis (Krippendorff 1980). *Consistency* is related to *stability* and *reproducibility*. *Reliability* thus measures the differences that occur when repeatedly annotating the same instances; it is empirical (Hardt and Recht 2022). It is required to infer *validity*, that is, to show that the annotations capture the underlying phenomenon targeted (Artstein and Poesio 2008), but not sufficient. *Validity* is latent and cannot be directly measured. Therefore, proxy metrics targeting reliability, for example agreement, need to be used instead.

3.1 Annotation Process

The following section describes the recommended annotation process. It is written concerning annotation but can easily be adapted to text production as well.

We suggest that an annotation project should start with a *planning phase*. It can encompass important preliminaries as setting the goal of data collection, making initial choices for data and annotators, setting a budget, desired quality level or reviewing the literature for similar datasets or relevant annotation practices. Ideally, these choices are documented and become part of the dataset documentation once the dataset gets released.

The annotation scheme is often developed during an annotation project and is a living document. Also, as annotators only get familiar with the task during the annotation process, issues are found just then, and the data or task needs to be adapted accordingly. Therefore, it is recommended to structure an annotation project as a sequence of cycles with iterative quality improvement actions (Hovy and Lavid 2010; Pustejovsky and Stubbs 2013; Monarch 2021). This approach is also called *agile corpus creation* (Alex et al. 2010). In each cycle, only a slice of the data is annotated: a batch. After the batch is annotated, it is *evaluated*, and *quality-improving/rectifying measures* are taken if needed. These cycles repeat until an acceptable quality level for a sufficient number of batches has been reached. Evaluation can be performed by inter-annotator agreement (§ 3.3.3), comparing annotations to a known gold standard to estimate annotator proficiency, or having experts or a different set of annotators inspect a subset or all instances and marking errors (§ 3.3.1 and § 3.3.2).

The advantage of this iterative approach is that changes are introduced at defined points during the process. Iterating, for example, mitigates the annotation scheme and annotations running out-of-sync and improves the chance of producing high-quality datasets. Our take on this annotation loop is depicted in Fig. 2. *Pilot studies* are the initial iterations used to create and improve the annotation process until it is good enough for the annotation of the dataset itself. Quality improvement measures can be, among others, retraining annotators, adjusting/clarifying the annotation guidelines or annotation scheme, onboarding or deboarding annotators, or giving back batches to annotators for correction (cf. § 3.4). In later iterations of an annotation project, when the setup has stabilized, the batch sizes can be increased, and quality control can be

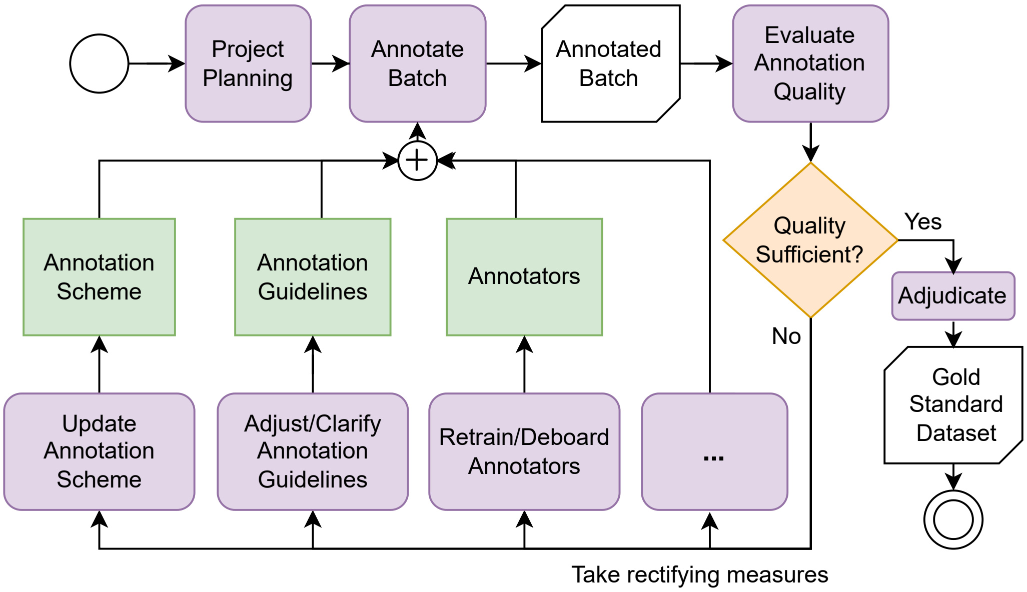


Figure 2: The recommended annotation process: After a batch of data is annotated, it is evaluated. If the quality is sufficient, it can be adjudicated. If not, several corrective measures can be taken, e.g., correcting the annotations in an additional step, annotator training, or adjusting the annotation scheme or guidelines. This is similarly applicable for text production workflows where usually no adjudication takes place.

performed less rigorously, e.g., reducing the fraction of samples inspected for quality checking or the annotations collected per instance. When using an iterative approach, stability of the annotation process needs to be taken into account, as changes to the process can cause differences in subsequently annotated batches. Also, if the annotation scheme or guidelines evolve too much, then re-annotation of previously annotated material might be necessary.

Careful Corpus Building. Not only are the labels assigned by the annotators important, but also the choice of texts that are annotated itself (Wynne 2005). Choosing texts that only rarely or even never contain the phenomena to annotate can be ineffective. Similarly, selecting texts that are of poor quality can be detrimental and cause issues in later stages of the machine learning pipeline. In order to achieve the best downstream task performance for trained machine learning models, texts should be representative of the data encountered in the target domain. Hence, it is vital to check the data for errors and unwanted aspects like non-representative content or biases, ideally before it reaches the annotators. This can be achieved by, e.g., manual inspection (Bastan et al. 2020; Govindarajan et al. 2020) (e.g., by the project manager or even as a separate preparatory annotation project) and filtering via rules (Reddy, Chen, and Manning 2019; Ghosal et al. 2022) or using spell-checking and text cleaning tools (Horbach, Ding, and Zesch 2017; Kim, Weiss, and Ravikumar 2022).

Annotation Scheme and Guideline Design. The annotation scheme defines the structure, features, and tagsets of the task to annotate. Its form and granularity can significantly impact the annotation process and the downstream machine-learning modeling. Therefore, it must capture the information of interest. The annotation scheme defines the annotation labels; the guidelines describe how to decide when to apply which label (e.g., disambiguating between different labels). Properly written guidelines are essential for annotator training to achieve consistency and reproducibility, e.g., when re-annotating, extending, or creating a similar dataset on different text. The way that the guidelines are written can by itself already introduce bias (Geva, Goldberg, and Berant 2019; Parmar et al. 2023) and therefore, great care needs to be taken when creating them. Instead of creating guidelines from scratch for every annotation project, existing guidelines can be reused and adapted for similar settings. In many annotation projects, the guidelines are revised several times as part of a pilot study before the actual annotation process starts (Hovy and Lavid 2010).

Guidelines for more complex annotation projects are often quite detailed and span many pages. They are usually very short in crowdsourcing and often fit into the annotation screen. Examples of excellent, extensive annotation guidelines can be found in Prasad et al. (2008) or Da San Martino et al. (2022). For crowdsourcing, good examples are given by Singh et al. (2021) or Mostafazadeh et al. (2020).

Pilot Study. When entering into an (iterative) annotation project, it is crucial to validate the annotation process on a smaller scale, i.e., by conducting one or more pilot studies with only a small annotator team (Pustejovsky and Stubbs 2013). Annotators in pilot studies are often the project managers themselves or a selected group of experts. We recommend that experts or project managers conduct the initial pilot study iterations; the annotation process should then be subsequently tested with the target annotators until all questions and issues are solved. This study should include developing the initial version of the annotation scheme and guidelines, configuring the respective annotation tooling, and developing the data pre-processing and post-processing steps (Kummerfeld et al. 2019). This way, issues can be spotted before investing too much effort into a flawed setup. Ideally, the data used for pilot studies should be selected to contain as many corner cases and difficult instances as possible. This reduces the chance that later, during the main part of the annotation project, significant adjustments need to be made that could cause costly re-annotation in case changes are not backward compatible. The overall difficulty of the task can be gauged, and it can be tested whether experts are needed or whether well-trained contractors or crowdworkers can achieve a desirable quality level. The expected cost can also be estimated by measuring annotation time per instance. The feedback annotators give during this phase is essential for a well-working annotation project (Monarch 2021). It has to be noted, however, that if experts or project managers conduct the initial pilot study, then they may use implicit knowledge that will not transfer to more general annotators (Krippendorff 1980).

Validation. After an annotation step has been completed, a validation step can (and should!) be added to check whether annotations are correct and of sufficient quality. Validation steps can take different forms based on the task and setup, e.g., experts can inspect a subset of annotations, or there can be a separate annotation phase asking for binary correctness labels. While validation is important, it needs to be weighed against spending on annotating more instances instead if the budget is limited.

It is also possible to design a more task-dependent validation step, for which we give examples in the following. We call this flavor of validation *indirect validation*. It is often applicable if the annotation task consists of different subtasks that depend on each other and are hence annotated sequentially. For question answering, a first step might be to write questions and answers. The validation step could then annotate which answer best fits a given question (Mihaylov et al. 2018). For relation extraction, the first step can be marking spans and labeling their relation (Yao et al. 2019). The validation step could be that annotators are given only the marked spans and are asked to label the relation. Alternatively, the relation label could be given, and annotators are asked to mark the spans with this relation. If annotations differ between subsequent steps, then they are potentially incorrect. For natural language inference, the first task can be defined as writing a premise and hypothesis, given a relation (entailment, neutral, contradiction). In the second step, the task can be to label the relation between the two given the premise and hypothesis. If the results in the first and second steps differ, these instances require further treatment (Bowman et al. 2015).

Validation is also relevant for automatically created datasets, e.g., by crawling external resources, distant or self-supervision. It should be performed after a batch of annotations have been made and before they are adjudicated. Validation can be part of quality estimation, which we discuss in more detail in § 3.3.1.

3.2 Annotator Management

Dataset creation projects stand or fall by the quality of the annotators; such a project often is an exercise in people management (Monarch 2021). At every step, it is vital to treat annotators fairly and respectfully. Here, we give a high-level overview of the different aspects of annotator management. An in-depth survey of annotator management focusing on crowdsourcing is also given in (Daniel et al. 2019; Monarch 2021). We consider both "classic" expert annotation and crowdsourcing in this work and point out when methods are more applicable for one or the other.

Workforce Selection. The type of workforce employed considerably impacts annotation time, cost, and quality (Hovy, Plank, and Søgaard 2014). Which kind of annotators to employ depends, among others, on the task difficulty, availability, target language, and whether particular expertise is needed. If the annotation task is solvable by crowdworkers, it is often an efficient way to annotate (Snow et al. 2008). For more involved tasks, trained contractors can be an alternative to hiring domain experts (Chen et al. 2021). Contractors are a middle ground between crowdworkers and experts; they are experienced in conducting annotation tasks but are not necessarily domain experts. It is recommended to validate the workforce choice in one or more pilot studies.

Qualification Filter. As a common way to filter out crowdworkers that might produce low-quality work, many crowdsourcing tools offer setting requirements for the worker. These, for instance, can be requiring a certain percentage of accepted tasks or a certain number of already completed tasks. Kummerfeld (2021) analyzes the impact of these measures on quality and discusses the ethical aspects of requiring a minimum number of tasks. They argue that it forces workers to accept a substantial amount of low-paying tasks to overcome this hurdle. The conclusion is that there is no clear relation between quality and filtering based on the percentage of accepted, previous tasks, and number of completed tasks. They also note that in practice, limits are often set too high. Thus, the paper recommends either running a pilot study to get estimates for the actual requirement values or prefer qualification tests (see below) over simple filters.

Qualification Test. A more elaborate way to identify good annotators is to use (paid) qualification tests (Kummerfeld 2021). Before an interested annotator can participate in the primary annotation process, they must work on a small set of qualification tasks. The answers are either compared against known answers or judged by experts.

If the performance is acceptable, then the annotator is allowed to work on the actual annotations themselves. The difficulty of the test can be varied based on how strictly the test should filter. For instance, task examples from the guidelines can be handed out to annotators to check whether they have been read and understood. A more challenging test would be to use new, previously unseen tasks. Qualification tests can and should not only be used for crowdsourcing but also when hiring contract annotators.

Annotator Training. Before involving new annotators in a project, it is often helpful to train them in the annotation task at hand, to go through the guidelines with them, and make sure that everything is clear (Neuendorf (2016, p. 133); Sabou et al. (2014)). Project managers and annotators can give each other feedback that can then be worked into the annotation scheme and guidelines. Feedback is especially important if annotators find the guidelines difficult to understand or if they contain errors. Bayerl and Paul (2011) conduct a meta-study and analyze, among other aspects, the effect of training on agreement. They show that the better and more intensely annotators are trained, the higher the agreement becomes. Also, they point out that training is beneficial not only to crowdworkers but also to experts, as the latter might be familiar with the domain but not with the project setup at hand. Training is also essential for annotation stability, as early in the process, annotators are often unsure and unfamiliar with the annotation process. This changes with more time spent annotating, rendering earlier annotations potentially inconsistent with later ones.

Annotator Debriefing. During and after the run of an annotation project, it is often helpful to ask one's annotators for feedback about the annotation project (Neuendorf 2016, p. 134). This feedback can then be used to improve the guidelines, update the annotation scheme, or alleviate issues that only became apparent while annotating. For instance, usability issues of the annotation editor, ways to make annotation faster, or data quality issues can be spotted and fixed before it is too late.

Monetary Incentive. Giving annotators additional monetary compensation in addition to their base pay might be an option (Harris 2011; Ho et al. 2015). The amount, for instance, can be based on their performance on control questions or after feedback rounds have shown that they reach the target for a bonus. Another way is to pay annotators more for sticking to a task (Parrish et al. 2021). If monetary incentives are used, it is essential to be transparent about it, communicate the requirements beforehand, be fair, and not change the rules post-hoc. Also, one needs to be careful that the targets for which monetary incentives are promised are not gamed with detrimental effect towards annotation quality. [6]

3.3 Quality Estimation

After annotations have been made, their quality should be estimated and compared to the desired quality level. In case it is insufficient, counter-measures should be taken to improve it.

[6] This is also known as Goodhart's law: *"When a measure becomes a target, it ceases to be a good measure"* (Goodhart 1984)

3.3.1 Manual Inspection. In order to judge the quality of an instance dichotomously as correct or incorrect, annotators (usually, they are different from the initial annotators) or project managers can manually inspect and grade them (Pustejovsky and Stubbs 2013). Validation can either be done on a subset of instances or as a complete validation step. In addition, after the dataset has been completely annotated, its error rate can be estimated and reported because even datasets considered gold often still contain errors (Northcutt, Athalye, and Mueller 2021). The error rate is computed by dividing the number of errors found by the number of instances inspected. Therefore, we strongly recommend inspecting a subset of instances of the final dataset, labeling their correctness, and thereby estimating the error rate. The notion of what is correct/of sufficient quality or incorrect/insufficient depends on the task at hand. Hence, manual inspection is not only applicable to annotation tasks but also to text production. There, it can be determined whether the produced instance is of sufficient quality. For ambiguous instances in annotation tasks, one would judge whether the label makes sense at all in this context.

3.3.2 Control Instances. In order to gauge the performance of annotators, instances can be injected into the annotation process for which the answer is known (Callison-Burch and Dredze 2010). These gold instances are often obtained by having experts annotate a subset beforehand. Another way is to compare a single annotator's submissions to the others'; the performance estimate is then the deviation from the majority vote (Hsueh, Melville, and Sindhwani 2009) or the agreement (Monarch 2021). For example, the resulting estimates can be used to retrain annotators if they annotated too many instances incorrectly, send batches created by underperforming annotators back for re-annotation, or remove annotators from the workforce. Well-performing annotators can also be monetarily rewarded or given tasks requiring more expertise, such as task validation or manual adjudication.

3.3.3 Agreement. A common way to quantify the reliability of annotations and annotators is to compute their inter-annotator agreement (IAA) (Ebel 1951; Krippendorff 1980, 2004). For NLP, it has been increasingly adopted after Carletta (1996) introduced agreement, coming from the field of content analysis, as an alternative to previously used ad-hoc measures. Here, we briefly present the most popular and recommended agreement measures. For a more in-depth treatment of agreement and how to apply it, we refer the interested reader to the excellent works of Krippendorff (1980); Lombard, Snyder-Duch, and Bracken (2002); Neuendorf (2016); Artstein and Poesio (2008); Monarch (2021).

Percent agreement. This is the most straightforward agreement measure. It considers the percentage of coded units on which two annotators have agreed. This measure, however, suffers from several issues (Krippendorff 1980, 2004; Artstein and Poesio 2008). First, it yields skewed results for imbalanced datasets, similar to accuracy when evaluating classification. Second, it does not consider when annotators assign the same label by chance, for instance, in case they randomly guess or spam. Third, percent agreement is influenced by the size of the tagset. Therefore, it is difficult to compare across annotation schemes. Finally, there are only two values of percent agreement that are meaningful and intuitive, which are 0% and 100%. These issues together cause percent agreement to be uninformative and difficult to interpret and compare when estimating reliability. Therefore, the usage of percent agreement is discouraged and should especially not be the only agreement measure reported.

Cohen's κ. In order to remedy the issues of percent agreement, Cohen (1960) proposes a chance-corrected coefficient, normalized to $[-1, 1]$, to measure the agreement between two annotators. Negative values indicate disagreement, 0 the expected chance agreement, and values greater than 0 indicate agreement. κ requires that the same number of annotators annotate all instances; no entries may be missing. Also, annotations need to be categorical. It is defined as

$$\kappa_C = \frac{p_o - p_e}{1 - p_e}$$

where p_o is the observed proportionate agreement and p_e the chance agreement.

Fleiss's κ. Fleiss (1971) extend Scott's π (Scott 1955) to multiple annotators.[7] Similarly to Cohen's κ, each instance needs to be labeled by the same number of annotators. In addition, Fleiss' κ assumes that annotators for each instance are sampled randomly, it is not suitable for settings where all annotators annotate all instances (Fleiss, Levin, and Paik 2003). It is defined as

$$\kappa_F = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}$$

where $\bar{P}$ measures observed agreement as the average agreement over annotator pairs and P_e is the expected agreement by chance.

Krippendorff's α. A different way to estimate agreement has been proposed by Krippendorff (1980). It is based on the quotient of observed disagreement D_o and chance disagreement D_e:

$$\alpha = 1 - \frac{D_o}{D_e}.$$

Compared to Fleiss's κ, Krippendorff's α is more powerful and versatile: it can deal with missing annotations, supports more than two annotations per instance, and can be generalized to handle even categorical, ordinal, hierarchical, or continuous data (Hayes and Krippendorff 2007). For instance, span labeling tasks like named entity recognition or relation extraction can be evaluated using a coefficient of the Krippendorff's unitized α (α_u) family (Krippendorff et al. 2016).[8] Unitizing means that annotators first divide the instances into smaller units and only then assign labels (Lombard, Snyder-Duch, and Bracken 2002, Chapter 4). In the context of named entity annotation, unitizing, for instance, can be marking spans that contain entities or, for object detection, drawing bounding boxes around objects of interest. Hence, Krippendorff's α can also be applied

[7] Fleiss' κ is not an extension of Cohen's κ, as it assumes similarly to Scott's π that the labeling distributions are the same for each annotator, which Cohen's κ does not (Artstein and Poesio 2008).

[8] The α_u family currently consists of four different coefficients (Krippendorff et al. 2016). They differ in how and whether 'gaps' (unannotated units) are take into consideration, whether labels or only units are used, or whether only a subset of labels are used when computing agreement. α_{cu} is the most applicable choice of the four that ignores gaps and takes label values into account.

to any task with a one-to-many relation between instances and annotations of different sizes. The amount of overlap between annotations made by different annotators is also considered by α_u when computing agreement. While being flexible, α is also more complicated to implement (especially in its unitizing form), has a higher runtime, and is more challenging to interpret and to compute confidence intervals for (Artstein and Poesio 2008).

Correlation. For specific tasks, annotation consists of assigning scores to instances on a numerical, continuous, or discrete rating scale or a Likert scale. These tasks are, among others, annotating sentiment (Socher et al. 2013), emotions (Demszky et al. 2020), or semantic textual similarity (Cer et al. 2017). Correlation measures like Pearson's r (linear correlation), Spearman's ϱ (linear correlation of ranks), or Kendall's τ (correlation of concordant/discordant ranks) are often used to compute agreement. However, using correlation coefficients as an agreement measure is controversial, as they measure covariation, not agreement, i.e., they measure whether variables move together, but not whether they really are similar (van Stralen et al. 2012; Ranganathan, Pramesh, and Aggarwal 2017; Edwards, Allen, and Chamunyonga 2021). This means that two annotators with different biases when assigning scores, e.g., one annotator systematically gives overly large scores while the other systematically underscores, would still have a high correlation but low agreement. A better alternative to the aforementioned correlation coefficients is using Intraclass Correlation (ICC) (Fisher 1925), which is explicitly designed to measure agreement. Note that there are several different formulations of ICC depending on the number of judgments per instance, whether judgments are averaged before comparison, and whether there are missing observations (Shrout and Fleiss 1979). A visual method to assess agreement between continuous variables is the Bland–Altman plot (Bland and Altman 1986). A worked example can be found in Appendix 3.

Classification Metrics. Especially for sequence labeling tasks like named entity recognition, classification metrics like *accuracy*, *precision*, *recall*, and F_1 are often used between two annotators to compute agreement (Brandsen et al. 2020). We could not find any work formally analyzing the theoretical background and implications of using these metrics as an agreement measure. However, they seem to suffer from several issues. First, they are only applicable as pairwise agreement; having more annotators would require averaging, which might cause information loss. Second, they are not chance-corrected (Powers 2011). Third, using precision and recall for computing agreement also has the downside of not being symmetric. Given two lists of labels a and b, the precision value of a and b turns into the recall when swapping its arguments: precision(a, b) = recall(b, a). Being symmetrical is essential for agreement metrics, as one annotator should not be preferred over another. This differs from classification metrics, where one input is from the gold data, and the other is usually from model predictions.

Although it is often treated as such, agreement is no panacea; high agreement does not automatically guarantee high-quality labels. Krippendorff (2004); Artstein and Poesio (2008) emphasize that agreement only demonstrates a reliable annotation process, which is necessary for high-quality labels but is by itself not sufficient. Further quality management, especially manual inspection, should be applied. Agreement also does not cover whether the annotation scheme and guidelines capture the desired phenomena. Low agreement also does not automatically mean low-quality labels, as tasks can inherently be subjective (Aroyo and Welty 2015; Uma et al. 2021), i.e., there are cases where no distinct gold label exists for an instance.

Using only a single agreement coefficient value to gauge quality is often insufficient for a reliable estimate. Therefore, more in-depth analysis is recommended (Artstein and Poesio 2008). This can be done by manually validating the annotations (cf. 3.3.1) to get an intuition for the resulting labels and why annotators disagree. Disagreements can be caused by differences in annotator skill, differences in the data or its difficulty (Jamison and Gurevych 2015), or due to ambiguity. Other insights can be gained by computing pairwise agreement between individual annotators or by computing agreement per label (Monarch 2021). These statistics may identify poorly performing annotators or particularly difficult-to-decide labels.

If the sample size is chosen too small, the resulting agreement value might have only limited explanatory power (Allan 1999; Shoukri, Asyali, and Donner 2004; Sim and Wright 2005). It is therefore recommended to have large parts of the dataset annotated by multiple annotators for a representative agreement value (Passonneau and Carpenter 2014). Ideally, every instance should be annotated by at least two annotators to draw reliable conclusions from agreement.

Several works propose value ranges for agreement coefficients and attach a semantic meaning to them. For instance, Landis and Koch (1977) give labels for certain value ranges of Cohen's κ (κ_c), e.g., $0.01 - 0.20$ *slight agreement*, $0.21 - 0.40$ *fair agreement*, $0.41 - 0.60$ *moderate agreement*, $0.61 - 0.80$ *substantial agreement*, $0.81 - 1.00$ *almost perfect agreement*. Similarly, Banerjee et al. (1999) say $\kappa_c > 0.75$ indicates excellent agreement, between 0.40 and 0.75 as fair to good agreement, and lower indicates poor agreement. Popping (1988) considers κ_c above 0.8 as reliable. Krippendorff (2004) considers their $\alpha \geq 0.8$ as reliable (later, they stated that it is the absolute lower limit and should better be 0.9) and $0.667 < \alpha < 0.8$ should only be used to draw tentative conclusions. An α value below 0.667 is said to indicate that the underlying labels are unreliable.

However, it must be noted that those boundaries are arbitrary, have certain assumptions (for instance, Landis and Koch (1977) consider only binary classification) to the task setup, and have no theoretical foundation. In general, choosing a target agreement level that is considered good enough is very difficult; there is no universally acceptable agreement level that is correct for every setting (Bakeman et al. 1997; Neuendorf 2016). Lombard, Snyder-Duch, and Bracken (2002) find that values above 0.9 are nearly always acceptable, greater than 0.8 acceptable in most situations and greater than 0.7 acceptable for exploratory studies for some indices. Artstein and Poesio (2008) state that these limits work well in their experience, and datasets reported with lower agreement values tend to be unreliable. The threshold may also depend on the difficulty and subjectivity of the annotation task. When stating agreement values, it is therefore essential to report boundaries and justify their value. It is also recommended to compare agreement value to other works that annotate similar phenomena and tasks if possible.

Finally, the different agreement methods have several idiosyncrasies related to how they are computed and how they behave (Zhao, Liu, and Deng 2013; Checco et al. 2017). For instance, annotations with near-perfect percent agreement can have low Cohen's κ. When Krippendorf's α is applied to a large number of instances, then its computed chance agreement term increases while α reduces, thereby favoring smaller samples. Agreement also decreases when having more annotators per instance, but this does not indicate worse quality; fewer annotators often just do not annotate the whole possible range (Bayerl and Paul 2011), and therefore, the agreement is an overestimate. These characteristics can lead to non-intuitive behavior and render interpretation more difficult.

3.4 Quality Improvement

If the quality estimation shows that the annotation quality is insufficient, rectifying measures must be taken to improve it.

Manual Correction. If the quality in a batch of annotations is too low, it can be returned to the annotators for further improvement. Also, it can be routed to different, more experienced annotators to resolve issues in case instances are too difficult for the original annotators.

Updating Guidelines. It can happen that the annotation guidelines do not cover certain phenomena in the underlying text, are ambiguous, or are difficult to understand. Then, it might be appropriate to go back to the annotation scheme or guidelines and improve them (Bareket and Tsarfaty 2021). Updating the guidelines may require discarding previously created annotations or at least reviewing and updating them. If quality estimation shows that similar categories have low agreement, then this can hint at that annotators have issues discerning between them. One possible solution could be updating the annotation schema so that these categories are collapsed to a single label (Lindahl, Borin, and Rouces 2019).

Data Filtering. There are several scenarios in which already annotated instances should be prevented from making it into the final dataset. Sometimes, certain instances are too ambiguous for which annotators then strongly disagree on a single, correct label (Uma et al. 2021). Occasionally, annotations can be of low quality and should be removed. A simple solution is to filter out these instances and not process them further. The filtering can, for instance, be based on expert judgment or if there is no majority agreement (Bastan et al. 2020). Sometimes, measuring the time it takes for annotators to process instances and filter out annotations with improbably high annotation times might also be helpful (Ferracane et al. 2021).

Before filtering based on agreement, the source of disagreement should be understood, and ideally, manual inspection of flagged instances should be performed. Disagreements can for instance be visualized using confusion matrices. Filtering instances has the potential disadvantage of reducing diversity, which should be considered. Recent work also emphasizes that disagreement is inherent to natural language (Aroyo and Welty 2015) and can, for instance, be used to create a hard dataset split or even directly learn from them (Checco et al. 2017; Uma et al. 2021). Improving the annotation guidelines to incorporate edge cases should therefore be preferred over filtering.

Annotator Training through Feedback. After annotators complete a batch, experts can manually inspect the data and give annotators feedback. Thereby, common errors can be pointed out, and aspects to improve can be discussed (Ghosal et al. 2022; Kirk et al. 2022). More detailed and extensive feedback might be more feasible for smaller annotator pools, e.g., contractors or expert annotators.

Annotator Deboarding. If certain annotators repeatedly deliver low-quality work, removing them from the annotator team might be desirable. One way to find these annotators is via annotation noise (Hsueh, Melville, and Sindhwani 2009), which describes the deviation of each annotator from the majority. Another is a manual inspection by the dataset creators or more seasoned annotators. Spammers can also be detected during adjudication (§ 3.5), for instance, by using MACE (Hovy et al. 2013, multiannotator competence estimation). After deboarding annotators, it is recommended that their annotations are marked to be redone. Even though some platforms like Amazon Mechanical Turk make it possible to withhold payment, they should still be paid for the work already done unless there is compelling evidence for excessive fraudulent behavior.

Automatic Annotation Error Detection (and Correction). Instead of having human annotators manually inspect instances and search for errors, automatic approaches can be used. For some error types, it is possible to write checks that automatically find issues and sometimes even correct them (Květoň and Oliva 2002; Qian et al. 2021). These checks can be simple rules that define wrong surface form and label combinations and are derived from the data. For noisy text like Twitter data or crawled forum texts, spell-checking might improve the underlying text before it is given to annotators. A more involved approach is annotation error detection, which leverages machine learning models to automatically find error candidates, which can then be given to annotators for manual inspection and an eventual correction (e.g., Dickinson and Meurers 2003; Northcutt, Jiang, and Chuang 2021; Klie, Webber, and Gurevych 2023). Automatic checks should always be validated by human annotators to not accidentally introduce new errors.

3.5 Adjudication

In order to increase overall annotation reliability, oftentimes, more than one label per instance is collected. These usually need to be *adjudicated*, that is, finding a consensus to create the final dataset with one single label per instance (Hovy and Lavid 2010). For reproducibility, it is suggested to not only publish the adjudicated corpus but also raw annotations by the respective annotators. Learning from individual labels is also an option, especially in tasks with considerable ambiguity and disagreement (Uma et al. 2021); then, no adjudication is used. While being an effective way to improve reliability, collecting more than one label per instance needs to be weighed against annotating more instances when working on a limited budget. The most common adjudication methods are described in the following.

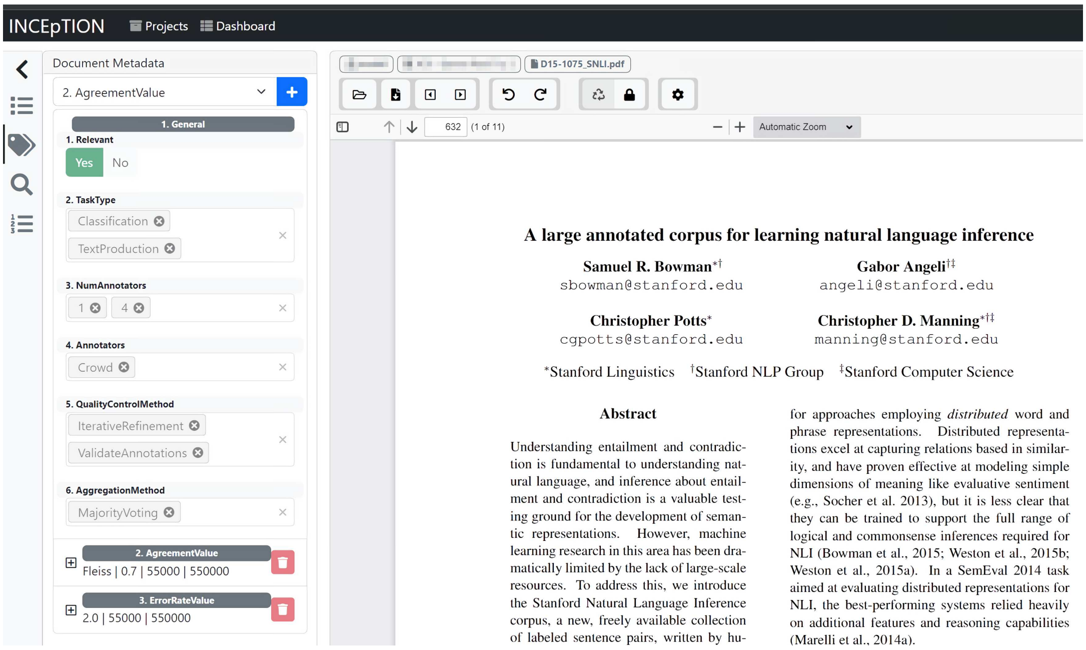
Manual Adjudication. To create a gold corpus, skilled annotators, often domain experts, manually inspect and curate each instance to a single label (Bareket and Tsarfaty 2021). While slow and expensive, this approach can yield high-quality data because ties can be broken and errors corrected during this inspection procedure. Curation can be sped up with automatic tooling, for instance, by automatically merging instances for which there is no disagreement or where the disagreement is below a certain threshold.

Majority Voting. When using majority voting, given an instance rated by multiple annotators, its resulting label is the one that has been chosen most often. Instances without majority label can be discarded or given to an additional annotator to break the tie. These are often experts but can also be (experienced) crowdworkers or contractors. In some works, supermajority voting is used. It means that more than 50% of annotators must agree, e.g., at most one differing label is allowed, or even a unanimous vote is required. Majority voting is easy to implement and a strong baseline compared to the more complex methods described in the following (Paun et al. 2018). But Lease (2011) notes that using majority voting might drown out valid minority voices and can reduce diversity, which should be taken into account.

Probabilistic Aggregation. In majority voting, it is assumed that all annotators are equally reliable as well as skilled and that errors are made uniformly at random. This assumption does not always apply in real annotation settings, especially for crowdsourcing. Annotators can be better or worse in certain aspects, might be biased, spamming, or even adversarial (Passonneau and Carpenter 2014). To alleviate these issues, Dawid and Skene (1979) propose a probabilistic graphical model (that is referred to as Dawid-Skene, named after its inventors) that associates a confusion matrix over label classes for each annotator, thereby modeling their proficiency and bias. The resulting aggregation is then based on weighing labels with the respective annotator's expertise for this label. An alternative formulation called *MACE* that also models spammers is given by Hovy et al. (2013).

It has been shown that using more sophisticated aggregation techniques can yield higher-quality gold standards (Passonneau and Carpenter 2014; Paun et al. 2018; Simpson and Gurevych 2019), but majority voting is often a strong baseline. The works mentioned above also discuss probabilistic aggregation in more detail.

4. Data Collection and Annotation



2. AgreementValue

1. General

1. Relevant: Yes No

2. TaskType: Classification, TextProduction

3. NumAnnotators: 1, 4

4. Annotators: Crowd

5. QualityControlMethod: IterativeRefinement, ValidateAnnotations

6. AggregationMethod: MajorityVoting

2. AgreementValue: Fleiss | 0.7 | 55000 | 550000

3. ErrorRateValue: 2.0 | 55000 | 550000

A large annotated corpus for learning natural language inference

Samuel R. Bowman[*†] sbowman@stanford.edu
Gabor Angeli[†‡] angeli@stanford.edu
Christopher Potts[*] cgpotts@stanford.edu
Christopher D. Manning[*†‡] manning@stanford.edu

[*]Stanford Linguistics [†]Stanford NLP Group [‡]Stanford Computer Science

Abstract

Understanding entailment and contradiction is fundamental to understanding natural language, and inference about entailment and contradiction is a valuable testing ground for the development of semantic representations. However, machine learning research in this area has been dramatically limited by the lack of large-scale resources. To address this, we introduce the Stanford Natural Language Inference corpus, a new, freely available collection of labeled sentence pairs, written by hu-

for approaches employing *distributed* word and phrase representations. Distributed representations excel at capturing relations based in similarity, and have proven effective at modeling simple dimensions of meaning like evaluative sentiment (e.g., Socher et al. 2013), but it is less clear that they can be trained to support the full range of logical and commonsense inferences required for NLI (Bowman et al., 2015; Weston et al., 2015b; Weston et al., 2015a). In a SemEval 2014 task aimed at evaluating distributed representations for NLI, the best-performing systems relied heavily on additional features and reasoning capabilities (Marelli et al., 2014a).

Figure 3: Annotation setup in INCEpTION. On the left, the annotation editors can be seen; on the right, a PDF viewer shows the publication to annotate directly in the browser.

To answer **RQ 2** and **RQ 3**, that is, to analyze which quality management measures are actually used when creating machine learning (research) datasets and how well works adhere to these, we collected publications that introduced new datasets and annotated them for quality aspects.

4.1 Data Selection

To collect relevant papers, we first attempted to use full-text search in abstracts from papers contained in the ACL anthology (Gildea et al. 2018) for keywords like `dataset`, `corpus`, `treebank` or `crowdsourcing`. This was quickly shown to be infeasible, as our search selected $13,776$ out of $36,501$ publications, showing low precision.

Instead, we chose to leverage *Papers With Code*[9]. This project – among other things – curates a list of datasets used in machine learning research with references to the publications that introduced them. We first selected all text datasets and matched the publication title that introduced it against the ACL anthology. We only considered papers published in top conferences as well as in their respective *Findings* for the following reasons. First, as the annotation is expensive and the budget was limited, this made the annotation more feasible by reducing the overall number of papers to read and annotate. Second, as we are interested in collecting good practices, we hope that these publications that also passed peer review are of higher quality. Publications from the following conferences were considered:

- AACL
- ACL
- CL
- COLING
- CoNLL
- EMNLP
- EACL
- Findings
- LREC
- NAACL
- TACL

This yielded a total of 591 publications to annotate, of which 314 mentioned human annotation or validation. More details about our data selection and the guidelines, in particular the entire annotation scheme, including all the label values, can be found in Appendix 1 and Appendix 2.

4.2 Annotation Scheme

We annotated the following aspects at document level:

Manual Annotation For our analysis, we are primarily interested in scientific publications introducing text datasets that use manual annotation in any form, which is why we annotate this aspect. Manual annotation may serve, e.g., for creating the labels or writing text. This also includes papers that only have human validation.

Task Type There are two task types we consider, *annotation* and *text production*, as they require different methods for quality management. For instance, computing agreement is only possible for the former. Text production also does not lend itself to adjudication.

Number of Annotators The number of annotators per instance whose labels are later adjudicated. This is only annotated for *annotation* datasets, as freeform text usually is not adjudicated.

Mode of Employment We differentiate between *volunteers*, *crowdworkers*, *contractors* and *expert annotators* (§ 3.2).

Quality Management Measures The measures mentioned in the publication to manage quality (§ 3).

[9] https://paperswithcode.com/

Adjudication The method of converting several annotations per instance into a single ground truth (§ 3.5).

Agreement In case inter-annotator agreement was computed, we record the metric's name, the subset size if not computed on all the annotated data, and the actual value. Note that a given dataset can have more than one agreement calculation (§ 3.3.3).

Error rate In case the error rate was estimated, we record the actual value and the size of the subset that was inspected (§ 3.3).

Overall We assign an overall rating to each publication having human annotators based on their quality management conducted and reported. The grades are in three categories:

Excellent Does most of the following: uses the iterative annotation process, trains annotators, computes agreement and error rate, performs extensive validation, and does human inspection throughout.

Sufficient Uses some of the recommended techniques, but not as extensive as excellent. Has at least some validation and manual inspection.

Subpar No agreement, validation, manual inspection, error rate, or other quality management performed and reported. The data quality, at most, relies on aggregating multiple annotations.

We discuss limitations due to the potential subjectivity of this rating in § 8.

A screenshot depicting the annotation editor using this annotation scheme can be found in Fig. 3.

4.3 Bias

Using *Papers With Code* as the source of publications potentially introduces several forms of bias, which we discuss in the following:

Quality As we only analyze publications from top NLP venues and for instance exclude works published in workshops, we suspect that our analysis is biased towards analyzing datasets of better quality.

Time When looking at the distribution over publication years, we see a bias towards more recent publications.

Popularity *Papers With Code* requires volunteers to manually add datasets to the website. Therefore, the resulting collection as well as our analysis might be biased towards more popular and commonly used datasets.

Availability As we analyze annotation quality management by using the publication that introduced it as their proxy, we first rely on that the dataset was described in such a publication and that the publication was accepted in a top venue. Other datasets might not have been published with such an accompanying publication (this is often the case for LDC datasets), or it might have been rejected, making it unavailable for our analysis.

Domain As we only analyze publications from general venues and not specialized venues like workshops for narrower domains as legal or medical NLP, our collection might be biased to contain datasets that are of more general interest; particular domains might be underrepresented.

In order to quantify the bias and to estimate how well Papers With Code (PwC) covers the ACL anthology, we additionally annotated a random subset of 500 papers from the years 2013 to 2022 for the datasets they use. 2013 as the minimal year is chosen as older datasets are for the most part not covered by PwC (see Fig. 4). 2022 as the maximum year was chosen as our snapshot of PwC is from the 26th of November, 2022 (see Appendix 1). Again, we limited ourselves to the aforementioned top conferences and sample 50 papers per year randomly, resulting in 500 papers total. We annotated for two aspects: datasets used in the publication and whether a publication introduces new datasets. Datasets were marked as not relevant if they do not contain dataset usage or use any other modality than text. Subsequently, we deduplicated dataset mentions and linked them to PwC in case they have an entry there. The coverage analysis can be found in § 5.1.

4.4 Annotation Process

The annotation process we used was the same for both quality and coverage annotations. It slightly deviates from our best practices due to limited time and money. We downloaded the full-text PDFs of the selected paper and annotated them in *INCEpTION* (Klie et al. 2018). This annotation tool was chosen because it is free to use and supports annotating PDF documents out-of-the-box. The annotations were created by the first author of this work, an experienced researcher in NLP with a strong data annotation background.

We first conducted an initial pilot study to determine the aspects to annotate, followed by the annotation itself. The tagset was iteratively extended during the annotation process. After all papers had been annotated once, we did a second round to make the annotations more consistent with the now complete tagset. Finally, we did another validation round and additionally used semi-automatic checking to improve consistency and quality further. Thus, each publication was only annotated by a single author but inspected several times to guarantee correctness and consistency. Due to the intricate and complex annotation scheme with many aspects, the expertise needed, and the exploratory nature of the annotations, we were only able to employ a single expert annotator. Instead, we opted for repeated validation and correction. In total, annotation alone took over 100 hours. While not ideal, this is a similar setup as used in previous works surveying NLP publications (Sabou et al. 2014; Amidei, Piwek, and Willis 2019; Dror et al. 2018; Shmueli et al. 2021).

5. Analysis

After having annotated a large corpus of dataset introducing data, we now use it to investigate how annotation quality management is practiced quantitatively (RQ 2) and qualitatively (RQ 3). An overview of the overall usage of each method can be found in Table 1. Regarding recommended good practices, it must be noted that there is no way of managing the dataset creation process that guarantees high-quality results. Nevertheless, some methods have been shown to yield better quality than others Bayerl and Paul (e.g., 2011); Monarch (e.g., 2021). These choices of how to manage quality have to be looked at in the context of the task to annotate for and the constraints at hand, for instance, concerning available budget, time constraints, annotator number, and experience.

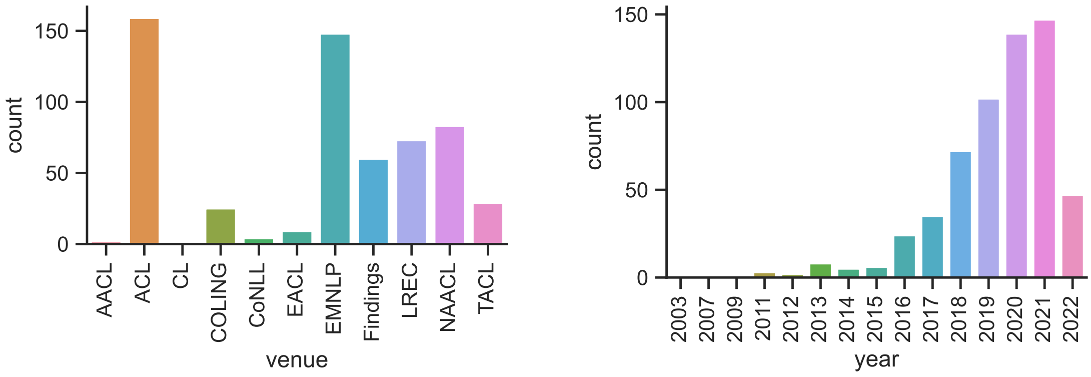
Our analysis is based on what is explicitly reported in the publication; if it was not reported, we are unable consider it. While this might cause our analysis to be less expressive and accurate, we see no simple way to study quality management in practice. Also, this issue further emphasizes the importance of proper reporting, even if it is just in an appendix or supplementary material.

Category	Method Name	#	%
Annotation Process	Agile Corpus Creation	68	22
	Pilot Study	67	22
	Validation Step	125	41
	Data Filtering	46	15
	None/Not specified	96	32
Annotator Management	Qualification Filter	80	26
	Qualification Test	56	18
	Annotator Training	55	18
	Annotator Debriefing	18	6
	Monetary Incentive	13	4
	None/Not specified	157	52
Quality Estimation	Error Rate	54	18
	Control Questions	28	9
	Agreement	156	52
	None/Not specified	102	34
Quality Improvement	Correction	68	22
	Scheme and Guideline Refinement	31	10
	Annotator Deboarding	39	13
	Annotator Feedback	24	8
	Agreement Filtering	29	9
	Manual Filtering	16	5
	Time Filtering	11	3
	Automatic Checks	34	11
	None/Not specified	135	45
Adjudication	Manual Curation	29	14
	Majority Voting	68	34
	Probabilistic Aggregation	2	1
	Unknown	92	46
	Other	5	2

Table 1: Overview of how often each quality management (see also Fig. 1) method was used in absolute numbers (#) and relative to all works that used manual annotation (%). For adjudication, the denominator is the number of publications for which adjudication is applicable. Except for agreement, validation, and error rate, counts are directly computed from the *Quality Management Measures* field of our dataset. For the other methods, we count it for the respective metric if there is at least one usage mentioned. Note that values are non-exclusive, as publications can make use of any combination of methods.

5.1 Dataset Statistics

Quality Statistics. In total, we selected and annotated 591 publications. These were organized into three groups based on the amount of human involvement. 277 did not report any human annotation for their dataset creation. In these cases, annotations were crawled or obtained via distant supervision or other means. 16 relied on humans to



(a) Publications per venue

(b) Publication count over time

Figure 4: Statistics over the dataset created by annotating text dataset introducing publications obtained from *Papers With Code*.

validate their algorithmically created data. 298 had humans annotating or producing the text. Of these 298 publications, 81 were introducing datasets that used annotators only for text production, 161 for labeling and 56 for both. Datasets that leveraged both text production and labeling were often created for tasks like natural language inference or question answering. There, the surface forms were usually written by workers before their relationships were annotated in a follow-up step.

The resulting dataset size exceeds Dror et al. (2018) who inspected 233 papers for their analysis of statistical testing in NLP research, as well as Amidei, Piwek, and Willis (2019) who inspected 135 publications for analyzing agreement in the context of natural language generation evaluations. The distributions of publications per venue and over time are depicted in Fig. 4. It can be seen that most were published in or after 2018.

Coverage Statistics. Papers With Code (PwC) only contains entries for a subset of dataset-introducing publications. To analyze the coverage and to better understand the potentially resulting bias (see § 4.3), we conducted another annotation of 500 papers from the anthology from the years 2013 – 2022 for their dataset usage. Based on these annotations, we first of all can see that 430 of publications mention relevant dataset usages.[10] 132 (30%) publications introduced new datasets of any kind.

In total, we found 993 mentions of 622 unique datasets, 495 datasets are only mentioned once. Of the 622 unique datasets, 172 (27%) are also contained in our dump of PwC. When taking our filtering of publication venues into account, we see that from the papers that we annotated for quality management, 49 of all papers and 30 of relevant papers are in the sample annotated for coverage as well as in the sample for quality. In relation to our quality dataset, these make up 8% of all and 10% of relevant publications.

To better understand the popularity of the annotated datasets, we analyze their mention frequency. We can see that on average, a dataset in the coverage sample was mentioned 1.60 times. In the sample for quality annotations, this was 1.96 for all publications and 2.13 for only the relevant ones. While not being a large difference, this

[10] The following metrics are with respect to relevant publications only.

still indicates that our sample based on PwC is slightly biased towards more popular datasets.

Finally, we find that our dataset in particular does not cover most LDC corpora or datasets introduced as part of shared tasks. These are, for instance CoNLL, WMT, SemEval, or TAC.

Bias. We used PwC in order to reduce the effort of finding publications that introduce new datasets in the first place. The aforementioned statistics indicate that our sampling using PwC introduces biases towards more popular, more recent and on average, higher quality dataset. While not ideal, we argue, however, that this is not necessarily a disadvantage, as the datasets that we analyzed are actually frequently used in practice. Thus, their quality has direct impact on the research community. Also, with being more popular, we hope that their quality management also follows good practices comparatively more often. While having a seemingly low coverage overall, our sample size nonetheless is much larger compared to previous work, still yields interesting insights, and was already costly to annotate.

Bias in time, popularity or domain might be an issue, as there could be practices from the past that are falling through our cracks that would be relevant and interesting for the general public. We alleviated this issue by also surveying other literature like books and by collecting and analyzing a large corpus of dataset-introducing publications.

Our analysis of annotation quality management it is still a valuable contribution, especially in combination with our survey of good practices and a good start for future work. Also, we are interested in finding issues and to offer solutions for their alleviation, having unbiased counts is desirable but not crucial. We hence suspect that the statistics derived overestimate quality compared to the general populace and that our analysis are potentially too positive. The statistics that follow thus should be seen as an optimistic estimate. Finally, it has to be noted that the the resulting dataset is a side product of the survey and should be seen in this context. While we have taken the utmost care during annotation, the dataset is not intended to be used in machine learning or other areas where quality needs to be very high and absolute.

5.2 Overall

To better understand how well quality management is performed in practice (RQ 3), we assigned each work an overall score. Their distribution is depicted in Fig. 5. It can be seen that around 45% of publications perform well, and 25% employ excellent quality management according to our annotation scheme and guidelines. However, we also find that circa 30% only conduct subpar quality management. These often either did not report the annotation process at all or just very briefly and did not mention that they applied any quality management.

5.3 Annotation Process

In the following, we analyze the publications concerning their annotation process.

Annotation Scheme and Guidelines. Of the 298 publications having human annotators, 68 (22%) reported having an iterative refinement loop, which is our recommended annotation process. This loop was mainly used for iteratively refining the annotation guidelines after doing pilot studies (10%) or repeatedly correcting instances until they

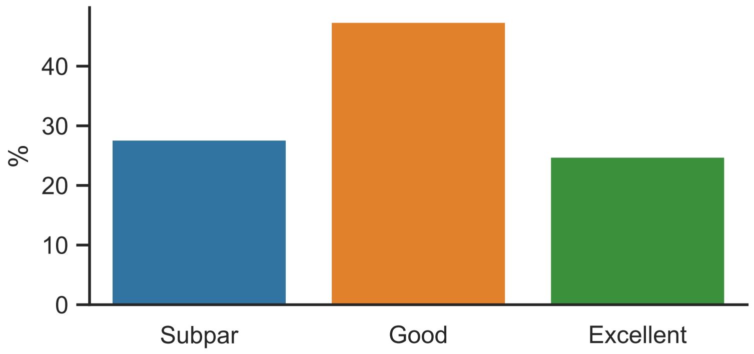


Figure 5: Distribution of percentage of papers over subjective quality management quality. Mostly, quality management was good or excellent, but a large fraction is only subpar.

reached sufficient quality (12%). 18 (6%) works reported that their annotators gave feedback on the task during annotation so that the annotation process could be improved.

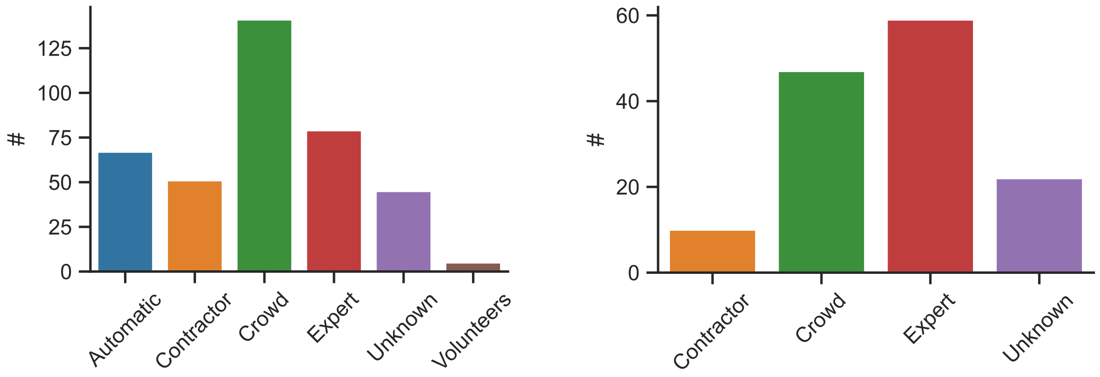
60% of publications with manual annotation described their annotation scheme, showed their annotation interface, or published their annotation guidelines together with the dataset itself in some form. Not reporting annotation schemes and guidelines causes several issues. First, these cannot be checked and reviewed, making it difficult to assess their quality. Second, not making it available is a significant obstacle to reproducibility or later extensions. In several cases, the reader was referred to supplementary material or appendices, which we could not find in the publication or online.

Pilot Study. Overall, only 22% of the publications mentioned to have conducted a pilot study. This value is relatively low, as pilot studies are an essential tool to dial in the annotation scheme and guidelines and to get feedback from the annotators. As we only rely on what is mentioned in publications, we cannot say whether the authors considered this method common and thus did not see the need to mention that they conducted a pilot study or that it is indeed not done often enough.

Validation. In many cases, annotations were validated as an additional step in the overall process either by the annotators themselves or by having experts check them (41%). For automatically annotated data, only 16 out of 293 reported that they employed human validators. Not validating can be an issue; for example, datasets created solely by distant supervision can contain many labeling errors (Mintz et al. 2009). 10 of these publications also reported the resulting error rate, which ranges from 1.40% to 16.60% with mean 8.93% and median 8.55%, showing the importance of validation. We found 25 publications that reported indirect validation (8%).

5.4 Annotator Management

The distribution over different annotator types is shown in Fig. 6. Overall, publications mostly used crowdworkers or experts for their annotations. For validation, experts were more commonly selected. In many cases, the kind of annotators used was also not reported.



(a) Absolute number of annotators by type.

(b) Absolute number of validators by type.

Figure 6: Distribution over annotator types. For annotation (a), crowdsourcing is used the most; for validation (b), it is experts. Note that a publication, respectively dataset, can leverage more than one annotation type.

We find that the preferred method to filter out annotators, especially crowdworkers, is by requiring a certain number of previous successful tasks and a high acceptance rate (26%). Qualification tests, recommended by Kummerfeld (2021) over filters, are also often employed (18%). Annotators are given training only in 18% of cases, which we find pretty low compared to the benefits it might give. Out of these cases, training was overwhelmingly given to contractors and crowdworkers; only one publication mentioned that experts were trained. We note, however, that even experts should be given training, as being an expert does not automatically indicate familiarity with the annotation setup and scheme at hand (Bayerl and Paul 2011). Only in a few cases (8%) is it explicitly stated that annotators were given feedback on their work or that annotators give feedback to improve the annotation process (6%). While not being reported, we assume that training and feedback were given in many more cases, especially for contractors. Better interaction between project leads and annotators is one reason contractors are typically chosen over crowdworkers. 13 (4%) publications mention some kind of additional monetary incentive.

5.5 Quality Estimation

The quality of the dataset created needs to be estimated during and after its creation so that its quality can be guaranteed and countermeasures can be taken to improve it if needed. Overall, we find that two main techniques were used for this, which are agreement (52%) and error rate estimation (18%). We analyze these in more detail in § 5.8 and § 5.9, respectively. Control questions were used by 9% of the publications to gauge annotator performance and task quality. Overall, 65% of works mention at least one way of estimating quality.

5.6 Quality Improvement

Next, we analyze rectifying measures used to improve the data quality after it has been estimated in a previous step and deemed insufficient. In most cases, incorrect or low-quality instances are corrected (22%) or filtered out (15%). Of the 46 publications that mention filtering, 29 report filtering based on agreement, 16 after manual inspection, and 11 based on unsound, improbably low annotation times. 11% of publications mentioned to have applied some kind of automatic checks to identify potential errors, such as spell checking or hand-crafted rules. Sometimes, annotators were removed from the workforce if they repeatedly delivered sub-par quality (13%). Rarely were they given feedback by experts or the project managers (8%). This number increases to 22% when excluding datasets only annotated by experts. Overall, we do not see much usage of rectifying measures; only 41% of publications using human annotation report at least one.

5.7 Adjudication

Similarly to Sabou et al. (2014), we find that majority voting was most often used to adjudicate labels (34%). In a few cases, publications reported that in addition to majority voting, ties were broken by consulting additional workers or experts (8%). The second most common way of adjudication was manual curation (14%). Overall, we find that in 46% of labeling datasets, adjudication methods were not reported clearly or at all. This leaves the reader to guess, which is concerning.

We only found two publications that used *Dawid-Skene* (Dawid and Skene 1979) and one that used *MACE* (Hovy et al. 2013). The latter was just used to filter out spammers during annotation and not for adjudication itself. One publication mentioned trying out probabilistic aggregation, yet they report that just using majority voting yielded better results for them. Some works also mentioned aggregation based on annotator confidence and skill, but no details were given describing the exact procedure used.

The fact that majority voting is by far the most frequently used method is interesting, as aggregation is a quite well-researched topic in the crowdsourcing research community (Sheshadri and Lease 2013). It has also been shown that using more intricate methods can create higher-quality gold standards (Paun et al. 2018; Simpson and Gurevych 2019).

5.8 Error Rate

While it is often assumed that (research) datasets represent a gold standard and do not contain errors, this is often not the case (e.g., Northcutt, Athalye, and Mueller 2021; Klie, Webber, and Gurevych 2023). To estimate the overall correctness of the dataset, its annotation error rate should be computed after adjudication is completed. Computing the error rate is typically done by randomly sampling a subset and marking instances as correct or incorrect. From our analysis, only a few publications (18% of all having human annotation) estimated and reported an error rate. The average error rate reported is 8.27%, and its median is 6.00%.

Sample Size. From the dataset we analyzed, 64 out of 80 error rates were computed by inspecting only a subset of the data. The inspected subset needs to be of sufficient size for the estimate to be reliable. If it is too small, the estimate has large error margins and hence low statistical power, potentially leading to over-optimistic or incorrect conclusions (Button et al. 2013; Passonneau and Carpenter 2014).

For instance, it was found that TACRED (Zhang et al. 2017), a dataset for relation classification, contains a large fraction of incorrect labels. During the dataset creation, 25% of the annotations were validated by crowdworkers; after adjudication, the authors finally inspected a sample of 300 instances and estimated an error rate of around 6.7%. It was then subsequently discovered that the dataset contains significantly more errors. First, it was claimed to be around 50% by Alt, Gabryszak, and Hennig (2020), who only analyzed a smaller and biased sample. Stoica, Platanios, and Poczos (2021) finally inspected all samples and found an error rate of 23.9%. This shows the importance of manual inspection of large enough sample sizes.

In the publications inspected, we did not find any work that based their choice of sample size on a statistical footing or gave reasoning for selecting that specific value. In most cases, pretty numbers were chosen without rationale (e.g., round numbers like 100 or 200 were picked often), or a percentage of the total size (e.g., 5%) was used. The mean sample size is 1305.68, while its median is 200.00 (see Fig. 7).

We also analyze the impact the sample size has on the estimate's reliability using confidence intervals and their interval half-widths. The interval half-width measures the margin of error associated with the confidence interval. It is computed as the largest distance between the point estimate of the error rate and its endpoints. The confidence interval for an estimated error rate $\hat{r}$ is then given as $[\hat{r} - h, \hat{r} + h]$. If h is relatively large, e.g., 0.05, then the error rate is with high probability within ± five percentage points. This is quite a large margin, especially for error rates, as $\hat{r}$ is usually small there and (hopefully) close to zero.

To compute the margin of error, we model estimating the error rate as sampling with replacement[11] where annotators randomly inspect a subset of instances and mark them as either correct or incorrect. For each mention of error rates in our analyzed publications, we then compute a 95% binomial exact confidence interval for each estimate and its half-width h.

The half-widths for each estimate are plotted in Fig. 7. For almost all estimates, the resulting confidence intervals are very wide, rendering a given point estimate statistically unreliable. When choosing a different sample to inspect and mark, the error would fluctuate by a large margin and has thereby only limited explanatory power. We suggest inspecting at least 500 instances[12] or the whole dataset, whichever is smaller, for a more sound estimate. Note that calculating the sample size that way is an optimistic estimate, as it assumes independent and identically distributed instances, which is often not the case. Also, giving a confidence interval when stating the error rate is recommended. This can either be done by computing a binomial/hypergeometric confidence interval or using techniques like bootstrapping. Otherwise, giving a point estimate implies precision which it has not, especially when giving several decimal places.

5.9 Agreement

For every paper inspected, we annotated whether agreement measure usage was mentioned and recorded its type and value if it was. In most cases, agreement has been used to demonstrate the dataset quality after the annotation was completed. Sometimes, agreement has also been used to either remove annotators or remove annotations. We observe that 52% of publications involving human annotators reported using at least one form of agreement. Concerning the form of dataset creation, it is 48% for labeling

[11] The sample size is usually much smaller than the dataset size, which is why we can approximate the hypergeometric distribution (sampling without replacement) with the binomial distribution for simplicity.

[12] Assuming a binomial model with a true error rate of 5%, a sample size of 456 yields a 95% CI with $h \approx 0.02$

and 31% for text production. In addition, we find that 7 publications that —while not employing humans for the annotation itself— leverage agreement during validation steps. The usage statistics are depicted in Fig. 8. Overall, Cohen's and Fleiss's κ, Krippendorff's α, and percent agreement were used the most, followed by F_1. On average, each publication used 1.33 agreement measures with median 1 (based on works that actually used at least one). Percent agreement as the only measure was used in around 11% of all publications that use at least one method. Only using percent agreement makes it difficult to estimate, interpret, and compare the dataset's quality, and its usage is therefore discouraged (Krippendorff 2004). In 10 cases, the used measures were not clearly named but only referenced as e.g. κ or IAA (this is noted by a '?' in Fig. 8).

Regarding the usage and reporting of agreement as an indicator for reliability, we found similar issues as described by Amidei, Piwek, and Willis (2019). Often, only the agreement value was stated without any interpretation or comment (52%), which limits its explanatory power. In many publications, the quality derived from the agreement was described with a freeform explanation, e.g., *high*, *fair*, *substantial* (27%). These frequently do not have a relation to the actual value, as, for example, values < 0.3 were described as *reasonable*. Rarely was agreement compared to previous studies (5%) or an interpretation based on a range given by the literature was cited (16%). This can partially be explained by only some datasets having a suitable predecessor as a reference.

In all cases, these ranges' limitations were not considered; for example, the ranges defined by Landis and Koch (1977) are based on binary classification. In contrast, several datasets introduced by the respective publications had more than two possible labels. Also, several times, the stated ranges did not match the metric. For example, the ranges from Landis and Koch (1977) that apply to Cohen's κ were instead used for Fleiss' κ. Several times, publications used pairwise agreement measures for more than two annotators and reported them pairwise. While that is valid in itself, additionally

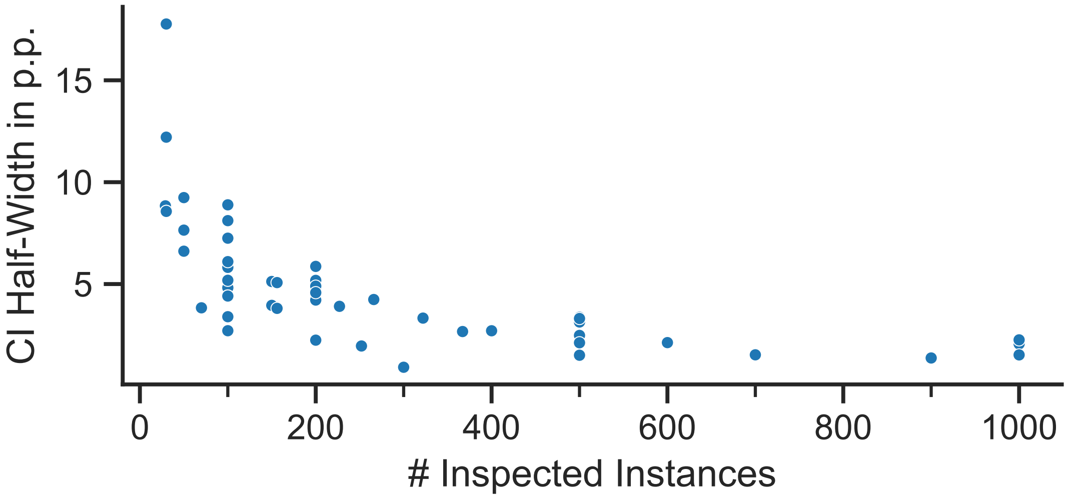


Figure 7: Number of inspected instances vs. the resulting confidence interval (CI) half-width for a 95% CI. It can be seen that overall, too few instances are inspected to estimate the error rate reliably, as they have a substantial margin of error. Four values above 1000 were filtered out to aid the visualization.

using multi-user measures like Fleiss' κ or α is recommended. We also found several cases where the usage of Cohen's κ was reported, but more than two annotations per instance were obtained. It is also discouraged to use correlation metrics as a measure of agreement. We found 7 (2%) of publications that still reported its usage. Last but not least, κ or α was sometimes given in percent. This can confuse the reader as these values are usually given as a value in $[-1, 1]$, and percent agreement is a distinct metric on its own.

Agreement values. We plot the agreement values for the most frequently used methods in Fig. 9 together with the boundaries suggested by the literature (even though they are often subjective). For Krippendorff's α, the values are rarely larger than 0.8, which would indicate acceptable agreement according to Krippendorff (2004). Some are in the zone $(0.67 \leq \kappa \leq 0.8)$, which indicates that the resulting annotations should only be used to draw tentative conclusions; the majority is even below that. Many agreement values are on the lower side, hinting towards lower agreement or considerable ambiguity in the underlying task.

Agreement for Sequence Labeling. For sequence labeling datasets (e.g., Named Entity Recognition or Slot Filling), dataset creators either did not compute agreement or relied on per-token κ, α, or classification metrics like precision, recall, and mainly F1. Brandsen et al. (2020) argue that per-token agreement for sequence labeling comes with two issues. First, annotators label sequences and not tokens, so the measure does not reflect the task well. Second, the data is imbalanced, as most tokens are labeled O, indicating no span. Excluding this would result in an underestimate of the agreement. They argue for using F_1 and averaging it between annotators. However, this is not chance-corrected and can only be used to compute pairwise agreement; averaging might lead to a loss of information. Only a single paper (Stab and Gurevych 2014) used Krippendorff's unitizing α_u (Krippendorff 1995) to compute agreement for sequence labeling. α_u in itself can directly support sequence labeling and is an excellent way to compute agreement in this setting. We hence agree with Meyer et al. (2014) that unitizing agreement measures should be used if not as the only measure, then at least additionally. Our conjecture for why unitizing measures are not used more often is that these are not very well-known, and their complex implementation hinders adoption.

Sample Size. Dataset creators sometimes decided only to have one annotation per instance for the majority of the dataset to save resources. Then, only a subset was annotated multiple times to compute the agreement. Similar to Passonneau and Carpenter (2014) and as described in § 5.8, we note that having too small sample sizes is an issue as even a relatively relaxed 95% confidence interval spans quite a wide range of values. A sample size that is too small can cause estimates to vary by a large margin. This might lead to a different interpretation based on a pre-determined, targeted agreement level or a range suggested by the literature.

Out of 288 papers that reported agreement values, 197 have had the complete dataset annotated multiple times, 91 were computed from a subset. The mean sample size for the latter was 1882 with median 200. 47 (51%) agreement values were computed on 200 instances or less, 26 (28%) even on less or equal than 100.

It is therefore recommended to 1) have large sample sizes to compute agreement on, ideally the complete dataset (which has the advantage of improved quality due to aggregation) and compute a confidence interval for the agreement value, e.g., by bootstrapping (Efron and Tibshirani 1986; Zapf et al. 2016). Computing the required sample

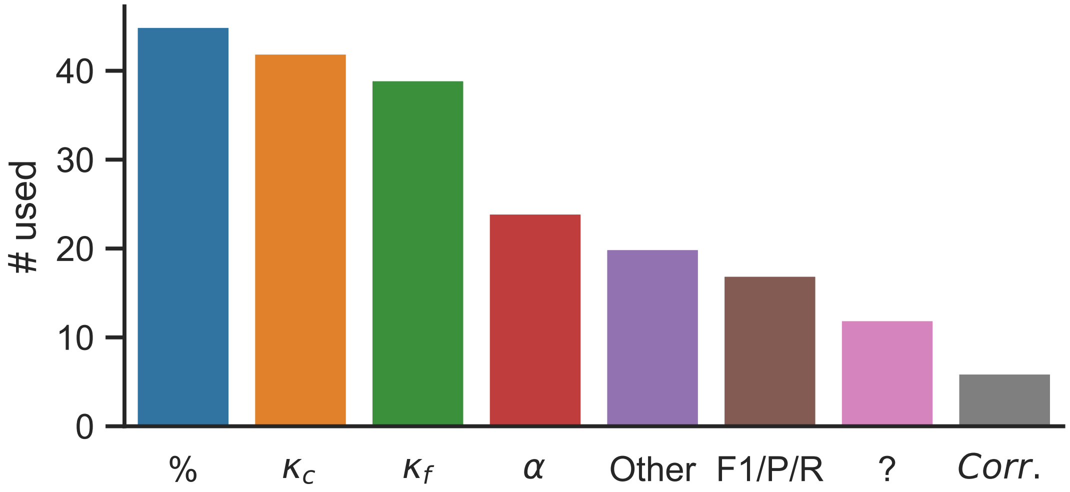


Figure 8: Distribution over counts of the agreement measures used. We count each method only once per publication, even if it has been used more than once. Overall, agreement measures were used in 156 publications involving human annotators.

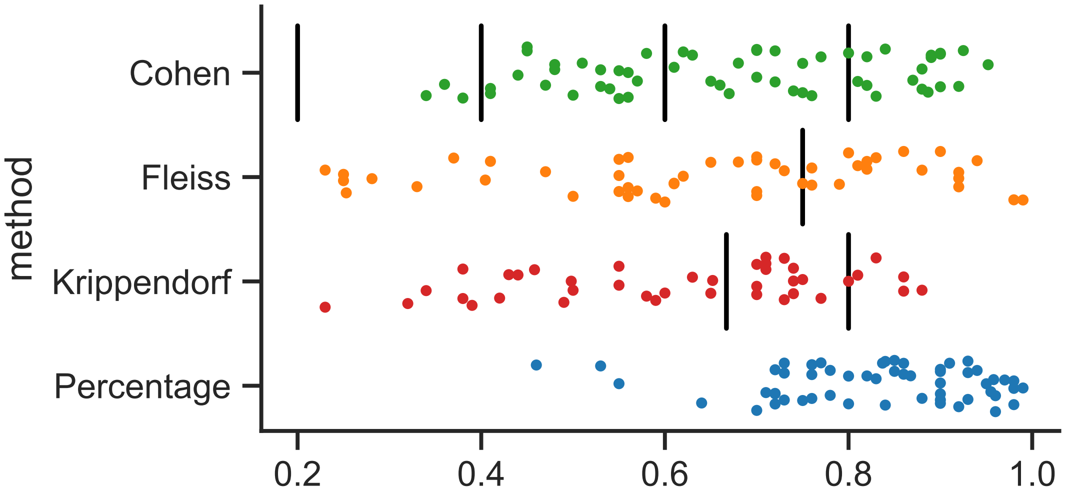


Figure 9: Agreement values for the papers inspected. Also shown are the ranges often used for interpreting these values.

size for a given precision and confidence level is not straightforward and depends on the metric (Shoukri, Asyali, and Donner 2004). For Cohen's κ, an approximation is described by Donner and Eliasziw (1992); for α, it is given by Krippendorff (2011). As a rule of thumb that works for both κ and α, given an expected/desired agreement value of 0.8 with a precision of $h \pm 0.05$ and a confidence level of 95%, at least ≈ 500 instances should be annotated.

While this is highly desirable, we notice that this comes with costs and additional effort. We did not find a single report of confidence intervals for agreement values in the publications analyzed for this work. As we do not have access to the raw, unadjudicated data used to compute the agreement value (which is needed for computing confidence intervals), we cannot easily conduct an analysis similar to the one for error rates in § 5.8.

6. Recommendations

Based on our analysis of 591 papers published in top NLP conferences as well as on our survey of the relevant literature, we derive the following recommendations and good practices for dataset creation quality control. A case-by-case ranking of measures should be done based on the circumstances of the project.

Annotation Process.

- Use an agile, iterative annotation process and annotate in batches (Alex et al. 2010; Pustejovsky and Stubbs 2013).
- Conduct pilot studies to validate the annotation setup before starting the actual annotation.
- Quality estimates after each batch should guide the improvement of guidelines and the scheme.
- Rectifying measures like corrective annotation, annotator retraining, or data filtering should be used to improve the overall data quality iteratively.
- Annotator feedback should be incorporated during a pilot study and annotation.

Annotator Management. Workforce selection and annotator management are crucial for a successful annotation project. Different annotator types can be viable depending on the task difficulty and the expertise or background knowledge required. Datasets these days are most often annotated by crowdworkers. A feasible alternative (even for tasks that usually require expert annotators) is hiring and training contractors via platforms like Upwork or Prolific. This can open up better ways to collaborate while having similar costs.

- The choice of annotator type (expert/contractor/crowdworkers, …) should be validated as part of a pilot study.
- Annotators should be paid properly and treated with respect.
- They should be trained before and during the annotation process for the best results, even experts.
- Annotator feedback should be used to fine-tune the guidelines, annotation scheme, or annotation editor and to spot errors or issues like low data quality.
- To select annotators, qualification tests are the recommended way; criteria like completed tasks or acceptance rate can be an addition, but should be rather lower than higher to not force workers into low-paying qualification jobs.

Quality Estimation. Precise quality estimation is essential to steer the annotation process after each batch and before the final release of the dataset.

- Inter-annotator agreement can be used to determine whether the annotation process is overall reliable.

- In addition to agreement, manual inspection is recommended to validate annotations and estimate accuracy. This can be done by either the annotators themselves or experienced/expert annotators.
- Disagreements can be visualized using confusion matrices.
- An alternative to having annotators validate instances by marking them correct or incorrect is to have an additional task after the annotation/instance creation itself.
- Control instances can be injected into the data to annotate for measuring individual annotator performance and batch quality.

Agreement. Agreement can be used to gauge how reliable the annotation process can be. High agreement, however, does not automatically guarantee high-quality annotations and should be used together with other quality estimating and improving measures, like validation between annotation rounds or error rate estimation after adjudication. Krippendorff's α can be used in almost all circumstances, even for sequence tagging in the form of unitized α (Krippendorff 1995), continuous judgments, or with varying numbers of annotations per task and is therefore recommended. The agreement value targeted should be chosen beforehand, either by pilot (expert) studies or previous annotation studies annotating similar tasks. When the same number of annotators annotates each instance, Cohen's κ for two annotators or Fleiss's κ for multiple annotators can additionally be used, the latter only if annotators are randomly assigned to instances. Percent agreement should rarely be used and never the only employed agreement measure. Correlation coefficients like Pearson's r, Spearman's ϱ, or Kendall's τ should not be used to assess reliability. Instead, Krippendorff's α or intraclass correlation is recommended as an alternative.

For a reliable estimate, agreement should be either computed on the whole dataset, or a sufficiently large ($\gtrapprox 500$ instances) subset should be annotated by multiple annotators. Subset sample sizes should be statistically grounded, for instance, by computing them based on confidence intervals. They should also be justified in the dataset description. When using agreement, its usage should be reported in detail. The documentation should include which measures were used and why, how many judgments per instance were obtained, the background of the annotators, and the sample size used. Agreement values require interpretation and should not stand alone. This can be done by defining a target agreement value, for instance, based on an expert study before the annotation itself, using a sufficiently high value like 0.9, or comparing it to previous works. Using thresholds from the literature like the ones from Landis and Koch (1977) is not recommended, as these are arbitrary. Confidence intervals should be employed to gauge the confidence of the agreement computation, whether they are reported as closed-form solutions given by the coefficient or via bootstrap. More recommendations concerning agreement usage can also be found in the conclusion of Lombard, Snyder-Duch, and Bracken (2002).

Quality Improvement. Annotations are often not good enough at the beginning of an annotation project. Therefore, estimating the quality and taking quality improvement steps is essential. These can be, e.g., to correct low-quality instances or filter them out, improve guidelines and the annotation scheme, or train annotators. Underperforming or adversarial annotators can be removed from the annotation project if required.

Adjudication. Ideally, each instance should be annotated by multiple annotators in order to compute agreement and increase reliability via adjudication. Majority voting is a strong baseline for aggregation; using more sophisticated approaches like Dawid and Skene (1979) or MACE (Hovy et al. 2013) might be worth trying, especially in settings where individual annotators are underperforming, or spammers are potentially prevalent. Alternatively, expert curation or majority voting with experts breaking ties can be used to create a high-quality gold standard. For reproducibility and better error analysis, it is suggested to not only publish the adjudicated corpus but also annotations by individual annotators. These can then also be used to study and learn from the disagreement (Uma et al. 2021).

Error Rate Analysis. During and after the data has been annotated, it is crucial to have experts check the actual percentage of errors. The sample size should be large enough to reach a high confidence estimate, which usually requires at least 500 instances (see § 5.8) to inspect. This sample size should be computed by considering the desired statistical guarantees, for instance, confidence level and estimated precision.

Reporting. We urge authors to accurately report on the annotation process when creating new datasets. This includes, among others, annotator type and background, number of annotators, number of validators, dataset and subset sizes, agreement measures and values, adjudication methodology, and error rates. In addition to that, we suggest augmenting the dataset documentation and reproducibility checklists (which are at the time of writing mainly concerned with model training and have only a few, if any, sections for dataset quality, see § 2), often required when submitting papers to conferences, with a section that is targeted with questions towards quality management good practices. The checklist from Kottner et al. (2011) can be a good start for checking and guiding dataset creators toward the proper use of agreement.

7. Conclusion

High-quality datasets are essential for —among others— deducing new knowledge, for policy making, and to suggest appropriate revisions to existing theories. They are also crucial for training correct and unbiased machine learning models. If trained on datasets containing errors, inference can lead to wrong or biased predictions, which can cause material damage or even harm to other humans. These potential issues are especially relevant with the recent, widespread adoption of conversational agents based on instruction-finetuned large language models. Using datasets containing errors for evaluation can lead to incorrect estimates of task performance and, thus, to wrong conclusions when comparing models or approaches.

Quality management is an essential part of creating high-quality annotated datasets. Therefore, we set out to better understand which methods exist (RQ 1), which methods are actually applied in practice (RQ 2), and how thorough (RQ 3). For this, we surveyed the literature and inspected 591 publications introducing new datasets from which 314 reported human annotation or validation, which we annotated for their quality management usage.

We answered our first research question by summarizing good practices for annotation quality management (§ 3). These are methods suggested in the literature or commonly used during dataset creation. Then, we used the dataset of annotated publications for their quality management to investigate which methods are used frequently and which are not. Finally, we rated each publication for how well overall they conducted their quality management. We found that, on the one hand, many works implement good practices very well. On the other hand, there are still issues that need to be improved on, for instance, better usage of agreement, annotator management, quality as well as error rate estimation, or reporting. To be more precise, many papers used agreement without interpreting it, making it difficult to understand its implications. Error rate and agreement were often computed on too small sample sizes, which renders the value imprecise and less expressive. Frequently, annotation guidelines were not published, hindering reproducibility.

We conclude that many widely applicable techniques should be used more often or their use properly reported, especially iterative corpus creation as the annotation process of choice, pilot studies, validation, annotator training, qualification tests, control questions, annotation feedback, and debriefing, and maybe more complex adjudication.

We hope that our recommendations foster an adoption of good practices and an increase in dataset quality in the future.

Future Work. In this paper, we analyzed 591 scientific publications introducing new datasets and annotated them for their annotation quality management. We see several ways to build on this work. First, while we already annotated a sizeable corpus of publications, using *Papers With Code* introduced bias, limits analyzing quality management to what is reported in the paper and only contains a subset of dataset-introducing publications. Therefore, we see the next step in a larger scale effort, ideally by directly asking authors to fill out a structured survey questioning them about their quality management. While it might be difficult retroactively, it can be a good way for new datasets, especially when it is done as part of the publication and peer review process itself. Second, it would be interesting to graph how quality management evolves over time and to analyze trends. For instance, Meyer et al. (2014) state that agreement was not used very often in their small-scale analysis at the time, but we see that, on average, it is now used quite frequently. Third, we only annotated which methods were used, but not what their actual, quantifiable impact was. Hence, conducting such studies, similar to Bayerl and Paul (2011) would be insightful, which analyze which factors contributed to higher agreement. Fourth, as our work mainly focused on annotation and less on text production, we would like to see an extension in that direction. Fifth, in this work, we focused on analyzing scientific publications concerning their quality management. We leave analyzing other aspects for future work, for instance, how well publications adhere to aspects checked for in dataset documentation or reproducibility checklists. Sixth, it would be compelling to annotate the dataset by introducing publications on a large scale to alleviate the issues that our biased sampling might have caused. This can then also be extended to other areas of machine learning, like computer vision. Finally, we recommend that conference organizers and steering committees develop and adopt a dataset quality management checklist similar to existing ones and cover aspects like bias, intended use, or reproducibility.

8. Limitations

In this work, one of our goals was to analyze how quality management of annotated datasets is done by inspecting and annotating the publications that describe their creation. Our analysis already yields several relevant findings and common issues. We also were able to derive recommendations that future dataset creators can leverage for their own annotation projects. However, we did not analyze the impact these practices have on the resulting dataset quality. It is an interesting problem (but complex, as it requires manually analyzing not only the publications but also the datasets themselves) extension that we leave for future work.

We chose *Papers With Code* as the source of publications to annotate. While our collection approach introduces bias and does not find all publications presenting new datasets, the papers annotated this way are for popular and frequently used datasets. Otherwise, they would not be listed in *Papers With Code*. Our annotation still captures an important slice of quality management directly impacting research and state-of-the-art evaluation. However, a larger-scale annotation project would be the logical next step.

Our analysis relies on publications reporting their quality management. Hence, there might be a non-negligible underestimate of the numbers presented here. New publications are inspired by how established datasets conduct their annotation process; therefore, even if good quality management is conducted, non-reporting is also an important issue that needs to be pointed out.

Our study is limited to primarily academic datasets and may have a blind spot in the industrial field, not only in terms of data but also in terms of methods. However, this issue is difficult to alleviate, as industry datasets are often publicly unavailable.

The dataset is not intended to be used in machine learning, but is used to empirically underpin our survey. Due to limited resources and the difficulty of the annotation task, each publication was only annotated by one annotator. The impact on quality and consistency was reduced by repeatedly validating the annotations and using automatic rules to clean and improve them. Ideally, more than one set of annotations would be available to compute agreement, adjudicate, and find errors, which we recommend for the next time.

For the overall rating, when conceiving the annotation guidelines and the scheme and during annotation, we tried our best to make it as objective as possible. We still admit that the distinction between *excellent* and *sufficient* is relatively fluid. However, we argue that our definition is relatively objective for *subpar* quality management, which is the most relevant category for this work. We were relatively lenient during annotation and assigned a better rating in case of doubt. To further reduce the issue of subjectivity, we thought of alternatives like assigning scores based on the number of quality measures and their relative importance. However, we ultimately abandoned this idea because not all works can use each measure, and we would have swapped one kind of subjectivity with another.

Acknowledgements

We thank Falko Helm, Ivan Habernal, Ji-Ung Lee, Qian Ruan, Nils Dycke, Max Glockner, and our anonymous reviewers for the fruitful discussions and helpful feedback that improved this article. This work has been funded by the German Research Foundation (DFG) as part of the Evidence (grant GU 798/27-1) and the PEER project (grant GU 798/28-1).

References

Alex, Bea, Claire Grover, Rongzhou Shen, and Mijail Kabadjov. 2010. Agile corpus annotation in practice: An overview of manual and automatic annotation of CVs. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 29–37, Uppsala, Sweden.

Allan, Donner. 1999. Sample size requirements for interval estimation of the intraclass kappa statistic. *Communications in Statistics - Simulation and Computation*, 28(2):415–429.

Alt, Christoph, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online.

Amidei, Jacopo, Paul Piwek, and Alistair Willis. 2019. Agreement is overrated: A plea for correlation to assess human evaluation reliability. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 344–354, Tokyo, Japan.

Aroyo, Lora and Chris Welty. 2015. Truth Is a Lie: Crowd Truth and the Seven Myths of Human Annotation. *AI Magazine*, 36(1):15–24.

Artstein, Ron and Massimo Poesio. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.

Bakeman, Roger, Duncan McArthur, Vicenç Quera, and Byron F. Robinson. 1997. Detecting sequential patterns and determining their reliability with fallible observers. *Psychological Methods*, 2(4):357–370.

Banerjee, Mousumi, Michelle Capozzoli, Laura McSweeney, and Debajyoti Sinha. 1999. Beyond kappa: A review of interrater agreement measures. *Canadian Journal of Statistics*, 27(1):3–23.

Bareket, Dan and Reut Tsarfaty. 2021. Neural Modeling for Named Entities and Morphology (NEMO2). *Transactions of the Association for Computational Linguistics*, 9:909–928.

Bastan, Mohaddeseh, Mahnaz Koupaee, Youngseo Son, Richard Sicoli, and Niranjan Balasubramanian. 2020. Author's Sentiment Prediction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 604–615, Barcelona, Spain (Online).

Bayerl, Petra Saskia and Karsten Ingmar Paul. 2011. What Determines Inter-Coder Agreement in Manual Annotations? A Meta-Analytic Investigation. *Computational Linguistics*, 37(4):699–725.

Behrens, Heike. 2008. *Corpora in Language Acquisition Research: History, Methods, Perspectives*, volume 6 of *Trends in Language Acquisition Research*. John Benjamins Publishing Company, Amsterdam, The Netherlands.

Bender, Emily M. and Batya Friedman. 2018. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Bland, J. M. and D. G. Altman. 1986. Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet*, 1(8476):307–310.

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal.

Brandsen, Alex, Suzan Verberne, Milco Wansleeben, and Karsten Lambers. 2020. Creating a Dataset for Named Entity Recognition in the Archaeology Domain. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4573–4577, Marseille, France.

Button, Katherine S., John P. A. Ioannidis, Claire Mokrysz, Brian A. Nosek, Jonathan Flint, Emma S. J. Robinson, and Marcus R. Munafò. 2013. Power failure: Why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, 14(5):365–376.

Callison-Burch, Chris and Mark Dredze. 2010. Creating speech and language data with Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12, Los Angeles, California, USA.

Carletta, Jean. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254.

Cer, Daniel, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the*

11th International Workshop on Semantic Evaluation (SemEval-2017), pages 1–14, Vancouver, Canada.

Checco, Alessandro, Kevin Roitero, Eddy Maddalena, Stefano Mizzaro, and Gianluca Demartini. 2017. Let's Agree to Disagree: Fixing Agreement Measures for Crowdsourcing. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, pages 11–20, San Francisco, California, USA.

Chen, Zhiyu, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A Dataset of Numerical Reasoning over Financial Data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic.

Cohen, Jacob. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Da San Martino, Giovanni, Preslav Nakov, Jakub Piskorski, and Nicolas Stefanovitch. 2022. News Categorization, Framing and Persuasion Techniques: Annotation Guidelines.

Daniel, Florian, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2019. Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. *ACM Computing Surveys*, 51(1):1–40.

Dawid, A. P. and A. M. Skene. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied Statistics*, 28(1):20–28.

Demszky, Dorottya, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online.

Dickinson, Markus and W. Detmar Meurers. 2003. Detecting inconsistencies in treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, pages 1–12, Växjö, Sweden.

Donner, Allan and Michael Eliasziw. 1992. A goodness-of-fit approach to inference procedures for the kappa statistic: Confidence interval construction, significance-testing and sample size estimation. *Statistics in Medicine*, 11(11):1511–1519.

Dror, Rotem, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia.

Ebel, Robert L. 1951. Estimation of the reliability of ratings. *Psychometrika*, 16(4):407–424.

Edwards, Christopher, Heather Allen, and Crispen Chamunyonga. 2021. Correlation does not imply agreement: A cautionary tale for researchers and reviewers. *Sonography*, 8(4):185–190.

Efron, B. and R. Tibshirani. 1986. Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy. *Statistical Science*, 1(1):54–75.

Ferracane, Elisa, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2021. Did they answer? Subjective acts and intents in conversational discourse. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1626–1644, Online.

Fisher, Roland A. 1925. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh.

Fleiss, Joseph L. 1971. Measuring Nominal Scale Agreement among Many Raters. *Psychological Bulletin*, 76(5):378–382.

Fleiss, Joseph L., Bruce Levin, and Myunghee Cho Paik. 2003. *Statistical Methods for Rates and Proportions*, 1 edition. Wiley Series in Probability and Statistics. Wiley.

Gebru, Timnit, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Geva, Mor, Yoav Goldberg, and Jonathan Berant. 2019. Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China.

Ghosal, Deepanway, Siqi Shen, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2022. CICERO: A Dataset for Contextualized Commonsense Inference in Dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5010–5028, Dublin, Ireland.

Gildea, Daniel, Min-Yen Kan, Nitin Madnani, Christoph Teichmann, and Martín Villalba. 2018. The ACL Anthology: Current State and Future Directions. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 23–28, Melbourne, Australia.

Goodhart, C. A. E. 1984. *Problems of Monetary Management: The UK Experience*. Macmillan Education UK, London.

Govindarajan, Venkata Subrahmanyan, Benjamin Chen, Rebecca Warholic, Katrin Erk, and Junyi Jessy Li. 2020. Help! Need Advice on Identifying Advice. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5295–5306, Online.

Gururangan, Suchin, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online.

Hardt, Moritz and Benjamin Recht. 2022. *Patterns, Predictions, and Actions: Foundations of Machine Learning*. Princeton University Press, Princeton Oxford.

Harris, Christopher. 2011. Youre hired! An examination of crowdsourcing incentive models in human resource tasks. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 15–18, Hong Kong, China.

Haselbach, Boris, Kerstin Eckart, Wolfgang Seeker, Kurt Eberle, and Ulrich Heid. 2012. Approximating Theoretical Linguistics Classification in Real Data: The Case of German "nach" Particle Verbs. In *Proceedings of COLING 2012*, pages 1113–1128, Mumbai, India.

Hayes, Andrew F. and Klaus Krippendorff. 2007. Answering the Call for a Standard Reliability Measure for Coding Data. *Communication Methods and Measures*, 1(1):77–89.

Ho, Chien-Ju, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. 2015. Incentivizing High Quality Crowdwork. In *Proceedings of the 24th International Conference on World Wide Web*, pages 419–429, Florence Italy.

Holland, Sarah, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2018. The dataset nutrition label: A framework to drive higher data quality standards. *arXiv*, 1805(03677):1–21.

Horbach, Andrea, Yuning Ding, and Torsten Zesch. 2017. The influence of spelling errors on content scoring performance. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pages 45–53, Taipei, Taiwan.

Hovy, Dirk, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia, USA.

Hovy, Dirk, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 377–382, Baltimore, Maryland, USA.

Hovy, Eduard and Julia Lavid. 2010. Towards a ‘Science’ of Corpus Annotation: A New Methodological Challenge for Corpus Linguistics. *International Journal of Translation Studies*, 22:13–36.

Hsueh, Pei-Yun, Prem Melville, and Vikas Sindhwani. 2009. Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, Boulder, Colorado, USA.

Hutchinson, Ben, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. 2021. Towards Accountability for Machine Learning Datasets: Practices from Software Engineering and Infrastructure. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 560–575, Online.

Ide, Nancy and James Pustejovsky, editors. 2017. *Handbook of Linguistic Annotation*. Springer Netherlands, Dordrecht.

Jamison, Emily and Iryna Gurevych. 2015. Noise or additional information? Leveraging crowdsource annotation item agreement for natural language tasks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 291–297, Lisbon, Portugal.

Kim, Juyong, Jeremy C Weiss, and Pradeep Ravikumar. 2022. Context-Sensitive Spelling Correction of Clinical Text via Conditional Independence. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 234–247.

Kirk, Hannah, Bertie Vidgen, Paul Rottger, Tristan Thrush, and Scott Hale. 2022. Hatemoji: A Test Suite and Adversarially-Generated Dataset for Benchmarking and Detecting Emoji-Based Hate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1352–1368, Seattle, Washington, USA.

Klie, Jan-Christoph, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico, USA.

Klie, Jan-Christoph, Bonnie Webber, and Iryna Gurevych. 2023. Annotation Error Detection: Analyzing the Past and Present for a More Coherent Future. *Computational Linguistics*, 49(1):157–198.

Kottner, Jan, Laurent Audigé, Stig Brorson, Allan Donner, Byron J. Gajewski, Asbjørn Hróbjartsson, Chris Roberts, Mohamed Shoukri, and David L. Streiner. 2011. Guidelines for Reporting Reliability and Agreement Studies (GRRAS) were proposed. *Journal of Clinical Epidemiology*, 64(1):96–106.

Krippendorff, Klaus. 1980. *Content Analysis: An Introduction to Its Methodology*. SAGE, Los Angeles.

Krippendorff, Klaus. 1995. On the Reliability of Unitizing Continuous Data. *Sociological Methodology*, 25:47–76.

Krippendorff, Klaus. 2004. Reliability in Content Analysis.: Some Common Misconceptions and Recommendations. *Human Communication Research*, 30(3):411–433.

Krippendorff, Klaus. 2011. Agreement and Information in the Reliability of Coding. *Communication Methods and Measures*, 5(2):93–112.

Krippendorff, Klaus, Yann Mathet, Stéphane Bouvry, and Antoine Widlöcher. 2016. On the reliability of unitizing textual continua: Further developments. *Quality & Quantity*, 50(6):2347–2364.

Kummerfeld, Jonathan K. 2021. Quantifying and Avoiding Unfair Qualification Labour in Crowdsourcing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 343–349, Online.

Kummerfeld, Jonathan K., Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A Large-Scale Corpus for Conversation Disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy.

Květoň, Pavel and Karel Oliva. 2002. (Semi-)Automatic Detection of Errors in PoS-Tagged Corpora. In *COLING 2002: The 19th International Conference on Computational Linguistics*, pages 1–7, Taipei, Taiwan.

Landis, J. Richard and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174.

Lease, Matthew. 2011. On Quality Control and Machine Learning in Crowdsourcing. In *Proceedings of the 11th AAAI Conference on Human Computation*, AAAIWS'11-11, pages 97–102, San Francisco, California, USA.

Lindahl, Anna, Lars Borin, and Jacobo Rouces. 2019. Towards Assessing Argumentation Annotation - A First Step. In *Proceedings of the 6th Workshop on Argument Mining*, pages 177–186, Florence, Italy.

Lombard, Matthew, Jennifer Snyder-Duch, and Cheryl Campanella Bracken. 2002. Content Analysis in Mass Communication: Assessment and Reporting of Intercoder Reliability. *Human Communication Research*, 28(4):587–604.

McCoy, Tom, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics, pages 3428–3448, Florence, Italy.

Meyer, Christian M., Margot Mieskes, Christian Stab, and Iryna Gurevych. 2014. DKPro agreement: An open-source Java library for measuring inter-rater agreement. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 105–109, Dublin, Ireland.

Mihaylov, Todor, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium.

Mintz, Mike, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore.

Monarch, Robert. 2021. *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-Centered AI*. Manning Publications.

Mostafazadeh, Nasrin, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. GLUCOSE: GeneraLized and COntextualized Story Explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online.

Neuendorf, Kimberly A. 2016. *The Content Analysis Guidebook*. SAGE, Thousand Oaks, California, USA.

Northcutt, Curtis, Lu Jiang, and Isaac Chuang. 2021. Confident Learning: Estimating Uncertainty in Dataset Labels. *Journal of Artificial Intelligence Research*, 70:1373–1411.

Northcutt, Curtis G., Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. In *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, pages 1–13, Online.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 1–15.

Parmar, Mihir, Swaroop Mishra, Mor Geva, and Chitta Baral. 2023. Don't blame the annotator: Bias already starts in the annotation instructions. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1779–1789, Dubrovnik, Croatia.

Parrish, Alicia, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does Putting a Linguist in the Loop Improve NLU Data Collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic.

Passonneau, Rebecca J. and Bob Carpenter. 2014. The Benefits of a Model of Annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326.

Paun, Silviu, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian Models of Annotation. *Transactions of the Association for Computational Linguistics*, 6(0):571–585.

Peters, Matthew E., Sebastian Ruder, and Noah A. Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy.

Popping, R. 1988. On Agreement Indices for Nominal Data. In *Sociometric Research*. Palgrave Macmillan UK, London, pages 90–105.

Powers, David. 2011. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

Prasad, Rashmi, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 2961–2968, Marrakech, Morocco.

Pushkarna, Mahima, Andrew Zaldivar, and Oddur Kjartansson. 2022. Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1776–1826, Seoul, Republic of Korea.

Pustejovsky, J. and Amber Stubbs. 2013. *Natural Language Annotation for Machine Learning*. O'Reilly Media, Sebastopol, California, USA.

Qian, Kun, Ahmad Beirami, Zhouhan Lin, Ankita De, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. 2021. Annotation inconsistency and entity bias in MultiWOZ. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 326–337, Online.

Ranganathan, Priya, Cs Pramesh, and Rakesh Aggarwal. 2017. Common pitfalls in statistical analysis: Measures of agreement. *Perspectives in Clinical Research*, 8(4):187–191.

Reddy, Siva, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Reiss, Frederick, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger. 2020. Identifying Incorrect Labels in the CoNLL-2003 Corpus. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 215–226, Online.

Roh, Yuji, Geon Heo, and Steven Euijong Whang. 2021. A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1328–1347.

Sabou, Marta, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 859–866, Reykjavik, Iceland.

Sambasivan, Nithya, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Kumar Paritosh, and Lora Mois Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *SIGCHI*, pages 1–21.

Schreibman, Susan, Ray Siemens, and John Unsworth, editors. 2004. *A Companion to Digital Humanities*. Blackwell Publishing Ltd, Malden, Massachusetts, USA.

Scott, William A. 1955. Reliability of content analysis: The case of nominal scale coding. *The Public Opinion Quarterly*, 19(3):321–325.

Sheshadri, Aashish and Matthew Lease. 2013. SQUARE: A Benchmark for Research on Computing Crowd Consensus. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 1:156–164.

Shmueli, Boaz, Jan Fell, Soumya Ray, and Lun-Wei Ku. 2021. Beyond Fair Pay: Ethical Implications of NLP Crowdsourcing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3758–3769, Online.

Shoukri, M M, M H Asyali, and A Donner. 2004. Sample size requirements for the design of reliability study: Review and new results. *Statistical Methods in Medical Research*, 13(4):251–271.

Shrout, Patrick E. and Joseph L. Fleiss. 1979. Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420–428.

Sim, Julius and Chris C Wright. 2005. The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, 85(3):257–268.

Simpson, Edwin D. and Iryna Gurevych. 2019. A Bayesian Approach for Sequence Tagging with Crowds. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1093–1104, Hong Kong, China.

Singh, Shikhar, Nuan Wen, Yu Hou, Pegah Alipoormolabashi, Te-lin Wu, Xuezhe Ma, and Nanyun Peng. 2021. COM2SENSE: A Commonsense Reasoning Benchmark with Complementary Sentences. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 883–898, Online.

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, USA.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA.

Stab, Christian and Iryna Gurevych. 2014. Identifying Argumentative Discourse Structures in Persuasive Essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar.

Stoica, George, Emmanouil Antonios Platanios, and Barnabas Poczos. 2021. Re-TACRED: Addressing Shortcomings of the TACRED Dataset. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence 2021*, pages 13843–13850, Online.

Sun, Chen, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 843–852, Venice, Italy.

Suster, Simon, Stephan Tulkens, and Walter Daelemans. 2017. A Short Review of Ethical Challenges in Clinical Natural Language Processing. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 80–87, Valencia, Spain.

Tjong Kim Sang, Erik F. and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, Edmonton, Alberta, Canada.

Uma, Alexandra N., Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from Disagreement: A Survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Vădineanu, Șerban, Daniel Pelt, Oleh Dzyubachyk, and Joost Batenburg. 2022. An Analysis of the Impact of Annotation Errors on the Accuracy of Deep Learning for Cell Segmentation. In *Proceedings of Machine Learning Research*, pages 1251–1267, Honolulu, Hawaii, USA.

van Stralen, K.J., F.W. Dekker, C. Zoccali, and K.J. Jager. 2012. Measuring Agreement, More Complicated Than It Seems. *Nephron Clinical Practice*, 120(3):162–167.

Vasudevan, Vijay, Benjamin Caine, Raphael Gontijo-Lopes, Sara Fridovich-Keil, and Rebecca Roelofs. 2022. When does dough become a bagel? Analyzing the remaining mistakes on ImageNet. In *Proceedings of the 36th Conference on Neural Information Processing Systems*, pages 1–15, New Orleans, Louisiana, USA.

Wang, Zihan, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training Named Entity Tagger from Imperfect Annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5153–5162, Hong Kong, China.

Wei, Jason, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, pages 1–46, Online.

Wynne, Martin, editor. 2005. *Developing Linguistic Corpora: A Guide to Good Practice*. David Brown Book Company, Oakville, Connecticut.

Yao, Yuan, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy.

Zapf, Antonia, Stefanie Castell, Lars Morawietz, and André Karch. 2016. Measuring inter-rater reliability for nominal data – which coefficients and confidence intervals are appropriate? *BMC Medical Research Methodology*, 16(1):93–103.

Zeng, Zhiqiang, Hua Shi, Yun Wu, and Zhiling Hong. 2015. Survey of Natural Language Processing Techniques in Bioinformatics. *Computational and Mathematical Methods in Medicine*, 2015:1–10.

Zhang, Yuhao, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark.

Zhao, Xinshu, Jun S. Liu, and Ke Deng. 2013. Assumptions behind Intercoder Reliability Indices. *Annals of the International Communication Association*, 36(1):419–480.

1. Data Collection

We use a snapshot of the *Papers With Code*[13] data from the 26th of November, 2022. From that, we select the *text* datasets and match them against the ACL Anthology[14] with the commit `3e0966ac`. While the ACL Anthology also contains backlinks to *Papers With Code*, they were still very few (≈100 datasets marked at the time of writing). Hence, we opted to match them by title manually.

File Name	**md5**
datasets.json.gz	57193271ad26d827da3666e54e3c59dc
papers-with-abstracts.json.gz	4531a8b4bfbe449d2a9b87cc6a4869b5
links-between-papers-and-code.json.gz	424f1b2530184d3336cc497db2f965b2

Table 2: File names and checksums for the *Papers With Code* data.

2. Guidelines

This annotation project aims to analyze how quality management is conducted in the wild. In the following, we describe the different aspects we annotate.

2.1 Manual Annotation

We are mainly interested in analyzing works that use human annotators. Therefore, we annotate whether a dataset involves humans as either annotators or validators.

2.2 Task Type

We see two broad categories of tasks that require different quality management methods.

Annotation This encompasses annotation projects where annotators provide labels, for instance, text classification, named entity recognition, annotating entailment for natural language inference, or selecting the right question from a given set for question answering.

Text Production This encompasses annotation projects where annotators produce text. This can be, for instance, when writing surface forms that are later annotated. Other tasks include summarization, question answering, dialogues, and natural language generation.

A dataset publication can use both task types, e.g., when creating questions and selecting the correct answer from a predefined pool or for natural language inference, where the clauses are first written and then labeled for their entailment.

[13] https://github.com/paperswithcode/paperswithcode-data
[14] https://github.com/acl-org/acl-anthology

2.3 Annotators

Expert We consider an annotator an expert if they annotate due to their domain knowledge or prior experience with the task.

Contractor We consider an annotator a contractor if they are hired individually, for instance, student helpers or freelancers via platforms like Upwork or Prolific. The project managers usually know them by name and can directly interact with them. They can be managed on a more fine-grained level compared to crowdworkers.

Crowd Crowdworkers are annotators who participate via platforms like Crowdflower or Amazon Mechanical Turk. Annotation is usually done in the form of microtasks. The annotators are relatively anonymous. There are often tens or hundreds of different annotators, each annotating only a small part of the overall data.

Volunteer Volunteers are annotators who help for free and are not required to do so. This, for instance, excludes students who annotate as part of their coursework.

2.4 Quality Management Methods

2.4.1 Annotation Process.

Iterative Annotation Process Mentions that an iterative feedback loop is used as the annotation process.

Pilot Study It is mentioned that one or more pilot studies have been performed.

Data Filtering Data is filtered before annotation via automatic or manual checks.

Validation Mentions an explicit validation step. See Appendix 2.9.

Indirect Annotation The annotation process has several steps, where the later ones indirectly validate earlier ones.

2.4.2 Annotator Management.

Annotator Training Training of annotators is mentioned.

Qualification Filter It is mentioned that annotators are filtered out by criteria like native language, geographic location, previous acceptance rates, number of previously completed tasks, etc.

Qualification Test It is mentioned that annotators had to take a qualification test before being allowed to participate in the annotation process itself.

Monetary Incentive Give annotators additional payments if their quality is exceptional.

2.4.3 Quality Estimation.

Agreement Uses at least one agreement measure. This must have been used for the annotation process or validation, not the pilot study. See Appendix 2.8.

Error Rate Computes the error rate for the final, adjudicated corpus. See Appendix 2.11.

Control Questions Injects control questions for which the answer is known to estimate annotator and task performance.

2.4.4 Rectifying Measures.

Guideline Refinement Mentions that guidelines and annotation schemes are refined.

Correction Mentions that instances are improved and corrected.

Annotator Debriefing Annotators give feedback to improve the annotation process.

Give Annotators Feedback Annotators are given feedback to improve their annotation quality.

Agreement Filter Instances are filtered out if agreement is too low.
Annotator Deboarding Annotators are removed from the labor pool if their quality is deemed insufficient.
Manual Filter Instances are filtered out manually if agreement is too low.
Time Filter Instances are filtered out if annotators annotate improbably quickly.
Automatic Checks Automatic checks are applied, for instance, spell checking or handcrafted rules.

2.5 Adjudication

Adjudication describes the process of merging multiple annotations per instance into a single one.

Majority Voting The label assigned by at least half of the annotators is chosen. We also count adjudication as majority voting if all annotators must agree in the analysis, but label it as *TotalAgreement*.
Manual Tie Breaking A human annotator manually inspects instances without a majority and curates them. This adjudication method should be annotated together with *Majority Voting*.
Dawid-Skene This is an aggregation model that uses probabilistic graphical models to describe the expertise of the annotators.
MACE This is an aggregation model that uses probabilistic graphical models to describe the expertise and likeliness of being a spammer of the annotators.
Manual Curation A human annotator manually inspects and curates instances.
N/A If there is only one annotation per instance or the task type is text production.
? No mention of adjudication is found in the publication, but adjudication must have happened, e.g., because the publication mentioned more than one annotation per label.

If the task type is only text production, just enter `N/A` or leave the field empty; if annotation + text production, enter `?` or the mentioned one. If you encounter new or different adjudication procedures, then please add them to the tagset.

2.6 Guidelines available

For reproducibility and to judge the quality of the annotation process, it is crucial that the guidelines are available. We consider guidelines available either in the publication, appendix, or supplementary material,

- a detailed annotation tagset/task/scheme description
- a screenshot of the annotation interface with a task description for the annotators
- or the guidelines itself

are given. We only check the external supplementary material if it is referred to in the publication. In case the supplementary material is mentioned but not findable in the ACL anthology, we consider guidelines not to be available.

2.7 Overall Judgement

We assign an overall rating to each publication having human annotators based on their quality management conducted and reported. The grades are in three categories:

Excellent Does most of the following: uses the iterative annotation process, trains annotators, computes agreement and error rate, performs extensive validation, and does continuous human inspection.

Sufficient Uses some of the recommended techniques, but not as extensive as excellent. Has at least some validation and manual inspection.

Subpar No agreement, validation, manual inspection error rate, or other quality management performed and reported. The data quality, at most, relies on aggregation of multiple annotations.

2.8 Agreement

For each agreement value that is reported, create a new agreement annotation. Agreement used in pilot studies should not be entered; we are only interested in values computed for the final dataset.

2.8.1 Measure Name. Enter the name of the measure. We are at least interested in the following:

- Percent Agreement
- Cohen's κ
- Fleiss's κ
- Krippendorf's α
- Krippendorf's α unitized
- Pearson's r
- Spearman's ρ
- Kendall's τ
- Intraclass correlation coefficient
- Precision
- Recall
- F1

Enter ? if it is unclear what the agreement measure is. If you encounter new, different agreement measures, then please add them to the tagset.

2.8.2 Value. Enter the agreement value that is reported. If no value is reported, but the use of agreement is, fill in as much as possible and enter -1.

2.8.3 Inspection Size. Enter the size of the subset that is used to compute agreement and the overall dataset size. If the agreement is computed on the whole dataset, enter 0 for both sample and total sizes.

2.8.4 Interpretation. We annotate the interpretation that is given together with the agreement value. We are at least interested in the following works that give ranges for agreement measures and their interpretation.

Landis *The Measurement of Observer Agreement for Categorical Data* by J. Richard Landis and Gary G. Koch, 1977.

Kripppendorf *Validity in Content Analysis* by Klaus Krippendorff, 1980.

If you encounter new, different works referenced that give interpretations, then please add them to the tagset. We are also interested in

Custom Interpretation States that their agreement shows a certain level of quality, for instance, *sufficient*, *high*, *good* without referencing a work from the literature.

Compares To Previous Mentions a dataset that is similar to the one presented and compares its agreement to its predecessor.

2.9 Validation

We are interested in whether validation is done and who did the validation, if any.

2.10 Validators

The labels for who is validating are the same as for annotators.

2.10.1 Inspection Size. Enter the size of the subset that is validated, as well as the overall dataset size. If the complete dataset is validated, enter 0 for both sample and total sizes.

2.11 Error Rate

The error rate is the number of incorrect instances divided by the total number of instances in the dataset. We annotate it if it is computed on the adjudicated dataset. It is usually computed on a subset of instances.

2.11.1 Value. Enter the error rate value that is reported. If no value is reported, but the error rate is used, fill in as much as possible and enter -1.

2.11.2 Inspection Size. Enter the size of the subset that is used to compute the error rate as well as the overall dataset size. If the error rate is computed on the whole dataset, enter 0 for both sample and total sizes.

3. Correlation

In the following, we give an example where correlation between ratings is high but agreement is low. We assume two annotators rating four items on a scale in $[1, 5]$:

Item		a	b	c	d
Judge	A	1	2	3	4
	B	3	4	5	5

The resulting correlation scores and are:

Pearson's ϱ	Spearman's ϱ	Kendall τ	ICC1	ICC2	ICC3
0.944	0.949	0.913	0.204	0.418	0.903

It can be seen that standard correlation measures show very high correlation, while Intraclass Correlation scores are comparatively low.

Chapter 8

Annotation Error Detection: Analyzing the Past and Present for a More Coherent Future

Annotation Error Detection: Analyzing the Past and Present for a More Coherent Future

Jan-Christoph Klie*
Ubiquitous Knowledge Processing Lab
Department of Computer Science
Technical University of Darmstadt
www.ukp.tu-darmstadt.de

Bonnie Webber
School of Informatics,
University of Edinburgh

Iryna Gurevych
UKP Lab / TU Darmstadt

Annotated data is an essential ingredient in natural language processing for training and evaluating machine learning models. It is therefore very desirable for the annotations to be of high quality. Recent work, however, has shown that several popular datasets contain a surprising number of annotation errors or inconsistencies. To alleviate this issue, many methods for annotation error detection have been devised over the years. While researchers show that their approaches work well on their newly introduced datasets, they rarely compare their methods to previous work or on the same datasets. This raises strong concerns on methods' general performance and makes it difficult to assess their strengths and weaknesses. We therefore reimplement 18 methods for detecting potential annotation errors and evaluate them on 9 English datasets for text classification as well as token and span labeling. In addition, we define a uniform evaluation setup including a new formalization of the annotation error detection task, evaluation protocol, and general best practices. To facilitate future research and reproducibility, we release our datasets and implementations in an easy-to-use and open source software package.[1]

1. Introduction

Annotated corpora are an essential component in many scientific disciplines, including natural language processing (NLP) (Gururangan et al. 2020; Peters, Ruder, and Smith

* Corresponding author.

[1] `https://github.com/UKPLab/nessie`.

Action Editor: Nianwen Xue. Submission received: 7 June 2022; revised version received: 29 August 2022; accepted for publication: 23 September 2022.

https://doi.org/10.1162/coli_a_00464

2019), linguistics (Haselbach et al. 2012), language acquisition research (Behrens 2008), and the digital humanities (Schreibman, Siemens, and Unsworth 2004). Corpora are used to train and evaluate machine learning models, to deduce new knowledge, and to suggest appropriate revisions to existing theories. Especially in machine learning, high-quality datasets play a crucial role in advancing the field (Sun et al. 2017). It is often taken for granted that gold standard corpora do not contain errors—but alas, this is not always the case. Datasets are usually annotated by humans who can and do make mistakes (Northcutt, Athalye, and Mueller 2021). Annotation errors can even be found in corpora used for shared tasks such as CONLL-2003 (Tjong Kim Sang and De Meulder 2003). For instance, *Durban* is annotated there as PER (person) and *S.AFRICA* as MISC (miscellaneous), but both should be annotated as LOC (location).

Gold standard annotation is also subject to inconsistency, where words or phrases that are intended to refer to the same type of thing (and so should be labeled in the same way) are nevertheless assigned different labels (see, e.g., Hollenstein, Schneider, and Webber 2016). For example, in CONLL-2003, when *Fiorentina* was used to refer to the local football club, it was annotated as ORG, but when *Japan* was used to refer to the Japanese national football team, it was inconsistently annotated as LOC. One reason for annotation inconsistencies is that tokens can be ambiguous, either because they have multiple senses (e.g., the word *club* can refer to an organization or to a weapon), or because metonymy allows something to be referred to by one of its parts or attributes (e.g., the Scottish curling team being referred to as *Scotland*, as in *Scotland beat Canada in the final match*). We further define errors as well as inconsistencies and also discuss ambiguity in detail in § 3.1.

Such annotation errors or inconsistencies can negatively impact a model's performance or even lead to erroneous conclusions (Manning 2011; Northcutt, Athalye, and Mueller 2021; Larson et al. 2020; Zhang et al. 2021). A deployed model that learned errors during training can potentially cause harm, especially in critical applications like medical or legal settings. High-quality labels are needed to evaluate machine learning methods even if they themselves are robust to label noise (e.g., Song et al. 2020). Corpus linguistics relies on correctly annotated data to develop and confirm new theories. Learner corpora containing errors might be detrimental to the language learning experience and teach wrong lessons. Hence, it is imperative for datasets to have high-quality labels.

Cleaning the labels by hand, however, is expensive and time consuming. Therefore, many automatic methods for annotation error detection (AED) have been devised over the years. These methods enable dataset creators and machine learning practitioners to narrow down the instances that need manual inspection and—if necessary—correction. This reduces the overall work needed to find and fix annotation errors (see, e.g., Reiss et al. 2020). As an example, AED has been used to discover that widely used benchmark datasets contain errors and inconsistencies (Northcutt, Athalye, and Mueller 2021). Around 2% of the samples (sometimes even more than 5%) have been found incorrectly annotated in datasets like Penn Treebank (Dickinson and Meurers 2003a), sentiment analysis datasets like SST, Amazon Reviews, or IMDb (Barnes, Øvrelid, and Velldal 2019; Northcutt, Athalye, and Mueller 2021), CoNLL-2003 (Wang et al. 2019; Reiss et al. 2020), or relation extraction in TACRED (Alt, Gabryszak, and Hennig 2020; Stoica, Platanios, and Poczos 2021). AED has likewise been used to find ambiguous instances, for example, for part-of-speech (POS) annotation (Dickinson and Meurers 2003a). Additionally, it has been shown that errors in automatically annotated (silver) corpora can also be found and fixed with the help of AED (Rehbein 2014; Ménard and Mougeot 2019).

While AED methods have been applied successfully in the past (e.g., Reiss et al. 2020), there are several issues that hinder their widespread use. New approaches for AED are often only evaluated on newly introduced datasets that are proprietary or not otherwise available (e.g., Dligach and Palmer 2011; Amiri, Miller, and Savova 2018; Larson et al. 2019). Also, they rarely compare newly introduced methods to previous work or baselines. These issues make comparisons of AED methods very difficult. In addition to that, there is neither agreement on how to evaluate AED methods, nor which metrics to use during their development and application. As a result, it is often not clear how well AED works in practice, especially which AED methods should be applied to which kind of data and underlying tasks. To alleviate these issues, we define a unified evaluation setup for AED, conduct a large-scale analysis of 18 AED methods, and apply them to 9 datasets for text classification, token labeling, and span labeling. This work focuses on errors and inconsistencies related to instance labels. We leave issues such as boundary errors, sentence splitting, or tokenization for future work. The methods presented in this article are particularly suited to the NLP community, but many of them can also be adapted to other tasks (e.g., relation classification) and domains (like computer vision). The research questions we answer are:

RQ1 Which methods work well across tasks and datasets?

RQ2 Does model calibration help to improve AED performance?

RQ3 To what extent are model and AED performance correlated?

RQ4 What (performance) impact does using cross-validation have?

The research reported in this article addresses the aforementioned issues by providing the following contributions:

Evaluation Methodology To unify its findings and establish comparability, we first define the task of AED and a standardized evaluation setup, including an improvement for evaluating span labeling in this context (§ 3.1).

Easy to Use Reference Implementations We survey past work from the last 25 years and implement the 18 most common and generally applicable AED methods (§ 3.2). We publish our implementation in a Python package called NESSIE that is easy to use, thoroughly tested, and extensible to new methods and tasks. We provide abstractions for models, tasks, as well as helpers for cross validation to reduce the boilerplate code needed to a minimum. In addition, we provide extensive documentation and code examples. Our package makes it therefore significantly easier to get started with AED for researchers and practitioners alike.

Benchmarking Datasets We identify, vet, and generate datasets for benchmarking AED approaches, which results in 9 datasets for text classification, token labeling, and span labeling (§ 4). We also publish the collected datasets to facilitate easy comparison and reproducibility.

Evaluation and Analysis Using our implementation, we investigate several fundamental research questions regarding AED (§ 5). We specifically focus on how to achieve the best AED performance for each task and dataset, taking model calibration, usage of cross-validation, as well as model selection into account. Based on our results, we provide recipes and give recommendations on how to best use AED in practice (§ 6).

2. Related Work

This section provides a brief overview of annotation error detection and its related tasks.

Annotation Error Detection. In most works, AED is used as a means to improve the quality of an annotated corpus. As such, the method used is treated as secondary and possible methods are not compared. The work of Amiri, Miller, and Savova (2018) and Larson et al. (2020) are the few instances that implement different methods and baselines, but only use newly introduced datasets. In other cases, AED is just discussed as a minor contribution and not thoroughly evaluated (e.g., Swayamdipta et al. 2020, Rodriguez et al. 2021).

Therefore, to the best of our knowledge, no large-scale evaluation of AED methods exists. Closest to the current study is the work of Dickinson (2015), a survey about the history of annotation error detection. However, that survey does not reimplement, compare, or evaluate existing methods quantitatively. Its focus is also limited to part-of-speech and dependency annotations. Our work fills the aforementioned gaps by reimplementing 18 methods for AED, evaluating the methods against 9 datasets, and investigating the setups in which they perform best.

Annotation Error Correction. After potential errors have been detected, the next step is to have them corrected to obtain gold labels. This is usually done by human annotators who carefully examine those instances that have been detected. Some AED methods can also both detect and correct labels. Only a few groups have studied correction so far (e.g., Květoň and Oliva 2002; Loftsson 2009; Dickinson 2006; Angle, Mishra, and Sharma 2018; Qian et al. 2021). In this study, we focus on detection and leave an in-depth treatment of annotation error correction for future work.

Error Type Classification. Even if errors are not corrected automatically, it may still be worth identifying the type of each error. For instance, Larson et al. (2020) investigate the different errors for slot filling (e.g., incorrect span boundaries, incorrect labels, or omissions). Alt, Gabryszak, and Hennig (2020) investigate error types for relation classification. Yaghoub-Zadeh-Fard et al. (2019) collect tools and methods to find quality errors in paraphrases used to train conversational agents. Barnes, Øvrelid, and Velldal (2019) analyze the types of errors found in annotating for sentiment analysis. While error type classification has not been explicitly addressed in the current study, such classification requires good AED, so the results of the current study can contribute to automate error type classification in the future.

Training with Label Noise. Related to AED is the task of training with noise: Given a dataset that potentially contains label errors, train a model so that the performance impact due to noisy labels is as low as possible (Song et al. 2020). The goal in this setting is not to clean a dataset (labels are left as is), but to obtain a well-performing machine learning model. An example application is learning directly from crowdsourced data without adjudicating it (Rodrigues and Pereira 2018). Training with label noise and AED have in common that they both enable models to be trained when only noisy data is available. Evaluating these models still requires clean data, which AED can help to produce.

3. Annotation Error Detection

In this section we introduce AED and formalize the concept, then categorize state-of-the-art approaches according to our formalization.

3.1 Task Definition

Given an adjudicated dataset with one label per annotated instance, the goal of AED is to find those instances that are likely labeled incorrectly or inconsistently. These candidate instances then can be given to human annotators for manual inspection or used in annotation error correction methods. The definition of *instance* depends on the task and defines the granularity on which errors or inconsistencies are detected. In this article, we consider AED in text classification (where instances are sentences), in token labeling (instances are tokens; e.g., POS tagging), and in span labeling (instances are spans; e.g., named entity recognition [NER]). AED can and has been applied to many domains and tasks, for instance, sentiment analysis (Barnes, Øvrelid, and Velldal 2019; Northcutt, Athalye, and Mueller 2021), relation extraction (Alt, Gabryszak, and Hennig 2020), POS tagging (Dickinson and Meurers 2003a; Loftsson 2009), image classification (Northcutt, Athalye, and Mueller 2021), NER (Wang et al. 2019; Reiss et al. 2020), slot filling (Larson et al. 2020), or speech classification (Northcutt, Athalye, and Mueller 2021).

We consider a label to be **incorrect** if there is a unique, true label that should be assigned but it differs from the label that has been assigned. For example, there is a named entity span Durban in CONLL-2003 which has been labeled PER, whereas in context, it refers to a city in South Africa, so the label should be LOC.

Instances can also be **ambiguous**, that is, there are at least two different labels that are valid given the context. For instance, in the sentence *They were visiting relatives*, *visiting* can either be a verb or an adjective. Ambiguous instances themselves are often more difficult for machine learning models to learn from and predict. Choosing one label over another is neither inherently correct nor incorrect. But ambiguous instances can be annotated *inconsistently*. We consider a label **inconsistent** if there is more than one potential label for an instance, but the choice of resolution was different for similar instances. For example, in the sentence *Stefan Edberg produced some of his vintage best on Tuesday to extend his grand run at the Grand Slams by toppling Wimbledon champion Richard Krajicek*, the entity *Wimbledon* was annotated as LOC. But in the headline of this article, *Edberg extends Grand Slam run, topples Wimbledon champ*, the entity *Wimbledon* was annotated as MISC. We discuss the impact of ambiguity on AED further in § 3.1. An instance that is neither incorrect nor inconsistent is **correct**. If not explicitly stated otherwise, then we refer to both incorrect and inconsistent as incorrect or erroneous.

AED is typically used after a new dataset has been annotated and adjudicated. It is assumed that no already cleaned data and no other data having the same annotation scheme is available.

Flaggers vs. Scorers. We divide automatic methods for AED into two categories, which we dub *flaggers* and *scorers*. Flagging means that methods cast a binary judgment whether the label for an instance is correct or incorrect. Scoring methods give an estimate on how likely it is that an annotation is incorrect. These correspond to classification and ranking.

While flaggers are explicit as to whether they consider an annotation to be incorrect, they do not indicate the likelihood of that decision. On the other hand, while scorers provide a likelihood, they require a threshold value to decide when an annotation is considered an error—for example, those instances with a score above 80%. Those would then be given to human evaluation. Scorers can also be used in settings similar to active learning for error correction (Vlachos 2006).

This distinction between flaggers and scorers regarding AED has not been made in previous work, as typically approaches of one type or the other have been proposed per paper. But it is key to understanding why different metrics need to be used when evaluating flaggers compared to scorers, similarly to unranked and ranked evaluation from information retrieval (see § 5).

Ambiguity. In certain NLP tasks, there exists more than one valid label per instance (Kehler et al. 2007; Plank, Hovy, and Søgaard 2014b; Aroyo and Welty 2015; Pavlick and Kwiatkowski 2019; Basile et al. 2021). While this might reduce the usefulness of AED at first glance, gold labels are not required by AED, as it is about uncovering problems independent of their cause and not assigning a gold label. Instances detected this way are then marked for further processing. They can be, for instance, inspected for whether they are incorrectly or inconsistently annotated. Ambiguous or difficult instances especially deserve additional scrutiny when creating a corpus; finding them is therefore very useful. Once found, several alternatives are possible: (1) Ambiguous cases can be corrected (e.g., Alt, Gabryszak, and Hennig 2020; Reiss et al. 2020); (2) they can be removed (e.g., Jamison and Gurevych 2015); (3) their annotation guidelines can be adjusted to reduce disagreement (e.g., Pustejovsky and Stubbs 2013); (4) the task can eventually be redefined to use soft labels (Fornaciari et al. 2021) or used to learn from disagreement (Paun et al. 2018). Finding such instances is hence very desirable and can be achieved by AED. But similarly to past work on AED, we focus on detecting errors and inconsistencies as a first step and leave evaluating ambiguity detection performance for future work.

3.2 Survey of Existing AED Methods

Over the past three decades, several methods have been developed for AED. Here, we group them by how they detect annotation errors and briefly describe each of them. In this article, we focus on AED for natural language processing, but (as noted earlier in § 1), many of the presented methods can be and have been adjusted to different tasks and modalities. An overview of the different methods is also given in Table 1.

3.2.1 Variation-based. Methods based on the variation principle leverage the observation that similar surface forms are often annotated with only one or at most a few distinct labels. If an instance is annotated with a different, rarer label, then it is possibly an annotation error or an inconsistency. Variation-based methods are relatively easy to implement and can be used in settings in which it is difficult to train a machine learning model, such as low-resource scenarios or tasks that are difficult to train models for, for example, detecting lexical semantic units (Hollenstein, Schneider, and Webber 2016). The main disadvantage of variation-based methods is that they need similar surface forms to perform well, which is not the case in settings like text classification or datasets with diverse instances.

Variation n-grams. The most frequently used method of this kind is variation *n*-grams, which has been initially developed for POS tagging (Dickinson and Meurers 2003a) and later extended to discontinuous constituents (Dickinson and Meurers 2005),

Table 1
Annotation error detection methods evaluated in this work. In most scorer methods, scorer output and erroneous labels are positively correlated. Scorers marked with * show negative correlation.

Abbr.	Method Name	Tasks			Proposed by
		Text	Token	Span	
Flagger methods					
CL	Confident Learning	✓	✓	✓	Northcutt et al. (2021)
CS	Curriculum Spotter	✓	·	·	Amiri et al. (2018)
DE	Diverse Ensemble	✓	✓	✓	Loftsson (2009)
IRT	Item Response Theory	✓	✓	✓	Rodriguez et al. (2021)
LA	Label Aggregation	✓	✓	✓	Amiri et al. (2018)
LS	Leitner Spotter	✓	·	·	Amiri et al. (2018)
PE	Projection Ensemble	✓	✓	✓	Reiss et al. (2020)
RE	Retag	✓	✓	✓	van Halteren (2000)
VN	Variation N-Grams	·	✓	✓	Dickinson and Meurers (2003a)
Scorer methods					
BC	Borda Count	✓	✓	✓	Larson et al. (2020)
CU	Classification Uncertainty	✓	✓	✓	Hendrycks and Gimpel (2017)
DM*	Data Map Confidence	✓	·	·	Swayamdipta et al. (2020)
DU	Dropout Uncertainty	✓	✓	✓	Amiri et al. (2018)
KNN	k-Nearest Neighbor Entropy	✓	✓	✓	Grivas et al. (2020)
LE	Label Entropy	·	✓	✓	Hollenstein et al. (2016)
MD	Mean Distance	✓	✓	✓	Larson et al. (2019)
PM*	Prediction Margin	✓	✓	✓	Dligach and Palmer (2011)
WD	Weighted Discrepancy	·	✓	✓	Hollenstein et al. (2016)

predicate-argument structures (Dickinson and Lee 2008), dependency parsing (Boyd, Dickinson, and Meurers 2008), or slot filling (Larson et al. 2020). For each instance, n-gram contexts of different sizes are collected and compared to each other. It is considered incorrect if the label for an instance disagrees with labels from other instances with the same n-gram context.

Label Entropy and Weighted Discrepancy. Hollenstein, Schneider, and Webber (2016) derive metrics from the surface form and label counts that are then used as scorers. These are the entropy over the label count distribution per surface form or the weighted difference between most and least frequent labels. They apply their methods to find possible annotation errors in datasets for multi-word expressions and super-sense tagging, which are then reviewed manually for tokens that are actual errors.

3.2.2 Model-based. Probabilistic classifiers trained on the to-be-corrected dataset can be used to find annotation errors. Models in this context are usually trained via cross-validation (CV) and the respective holdout set is used to detect errors. After all folds have been used as holdout, the complete dataset is analyzed. Because some methods described below directly use model probabilities, it is of interest whether these are accurately describing the belief of the model. This is not always true, as models often are overconfident (Guo et al. 2017). Therefore, we will evaluate whether **calibration**, that is, tuning probabilities so that they are closer to the observed accuracy, can improve performance (see § 5.2). Several ways have been devised for model-based AED, which are described below. Note that most model-based methods are agnostic to the task itself and rely only on model predictions and confidences. This is why they can easily be used with different tasks and modalities.

Re-tagging. A simple way to use a trained model for AED is to use model predictions directly; when the predicted labels are different from the manually assigned ones, instances are flagged as annotation errors (van Halteren 2000). Larson et al. (2020) apply this using a conditional random field (CRF) tagger to find errors in crowdsourced slot-filling annotations. Similarly, Amiri, Miller, and Savova (2018) use *Retag* for text classification. Yaghoub-Zadeh-Fard et al. (2019) train machine learning models to classify whether paraphrases contain errors and if they do, what kind of error it is. To reduce the need of annotating instances twice for higher quality, Dligach and Palmer (2011) train a model on the labels given by an initial annotator. If the model disagrees with the instance's labeling, then it is flagged for re-annotation. For cleaning dependency annotations in a Hindi treebank, Ambati et al. (2011) train a logistic regression classifier. If the model's label does not agree with the original annotation and the model confidence is above a predefined threshold, then the annotation is considered to be incorrect. *CrossWeigh* (Wang et al. 2019) is similar to *Retag* with repeated CV. During CV, *entity disjoint filtering* is used to force more model errors: Instances are flagged as erroneous if the probability of their having the correct label falls below the respective threshold. As it is computationally much more expensive than *Retag* while being very similar, we did not include it in our comparison.

Classification Uncertainty. Probabilistic classification models assign probabilities that are typically higher for instances that are correctly labeled compared with erroneous ones (Hendrycks and Gimpel 2017). Therefore, the class probabilities of the noisy labels can be used to score these for being an annotation error. Using model uncertainty is basically identical to using the network loss (as, e.g., used by Amiri, Miller, and Savova 2018) because the cross-entropy function used to compute the loss is monotonic. The probability formulation, however, allows us to use calibration more easily later (see § 5.2), which is why we adapt the former instead of using the loss.

Prediction Margin. Inspired by active learning, *Predictive Margin* uses the probabilities of the two highest scoring labels for an instance. The resulting score is simply their difference (Dligach and Palmer 2011). The intuition behind this is that samples with a smaller margin are more likely to be an annotation error, since the smaller the decision margin is the more unsure the model was.

Confident Learning. This method estimates the joint distribution of noisy and true labels (Northcutt, Jiang, and Chuang 2021). A threshold is then learned (the average self-confidence) and instances whose computed probability of having the correct label is below the respective threshold are flagged as erroneous.

Dropout Uncertainty. Amiri, Miller, and Savova (2018) use Monte Carlo dropout (Gal and Ghahramani 2016) to estimate the uncertainty of an underlying model. There are different acquisition methods to compute uncertainty from the stochastic passes. A summary can be found in Shelmanov et al. (2021). The work of Amiri, Miller, and Savova (2018) uses the probability variance averaged over classes.

Label Aggregation. Given T predictions obtained via Monte Carlo dropout, Amiri, Miller, and Savova (2018) use MACE (Hovy et al. 2013), an aggregation technique from crowdsourcing to adjudicate the resulting repeated predictions.

3.2.3 Training Dynamics. Methods based on training dynamics use information derived from how a model behaves during training and how predictions change over the course of its training.

Curriculum and Leitner Spotter. Amiri, Miller, and Savova (2018) train a model via curriculum learning, where the network trains on easier instances during earlier epochs and is then gradually introduced to harder instances. Instances then are ranked by how hard they were perceived during training. They also adapt the ideas of the Zettelkasten (Ahrens 2017) and Leitner queue networks (Leitner 1974) to model training. There, difficult instances are presented more often during training than easier ones. The assumption behind both of these methods is that instances that are perceived harder or misclassified more frequently are more often annotation errors than are easier ones. These two methods require that the instances can be scheduled independently. This is, for instance, not the case for sequence labeling, as the model trains on complete sentences and not individual tokens or spans. Even if they have different difficulties, they would end up in the same batch nonetheless.

Data Map Confidence. Swayamdipta et al. (2020) use the class probability for each instance's gold label across epochs as a measure of confidence. In their experiments, low confidence correlates well with an item having an incorrect label.

3.2.4 Vector Space Proximity. Approaches of this kind leverage dense embeddings of tokens, spans, and texts into a vector space and use their distribution therein. The distance of an instance to semantically similar instances is expected to be smaller than the distance to semantically different ones. Embeddings are typically obtained by using BERT-type models for tokens and spans (Devlin et al. 2019) or S-BERT for sentences (Reimers and Gurevych 2019).

Mean Distance. Larson et al. (2019) compute the centroid of each class by averaging vector embeddings of the respective instances. Items are then scored by the distance between their embedding vector to their centroid. The underlying assumption is that semantically similar items should have the same label and be close together (and thereby close to the centroid) in the vector space. In the original publication, this method was only evaluated on detecting errors in sentence classification datasets, but we extend it to also token and span classification.

k-Nearest-Neighbor Entropy. In the context of NER in clinical reports, Grivas et al. (2020) leverage the work of Khandelwal et al. (2020) regarding nearest-neighbor language models to find mislabeled named entities. First, all instances are embedded into a vector space. Then, the k nearest neighbors of each instance according to their Euclidean distance are retrieved. Their distances to the instance embedding vector are then used to compute a distribution over labels by applying softmax. An instance's score is then the entropy of its distance distribution; if it is large, it indicates uncertainty, hinting at being mislabeled. Grivas et al. (2020) only used this method qualitatively; we have turned their qualitative approach into a method that can be used to score instances automatically and evaluated it on detecting errors in both NER and sentence classification—the latter using S-BERT embeddings. This method was only evaluated on detecting errors in NER datasets, but we apply it to sentence classification as well by using S-BERT embeddings.

3.2.5 Ensembling. Ensemble methods combine the scores or predictions of several individual flaggers or scorers to obtain better performance than the sum of their parts.

Diverse Ensemble. Instead of using a single prediction like *Retag* does, the predictions of several, architecturally different models are aggregated. If most of them disagree on the label for an instance, then it is likely to be an annotation error. Alt, Gabryszak, and Hennig (2020) use an ensemble of 49 different models to find annotation errors in the TACRED relation extraction corpus. In their setup, instances are ranked by how often a model suggests a label different from the original one. Barnes, Øvrelid, and Velldal (2019) use three models to analyze error types on several sentiment analysis datasets; they flag instances for which all models disagree with the gold label. Loftsson (2009) and Angle, Mishra, and Sharma (2018) use an ensemble of different taggers to correct POS tags.

Projection Ensemble. In order to correct the CONLL-2003 named entity corpus, Reiss et al. (2020) train 17 logistic regression models on different Gaussian projections of BERT embeddings. The aggregated predictions that disagree with the dataset were then corrected by hand.

Item Response Theory. Lord, Novick, and Birnbaum (1968) developed *Item Response Theory* as a mathematical framework to model relationships between measured responses of test subjects (e.g., answers to questions in an exam) for an underlying, latent trait (e.g., the overall grasp on the subject that is tested). It can also be used to estimate the discriminative power of an item, namely, how well the response to a question can be used to distinguish between subjects of different ability. In the context of AED, test subjects are trained models, the observations are the predictions on the dataset, and the latent trait is task performance. Rodriguez et al. (2021) have shown that items that negatively discriminate (i.e., where a better response indicates being less skilled) correlate with annotation errors.

Borda Count. Similarly to combining several flaggers into an ensemble, rankings obtained from different scorers can be combined as well. For that, Dwork et al. (2001) propose leveraging Borda counts, a voting scheme that assigns points based on their ranking. For each scorer, given scores for N instances, the instance that is ranked the highest is given N points, the second-highest $N - 1$, and so on (Szpiro 2010). The points assigned by different scorers are then summed up for each instance and form the aggregated ranking. Larson et al. (2019) use this to combine scores for runs of *Mean Distance* with different embeddings and show that this improves overall performance compared to only using individual scores.

3.2.6 Rule-based. Several studies leverage rules that describe which annotations are valid and which are not. For example, to find errors in POS annotated corpora, Květoň and Oliva (2002) developed a set of conditions that tags have to fulfill in order to be valid, especially n-grams that are impossible based on the underlying lexical or morphological information of their respective surface forms. Rule-based approaches for AED can be very effective but are hand-tailored to the respective dataset, its domain, language, and task. Our focus in this article is to evaluate generally applicable methods that can be used for many different tasks and settings. Therefore, we do not discuss rule-based methods further in the current work.

4. Datasets and Tasks

In order to compare the performance of AED methods on a large scale, we need datasets with parallel gold and noisy labels. But even with previous work on correcting noisy corpora, such datasets are hard to find.

We consider three kinds of approaches to obtain datasets that can be used for evaluating AED. First, existing datasets can be used whose labels are then randomly perturbed. Second, there exist adjudicated gold corpora for which the annotations of single annotators exist. Noisy labels are then the unadjucated annotations. These kinds of corpora are mainly obtained from crowdsourcing experiments. Third, there are manually corrected corpora whose both clean and noisy parts have been made public. Because only a few such datasets are available for AED, we have derived several datasets of the first two types from existing corpora.

When injecting random noise we use flipped label noise (Zheng, Awadallah, and Dumais 2021) with a noise level of 5%, which is in a similar range to error rates in previously examined datasets like PENN TREEBANK (Dickinson and Meurers 2003b) or CONLL-2003 (Reiss et al. 2020). In our settings, for a random subset of 5% instances, this kind of noise assigns uniformly a different label from the tagset without taking the original label into account. While randomly injecting noise is simple and can be applied to any existing gold corpus, errors in these datasets are often easy to spot (Larson et al. 2019). This is because errors typically made by human annotators vary with the actual label, which is not true for random noise (Hedderich, Zhu, and Klakow 2021). Note that evaluating AED methods does not require knowing true labels: All that is required are potentially noisy labels and whether or not they are erroneous. It is only correction that needs true labels as well as noisy ones.

As noted earlier, we will address AED in three broad NLP tasks: text classification, token labeling, and span labeling. These have been the tasks most frequently evaluated in AED and on which the majority of methods can be applied. Also, these tasks have many different machine learning models available to solve them. This is crucial for evaluating calibration (§ 5.2) and assessing whether well-performing models lead to better task performance for model-based methods (§ 5.3). To foster reproducibility and to obtain representative results, we then choose datasets that fulfill the following requirements: (1) they are available openly and free of charge, (2) they are for common and different NLP tasks, (3) they come from different domains, and (4) they have high inter-annotator agreement and very few annotation errors. Based on these criteria, we select 9 datasets. They are listed in Table 2 and are described in the following section. We manually inspected and carefully analyzed the corpora to verify that the given gold labels are of very high quality.

4.1 Text Classification

The goal of text classification is to assign a predefined category to a given text sequence (here, a sentence, paragraph, or a document). Example applications are news categorization, sentiment analysis, or intent detection. For text classification, each individual sentence or document is considered its own instance.

Table 2
Dataset statistics. We report the number of instances $|\mathcal{I}|$ and annotations $| \mathrm{A} |$ as well as the number of mislabeled ones ($|\mathcal{I}_\epsilon|$ and $|\mathcal{A}_\epsilon|$), their percentage, as well as the number of classes $|\mathcal{C}|$. For token and span labeling datasets, $|\mathcal{A}|$ counts the number of annotated tokens and spans, respectively. *Kind* indicates whether the noisy part was created by randomly corrupting labels (R), or by aggregation (A) from individual annotations like crowdsourcing, or whether the gold labels stem from manual correction (M). Errors for span labeling are calculated via exact span match. *Source* points to the work that introduced the dataset for use in AED if it was created via manual correction and to the work proposing the initial dataset for aggregation or randomly perturbed ones.

Name	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{I}_\epsilon\rvert$	$\frac{\lvert\mathcal{I}_\epsilon\rvert}{\lvert\mathcal{I}\rvert}$%	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{A}_\epsilon\rvert$	$\frac{\lvert\mathcal{A}_\epsilon\rvert}{\lvert\mathcal{A}\rvert}$%	$\lvert\mathcal{C}\rvert$	Kind	Source
Text classification									
ATIS	4,978	238	4.78	4,978	238	4.78	22	R	Hemphill et al. (1990)
IMDb	24,799	499	2.01	24,799	499	2.01	2	M	Northcutt et al. (2021)
SST	8,544	420	4.92	8,544	420	4.92	2	R	Socher et al. (2013)
Token labeling									
GUM	7,397	3,920	52.99	137,605	6,835	4.97	18	R	Zeldes (2017)
Plank	500	373	74.60	7,876	931	11.82	13	A	Plank et al. (2014a)
Span labeling									
CoNLL-2003	3,380	217	6.42	5,505	262	4.76	5	M	Reiss et al. (2020)
SI Companies	500	224	44.80	1,365	325	23.81	11	M	Larson et al. (2020)
SI Flights	500	43	8.60	1,196	49	4.10	7	M	Larson et al. (2020)
SI Forex	520	63	12.12	1,263	98	7.76	4	M	Larson et al. (2020)

ATIS contains transcripts of user interactions with travel inquiry systems, annotated with intents and slots. For AED on intent classification, we have randomly perturbed the labels.

IMDb contains movie reviews labeled with sentiment. Northcutt, Athalye, and Mueller (2021) discovered that it contains a non-negligible amount of annotation errors. They applied *Confident Learning* to the test set and let crowdworkers check whether the flags were genuine.

SST The STANFORD SENTIMENT TREEBANK is a dataset for sentiment analysis of movie reviews from Rotten Tomatoes. We use it for binary sentiment classification and randomly perturb the labels.

4.2 Token Labeling

The task of token labeling is to assign a label to each token. The most common task in this category is POS tagging. As there are not many other tasks with easily obtainable datasets, we only use two different POS tagging datasets. For token labeling, each individual token is considered an instance.

GUM The GEORGETOWN UNIVERSITY MULTILAYER CORPUS is an open source corpus annotated with several layers from the Universal Dependencies project (Nivre et al. 2020). It has been collected by linguistics students at Georgetown University as part of their course work. Here, the original labels have been perturbed with random noise.

Plank POS contains Twitter posts that were annotated by Gimpel et al. (2011). Plank, Hovy, and Søgaard (2014a) mapped their labels to Universal POS tags and had 500 tweets reannotated by two new annotators. We flag an instance as erroneous if its two annotations disagree.

4.3 Span Labeling

Span labeling assigns labels not to single tokens, but to spans of text. Common tasks that can be modeled that way are NER, slot filling, or chunking. In this work, we assume that spans have already been identified, focusing only on finding label errors and leaving detecting boundary errors and related issues for future work. We use the following datasets:

CoNLL-2003 is a widely used dataset for NER (Tjong Kim Sang and De Meulder 2003). It consists of news wire articles from the Reuters Corpus annotated by experts. Reiss et al. (2020) discovered several annotation errors in the English portion of the dataset. They developed *Projection Ensembles* and then manually corrected the instances flagged by it. While errors concerning tokenization and sentence splitting were also corrected, we ignore them here as being out of scope of the current study. Therefore, we report slightly fewer instances and errors overall in Table 2. Wang et al. (2019) also corrected errors in CONLL-2003 and named the resulting corpus CONLL++. As they only re-annotated the test set and found fewer errors, we use the corrected version of Reiss et al. (2020).

Slot Inconsistencies is a dataset that was created by Larson et al. (2020) to investigate and classify errors in slot filling annotations. It contains documents of three domains (COMPANIES, FOREX, FLIGHTS) that were annotated via crowdsourcing. Errors were then manually corrected by experts.

Span labeling is typically indicated using Begin-Inside-Out (BIO) tags.[2] When labeling a span as `X`, tokens outside the span are labeled `O`, the token at the beginning of the span is labeled `B-X`, and tokens within the span are labeled `I-X`. Datasets for span labeling are also usually represented in this format.

This raises the issues of (1) boundary differences and (2) split entities. First, for model-based methods, models might predict different spans and span boundaries from the original annotations. In many evaluation datasets, boundary issues were also corrected and therefore boundaries for the same span in the clean and noisy data can be different, which makes evaluation difficult. Second, for scorers it does not make much sense to order BIO tagged tokens independently of their neighbors or to alter only parts of a sequence to a different label. This can lead to corrections that split entities, which is often undesirable. Therefore, directly using BIO tags as the granularity of detection and correction for span labeling is problematic.

Hence, we suggest converting the BIO tagged sequences back to a span representation consisting of begin, end, and label. This first step solves the issue of entities potentially being torn apart by detection and correction. Spans from the original data and from the model predictions then need to be aligned for evaluation in order to reduce boundary issues. This is depicted in Figure 1.

[2] For simplicity, we describe the BIO tagging format. There are more advanced schemas like BIOES, but our resulting task-specific evaluation is independent of the actual schema used.

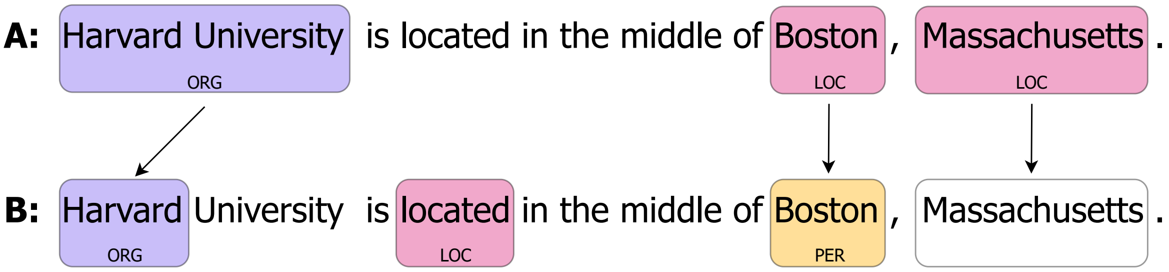


Figure 1
Alignment between original or corrected spans A and noisy or predicted spans B. The goal is to find an alignment that maximizes overlap. Spans that are in A but find no match in B are given a match with the same offsets but a special, unique label that is different from all other labels (e.g., *Massachusetts*). Spans that are in B but find no match in A are dropped (e.g., *located*). Spans from A that have no overlapping span in B are considered different and cannot be aligned (e.g., *Boston* in A and *Massachusetts* in B). Span colors here indicate their labels.

We require a good alignment to (1) maximize overlap between aligned spans so that the most likely spans are aligned, (2) be deterministic, (3) not use additional information like probabilities, and (4) not align spans that have no overlap to avoid aligning things that should not be aligned. If these properties are not given, then the alignment and resulting confidences or representations that are computed based on this can be subpar. This kind of alignment is related to evaluation, for example, for NER in the style of MUC-5 (Chinchor and Sundheim 1993), especially for partial matching. Their alignment does not, however, satisfy (1) and (3) in the case of multiple predictions overlapping with a single gold entity. For instance, if the gold entity is *New York City* and the system predicted *York* and *New York*, then in most implementations, the first prediction is chosen and other predictions that also could match are discarded. What prediction is first depends on the order of predictions which is non-deterministic. This also does not choose the optimal alignment with maximum span overlap, which requires a more involved approach.

We thus adopt the following alignment procedure: Given a sequence of tokens, a set of original spans A and predicted/noisy spans B, align both sets of spans and thereby allow certain leeway of boundaries. The goal is to find an assignment that maximizes overlap of spans in A and B; only spans of A that overlap in at least one token with spans in B are considered. This can be formulated as a linear sum assignment problem: Given two sets A, B of equal size and a function that assigns a cost to connect an element of A with an element of B, find the assignment that minimizes the overall cost (Burkard, Dell'Amico, and Martello 2012). It can happen that not all elements of A are assigned a match in B and vice versa—we assign a special label that indicates missing alignment in the first case and drop spans of B that have no overlap in A. For the latter, it is possible to also assign a special label to indicate that a gold entity is missing; in this work, we focus on correcting labels only and hence leave using this information to detect missing spans for future work.

We are not aware of previous work that proposes a certain methodology for this. While Larson et al. (2020) evaluate AED on slot filling, it is not clear on which granularity they measure detection performance or whether and how they align. To the best of our knowledge, we are the first to propose this span alignment approach for span-level AED. Span alignment requires aggregating token probabilities into span probabilities, which is described in § 1.1. This alignment approach can also be extended to other tasks like object classification or matching boxes for optical character recognition. In that case, the metric to optimize is the Jaccard index.

5. Experiments

In this section we first define the general evaluation setup, metrics to be used, and the models that are leveraged for model-based AED. Details on how each method was implemented for this work can be found in Appendix A. In § 5.1 through § 5.4, we then describe our results for the experiments we perform to answer the research questions raised in § 1.

Metrics. As described in § 3.1, we differentiate between two kinds of annotation error detectors, *flaggers* and *scorers*. These need different metrics during evaluation, similar to unranked and ranked evaluation from information retrieval (Manning, Raghavan, and Schütze 2008). Flagging is a binary classification task. Therefore, we use the standard metrics for this task, which are precision, recall, and F1. We also record the percentage of instances flagged (Larson et al. 2020). Scoring produces a ranking, as in information retrieval. We use average precision[3] (AP), Precision@10%, and Recall@10%, similarly to Amiri, Miller, and Savova (2018) and Larson et al. (2019). There are reasons why both precision and recall can be considered the more important metric of the two. A low precision leads to increased cost because many more instances than necessary need to be inspected manually after detection. Similarly, a low recall leads to problems because there still can be errors left after the application of AED. As both arguments have merit, we will mainly use the aggregated metrics F1 and AP. Precision and recall at 10% evaluate a scenario in which a scorer was applied and the first 10% with the highest score—most likely to be incorrectly annotated—are manually corrected. We use the PYTREC-EVAL toolkit to compute these ranking metrics.[4] Recall relies on knowing the exact number of correctly and incorrectly annotated instances. While this information may be available when developing and evaluating AED methods, it is generally not available when actually applying AED to clean real data. One solution to computing recall then is to have experts carefully annotate a subset of the data and then use it to estimate recall overall.

In contrast to previous work, we explicitly do not use ROC AUC and discourage its use for AED, as it heavily overestimates performance when applied to imbalanced datasets (Davis and Goadrich 2006; Saito and Rehmsmeier 2015). Datasets needing AED are typically very imbalanced because there are far more correct labels than incorrect ones.

Models. We use multiple different neural and non-neural model types per task for model-based AED. These are used to investigate the relationship between model and method performances, whether model calibration can improve method performances and for creating diverse ensembles.

For text classification we use seven different models: logistic regression as well as gradient boosting machines (Ke et al. 2017) with either bag-of-word or S-BERT features (Reimers and Gurevych 2019), transformer based on DistilRoBERTa (Sanh et al. 2019),

[3] Also known as Area Under the Precision-Recall Curve (AUPR/AUPRC). In AED, AP is also identical to mean average precision (mAP) used in other works.

[4] `https://github.com/cvangysel/pytrec_eval`.

BiLSTM based on Flair (Akbik et al. 2019), and FastText (Joulin et al. 2017). For S-BERT, we use `all-mpnet-base-v2` as the underlying model, as it has been shown by their creators to produce sentence embeddings of the highest quality overall.

For token and span labeling, we use four different models: CRFs with the handcrafted features as proposed by Gimpel et al. (2011), BiLSTM + CRF based on Flair (Akbik et al. 2019), transformers with CRF based on DistilRoBERTa (Sanh et al. 2019), and logistic regression (also called maximum entropy model). For the initialization of Flair-based models we use a combination of GloVe (Pennington, Socher, and Manning 2014) as well as Byte-Pair Encoding embeddings (Heinzerling and Strube 2018) and a hidden layer size of 256 for both text classification and sequence labeling. Note that we do not perform extensive hyperparameter tuning for model selection because when using AED in practice, no annotated in-domain data can be held out for tuning since all data must be checked for errors. Also, when comparing models as we do here, it would be prohibitively expensive to carry out hyperparameter tuning across all datasets and model combinations. Instead, we use default configurations that have been shown to work well on a wide range of tasks and datasets.

When using transformers for sequence labeling we use the probabilities of the first subword token. We use 10-fold cross-validation to train each model and use the same model weights for all methods evaluated on the same fold. Thereby, all methods applied to the same fold use the predictions of the same model.

5.1 RQ1 – Which Methods Work Well across Tasks and Datasets?

We first report the scores resulting from the best setup as a reference to the upcoming experiments. Then we describe the experiments and results that lead to this setup. We do not apply calibration to any of the methods for the reported scores because it only marginally improved performance (see § 5.2). For model-based methods, the best performance for text classification and span labeling was achieved using transformers; for token labeling, best performance was achieved using Flair (see § 5.3). Not using cross-validation for model-based AED was found to substantially reduce recall for model-based AED (see § 5.4), so we have used 10-fold cross-validation in comparing model-based methods.

In Table 3, we present the overall performance in F1 and AP across all datasets and tasks. Detailed results including scores for all metrics can be found in Appendix C.

First of all, it can be seen that in datasets with randomly injected noise (ATIS, SST, and GUM), errors are easier to find than in aggregated or hand-corrected ones. Especially in ATIS, many algorithms reach close-to-perfect scores, in particular scorer ($>$ 0.9 AP). We attribute this to the artificial noise injected. The more difficult datasets have usually natural noise patterns that are often harder to solve (Amiri, Miller, and Savova 2018; Larson et al. 2019; Hedderich, Zhu, and Klakow 2021). The three SLOT INCONSISTENCIES datasets are also easy compared to CONLL-2003. On some datasets with real errors—PLANK and SLOT INCONSISTENCIES—the performance of the best methods is already quite good with F1 $\approx$ 0.5 and AP $\approx$ 0.4 for PLANK and F1, AP $>$ 0.65 for SLOT INCONSISTENCIES.

Overall, methods that work well are *Classification Uncertainty* (CU), *Confident Learning* (CL), *Curriculum Spotter* (CS), *Datamap Confidence* (DM), *Diverse Ensemble* (DE), *Label Aggregation* (LA), *Leitner Spotter* (LS), *Projection Ensemble* (PE), and *Retag* (RE). Aggregating scorer judgments via *Borda Count* (BC) can improve performance and deliver the second-best AP score based on the harmonic mean. The downside here is very high total runtime (the sum of runtimes of individual scores aggregated), as it requires training

Table 3
F1 and **AP** for all implemented flaggers 🟠 as well as scorers 🟣 evaluated with the best overall setups. We also report the harmonic mean **H** 🔵 computed across all datasets. **L**abel **A**ggregation, **Re**tag, **D**iverse **E**nsemble, and **B**orda **C**ount perform especially well across tasks and datasets. Datasets created via injecting random noise (ATIS, SST, and GUM) are comparatively easier to detect errors in.

	Text			Token		Span				
Method	ATIS	IMDb	SST	GUM	Plank	Comp.	CoNLL	Flights	Forex	H
Flagger										
CL	0.35	0.33	0.34	0.80	0.37	0.50	0.24	0.42	0.57	0.39
DE	0.72	0.30	0.33	0.74	0.48	0.57	0.28	0.55	0.64	0.45
IRT	0.00	0.01	0.02	0.00	0.12	0.41	0.29	0.02	0.62	0.00
LA	0.83	0.33	0.35	0.68	0.49	0.59	0.30	0.66	0.70	0.48
PE	0.54	0.18	0.34	0.58	0.50	0.56	0.25	0.29	0.56	0.36
RE	0.81	0.33	0.34	0.69	0.49	0.64	0.32	0.67	0.70	0.49
VN	·	·	·	0.55	0.30	0.11	0.02	0.29	0.14	0.08
Scorer										
BC	0.98	0.35	0.50	0.92	0.38	0.68	0.14	0.49	0.54	0.41
CS	0.97	0.29	0.21	·	·	·	·	·	·	0.33
CU	0.87	0.28	0.27	0.98	0.42	0.70	0.17	0.68	0.70	0.41
DM	0.98	0.25	0.49	0.95	0.27	0.66	0.14	0.35	0.61	0.36
DU	0.05	0.06	0.05	0.05	0.24	0.43	0.07	0.18	0.32	0.08
KNN	0.13	0.05	0.11	0.21	0.31	0.61	0.12	0.07	0.16	0.12
LE	·	·	·	0.60	0.22	0.41	0.19	0.10	0.11	0.18
LS	0.91	0.31	0.46	·	·	·	·	·	·	0.46
MD	0.14	0.03	0.08	0.12	0.16	0.54	0.06	0.07	0.14	0.08
PM	0.06	0.05	0.05	0.05	0.23	0.54	0.06	0.12	0.25	0.08
WD	·	·	·	0.53	0.39	0.45	0.16	0.11	0.14	0.20

instances of all scorers beforehand, which already perform very well (H_{AP} of *Borda Count* is 0.41 and the best individual scorer has H_{AP} of 0.46). While aggregating scores requires well performing scorers (3 in our setup, see § 1.2) it is more stable across tasks than using individual methods on their own. Most model-based methods (*Classification Uncertainty*, *Confident Learning*, *Diverse Ensemble*, *Label Aggregation*, *Retag*) perform very well overall, but methods based on training dynamics that do not need cross-validation (*Curriculum Spotter*, *Datamap Confidence*, *Leitner Spotter*) are on par or better. In particular, *Datamap Confidence* shows a very solid performance and can keep up with the closely related *Classification Uncertainty*, sometimes even outperforming it while not needing CV. *Confident Learning* specifically has high precision for token and span labeling.

Amiri, Miller, and Savova (2018) argue that prediction loss is not enough to detect incorrect instances because easy ones still can have a large loss. Therefore, more intricate methods like *Leitner Spotter* and *Curriculum Spotter* are needed. We do not observe a large difference between *Classifier Uncertainty* and the two, though. *Datamap Confidence*, as a more complicated sibling of *Classification Uncertainty*, however, outperforms these from time to time, indicating that training dynamics offers an advantage over simply using class probabilities.

Variation n-grams (VN) has high precision and tends to be conservative in flagging items, that is, exhibit low false positives, especially for span classification. *Weighted Discrepancy* works overall better than *Label Entropy*, but both methods almost always perform worse than more intricate ones. When manually analyzing their scores, they mostly assign a score of 0.0 and rarely a different score (less than 10% from our observation, often even lower). This is because there are only very few instances with both surface form overlap and different labels. While the scores for *Prediction Margin* appear to be not good, the original paper (Dligach and Palmer 2011) reports a similarly low performance while their implementation of *Retag* reaches scores that are around two times higher (10% vs. 23% precision and 38% vs. 60% recall). This is similar to our observations. One potential reason why *Classification Uncertainty* produces better results than the related *Prediction Margin* is that the latter does not take the given label into account; it always uses the difference between the two most probable classes. Using a formulation of *Projection Ensemble* that uses the label did not improve results significantly, though.

Methods based on vector proximity—*k-Nearest Neighbor Entropy* (KNN) and *Mean Distance* (MD)—perform sub-par across tasks and datasets. We attribute this to issues in distance calculation for high-dimensional data, as noted for instance by Cui Zhu, Kitagawa, and Faloutsos (2005) in a related setting. In high-dimensional vector spaces, everything can appear equidistant (curse of dimensionality). Another performance-relevant issue is the embedding quality. In Grivas et al. (2020), KNN is used with domain-specific embeddings for biomedical texts. These could have potentially improved performance in their setting, but they do not report quantitative results, though, which makes a comparison difficult. With regard to *Mean Distance*, we only achieve $H = 0.08$. On real data for intent classification, Larson et al. (2019) achieve an average precision of around 0.35. They report high recall and good average precision on datasets with random labels but do not report precision on its own. Their datasets contain mainly paraphrased intents, which makes it potentially easier to achieve good performance. This is similar to how AED applied on our randomly perturbed ATIS dataset resulted in high detection scores. Code and data used in their original publication are no longer available. We were therefore not able to reproduce their reported performances with our implementation and on our data.

Item Response Theory (IRT) does not perform well across datasets and tends to overly flag instances. Therefore, it is preferable to use the model predictions in a *Diverse Ensemble*, which yields much better performance. IRT is also relatively slow for larger corpora as it is optimized via variational inference and needs many iterations to converge. Our hypothesis is that *Item Response Theory* needs more subjects (in our case models) to better estimate discriminability. Compared to our very few subjects (seven for text classification and four for token and span labeling), Rodriguez et al. (2021) used predictions of the SQuAD leaderboard with 161 development and 115 test subjects. To validate this hypothesis, we rerun *Item Response Theory* on the unaggregated predictions of *Projected Ensemble*. While this leads to slightly better performance, it still does not work as well as using predictions in *Diverse Ensemble* or *Projected Ensemble* directly. As it is often unfeasible to have that many models providing predictions, we see *Item Response Theory* only useful in very specific scenarios.

Regarding *Dropout Uncertainty*, after extensive debugging with different models, datasets, and formulations of the method, we were not able to achieve comparably good results with other AED methods evaluated in this work. On real data, Amiri, Miller, and Savova (2018) also report relatively low performances similar to ours. Our implementation delivers results similar to Shelmanov et al. (2021) on misclassification detection. In their paper, the reported scores appear to be very high. But we consider their reported scores an overestimate, as they use ROC AUC (which is overconfident for imbalanced datasets) and not AP to evaluate their experiments. Even when applying the method on debug datasets with the most advantageous conditions that are solvable by other methods with perfect scores, *Dropout Uncertainty* only achieves AP values of around 0.2. The main reason we see for the overall low scores for *Dropout Uncertainty* is that the different repeated prediction probabilities are highly correlated and do not differ much overall. This is similar to the observations of Shelmanov et al. (2021).

Qualitative Analysis. To better understand for which kinds of errors methods work well or fail, we manually analyze the instances in CONLL-2003. It is our dataset of choice for three reasons: (1) span labeling datasets potentially contain many different errors, (2) it is annotated and corrected by humans, and (3) it is quite difficult for AED to find errors in it, based on our previous evaluation (see Table 3). For spans whose noisy labels disagree with the correction, we annotate them as either being inconsistent, a true error, an incorrect correction, or a hallucinated entity. Descriptions and examples for each type of error are given in the following.

True errors are labels that are unambiguously incorrect, for instance, in the sentence *NATO military chiefs to visit Iberia*, the entity *Iberia* was annotated as ORG but should be LOC, as it refers to the Iberian peninsula.

Inconsistencies are instances that were assigned different labels in similar contexts. In CONLL-2003, these are mostly from sports teams that were sometimes annotated as LOC and sometimes as ORG.

Incorrect correction In very few cases, the correction introduced a new error, for example, *United Nations* was incorrectly corrected from ORG to LOC.

Hallucinated entity are spans that were labeled to contain an entity, but they should not have been annotated at all. For example, in the sentence *Returns on treasuries were also in negative territory*, *treasuries* was annotated as MISC but does not contain a named entity. Sometimes, entities that should consist of one span were annotated originally as two entities. This results in one unmatched entity after alignment. We consider this a hallucinated entity as well.

CONLL-2003 was corrected manually by Reiss et al. (2020). After aligning (see § 4.3), we find that there are in total 293 errors. We group them by difficulty based on how often methods were able to detect them. For scorers, we consider the instances with the highest 10% scores as flagged, similarly to how we evaluate precision and recall. For span labeling, we implemented a total of 16 methods. The errors detected at least by half of the methods (50%) are considered *easy*, the ones detected by at least four methods (25%) are considered *medium*, and the rest, *hard* (25%). This results in 50 easy, 78 medium, and 165 hard instances. The distribution of error types across difficulty levels is visualized in Figure 2. It can be seen that true errors are easier to detect than inconsistencies by a significant margin: The easy partition consists only of 50% inconsistencies, whereas in the hard partition, it consists of around 75% inconsistent instances. This can be intuitively explained by the fact that the inconsistencies are not rare, but make up a large fraction of all corrections. It is therefore difficult for a method to learn that it should be flagged when it is only given noisy labels.

We further analyze how each method can deal with the different types of errors across difficulty levels. The percentage of correctly detected errors per type and method is depicted in Table 4. It again can be seen that true errors are easier for methods to detect than inconsistencies; inconsistencies of hard difficulty were almost never detected. Interestingly, scorers that are not model-based (*k-Nearest Neighbor Entropy* (KNN), *Label Entropy* (LE), and *Weighted Discrepancy* (WD)) are able to better detect inconsistencies of

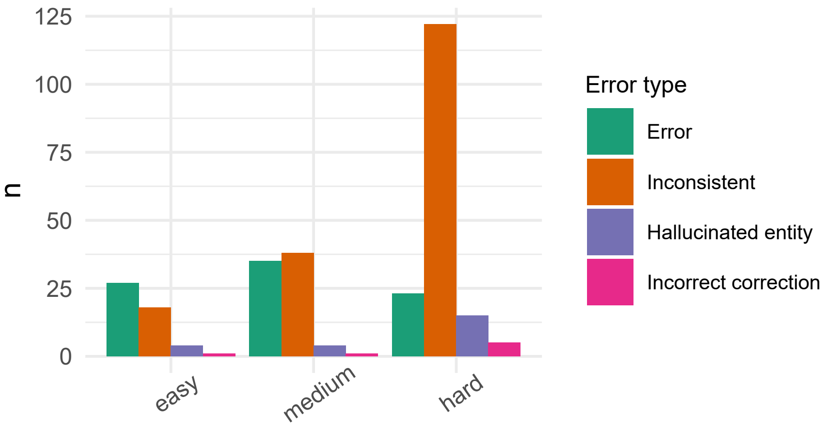


Figure 2
Error counts per difficulty level in CONLL-2003. It can be seen that the number of inconsistencies increases with the difficulty. This indicates that "real" annotation errors are easier to detect than inconsistencies.

Table 4
Percent of errors and inconsistencies detected on CONLL-2003 across methods and difficulty for flaggers ● and scorers ● grouped by error types. It can be seen that real errors (E) are more often detected than inconsistencies (I). Some methods not relying on models (KNN, LE, WD) are sometimes better in spotting inconsistencies than errors, whereas for model-based method it is the opposite. Note that errors concerning incorrect corrections (IC) and hallucinated entities (HE) are quite rare and not reliable to draw conclusions from.

	Flagger							Scorer									
Error	CL	DE	IRT	LA	PE	RE	VN	BC	CU	DM	DU	KNN	LE	MD	WD	PM
Easy																	
E	66	96	100	100	100	100	3	92	100	66	25	40	29	14	29	25
I	72	100	100	88	94	94	11	94	100	55	27	61	55	11	55	27
HE	25	100	100	100	100	100	0	100	100	100	100	0	0	50	0	100
IC	0	100	100	100	100	100	0	100	100	100	0	0	0	100	0	0
Medium																	
E	40	51	54	88	82	82	0	31	94	31	34	22	0	11	0	25
I	13	26	31	26	52	31	0	15	36	23	21	44	63	15	63	13
HE	0	75	75	25	75	100	0	75	50	75	25	0	0	0	0	50
IC	100	100	100	100	100	100	0	0	100	0	0	0	0	0	0	0
Hard																	
E	0	17	13	0	65	0	0	0	0	17	0	17	0	17	0	13
I	0	0	0	0	9	0	0	3	0	5	4	14	4	1	4	4
HE	0	13	13	0	53	6	0	0	0	0	6	0	0	6	0	0
IC	0	20	20	20	20	0	0	0	0	0	20	0	0	20	0	0

medium and sometimes hard difficulty but fail at detecting most errors. We explain this for KNN by the fact that it relies on semantic vector space embeddings that do not rely on the noisy label but on the semantics of its surface form in its context. As neighbors in the space have the same meaning, it is possible to detect errors even though the label is inconsistent in many cases. The same can be said about WD and LE, which rely only on the surface form and how often it is annotated differently. If the correct label is the majority, then it can still detect inconsistencies even if they are quite frequent. But both still do not perform as well as other methods on easier instances; they only find 50% of errors and inconsistencies whereas *Classification Uncertainty* or *Retag* detects almost all (but again fail to find inconsistencies on medium difficulty). *Variation n-grams* (VN), however, do not work well even for easy cases because they rely on contexts around annotations that need to match exactly, which is very rare in this dataset. To summarize, the methods that worked best overall across tasks and datasets are *Borda Count* (BC), *Diverse Ensemble* (DE), *Label Aggregation* (LA), and *Retag* (RE). Inconsistencies appear to be more difficult to detect for most methods, especially for model-based ones. Methods that do not rely on the noisy labels like *k-Nearest Neighbor Entropy*, *Label Entropy*, and *Weighted Discrepancy* were better in finding inconsistencies on more difficult instances when manually analyzing CONLL-2003.

5.2 RQ2 – Does Model Calibration Improve Model-based Method Performance?

Several model-based AED methods, for instance, *Classification Uncertainty*, directly leverage probability estimates provided by a machine learning model (§ 3.2.2). Therefore, it is of interest whether models output class probability distributions that are accurate. For instance, if a model predicts 100 instances and states for all 80% confidence, then the accuracy should be around 0.8. If this is the case for a model, then it is called **calibrated**. Previous studies have shown that models are often not calibrated very well, especially neural networks (Guo et al. 2017). To alleviate this issue, a number of calibration algorithms have been developed. The most common approaches are post hoc, which means that they are applied after the model has already been trained.

Probabilities that are an under- or overestimate can lead to non-optimal AED results. The question arises whether model-based AED methods can benefit from calibration and, if so, to what extent. We are only aware of one study mentioning calibration in the context of annotation error detection. Northcutt, Jiang, and Chuang (2021) claim that their approach does not require calibration to work well, but they did not evaluate it in detail. We only evaluate whether calibration helps for approaches that directly use probabilities and can leverage CV, as calibration needs to be trained on a holdout set. This, for instance, excludes *Curriculum Spotter*, *Leitner Spotter*, and *Datamap Confidence*. Methods that can benefit are *Confident Learning*, *Classifier Uncertainty*, *Dropout Uncertainty*, and *Prediction Margin*.

There are two groups of approaches for post hoc calibration: *parametric* (e.g., Platt Scaling/Logistic Calibration [Platt 1999] or Temperature Scaling [Guo et al. 2017]) or *non-parametric* (e.g., Histogram Binning [Zadrozny and Elkan 2001], Isotonic Regression [Zadrozny and Elkan 2002], or Bayesian Binning into Quantiles [Naeini, Cooper, and Hauskrecht 2015]). On a holdout corpus we evaluate several calibration methods to determine which calibration method to use (see Appendix B). As a result, we apply the best—Platt Scaling—for all experiments that leverage calibration.

Calibration is normally trained on a holdout set. As we already perform cross-validation, we use the holdout set both for training the calibration and for predicting annotation errors. While this would not be optimal if we are interested in generalizing calibrated probabilities to unseen data, we are more interested in downstream task performance. Using an additional fold per round would be theoretically more sound. But our preliminary experiments show that it has the issue of reducing the available training data and thereby hurts the error detection performance more than the calibration helps. Using the same fold for both calibration and applying AED, however, improves overall task performance, which is what matters in our special task setting. We do not leak the values for the downstream tasks (whether an instance is labeled incorrectly or not) but only the labels for the primary task.

To evaluate whether calibration helps model-based methods that leverage probabilities, we train models with cross-validation and then evaluate each applicable method with and without calibration. The same model and therefore the same initial probabilities are used for both. We measure the relative and total improvement in F1 (for flaggers) and AP (for scorers), which are our main metrics. The results are depicted in Figure 3. It can be seen that calibration has the potential of improving the performance of certain methods by quite a large margin. For *Confident Learning*, the absolute gain is up to 3 percentage points (pp) F1 on text classification, 5 pp for token labeling, and up to 10 pp for span labeling. On the latter two tasks, though, there are also many cases with performance reductions. A similar pattern can be seen for *Classification Uncertainty* with up to 2 pp, no impact, and up to 8 pp, respectively. *Dropout Uncertainty* and *Prediction Margin* do not perform well to begin with. But after calibration, they gain 5 to 10 pp AP, especially for span and in some instances for token labeling. In most cases on median, calibration does not hurt the overall performance.

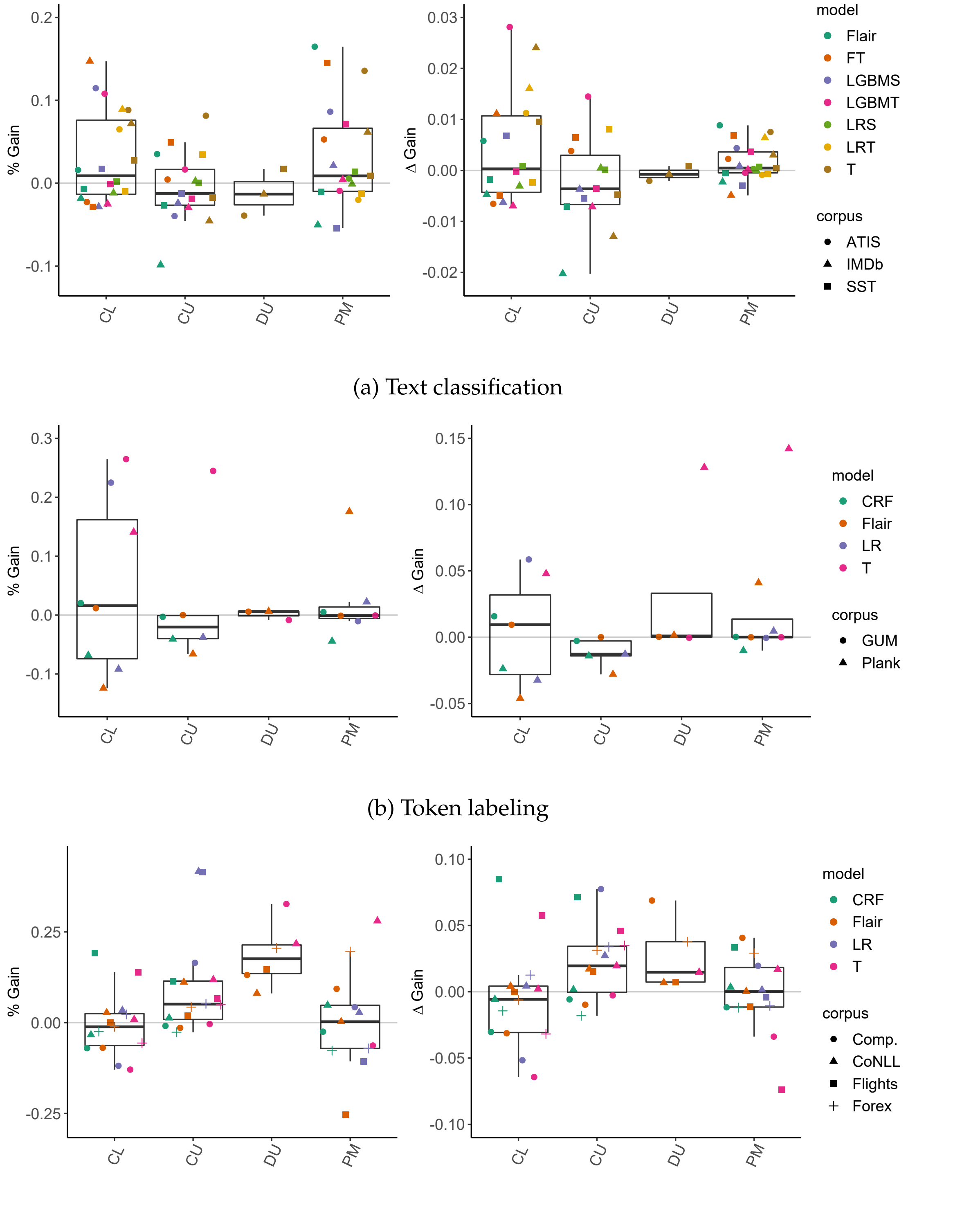
In order to check whether the improvement using calibration is statistically significant, we also use statistical testing. We choose the Wilcoxon signed-rank test (Wilcoxon 1945) because the data is not normally distributed, which is required by the more powerful paired t-test. The alternative hypothesis is that calibration improves method performance, resulting in a one-sided test.

We do not perform a multiple-comparison correction as each experiment works on different data. The p-values can be seen in Table 5. We can see that calibration can improve performance significantly overall in two task and method combinations (text classification + *Confident Learning* and span labeling + *Classification Uncertainty*). For text classification and token labeling, the absolute gain is relatively small. For span labeling, *Classification Uncertainty* benefits the most. The gains for *Dropout Uncertainty* and *Prediction Margins* appear large, but these methods do not perform well in the first place. Hence, our conclusion is that calibration can help model-based AED performance but it is very task- and dataset-specific.

We do not see a clear tendency for which models or datasets benefit the most from calibration. More investigation is needed regarding which calibrator works best for which model and task. We chose the one that reduces the calibration error the most, which is not necessarily the best choice for each setting.

5.3 RQ3 – To What Extent Are Model and Detection Performance Correlated?

Several AED methods directly use model predictions or probabilities to detect potential annotation errors. This raises the question of how model performance impacts AED performance. Reiss et al. (2020) state that they deliberately use simpler models to find more potential errors in CONLL-2003 and therefore developed *Projection Ensemble*, an ensemble of logistic regression classifiers that use BERT embeddings reduced by different Gaussian projections. Their motivation is to obtain a diverse collection of predictions to have disagreements. They conjecture that using very well-performing models might be detrimental to AED performance as their predictions potentially would not differ that much from the noisy labels as the models learned predicting the noise. In contrast to that, Barnes, Øvrelid, and Velldal (2019) use state-of-the-art models to find annotation errors in different sentiment datasets. But neither Reiss et al. (2020) nor Barnes, Øvrelid, and Velldal (2019) directly evaluate AED performance—rather, they use AED to clean noisy datasets for which the gold labels are unknown. Therefore, the question of how much model and detection performance are correlated has not yet been thoroughly evaluated.



(a) Text classification

(b) Token labeling

(c) Span labeling

Figure 3
Relative and total improvement of model-based AED methods over different corpora, methods, and models when calibrating probabilities. It can be seen that calibration can lead to good improvements, while on median mostly not hurting performance. This plot is best viewed in the electronic version of this paper. Not displayed are extreme (positive) outlier points.

Table 5
p-values forWilcoxon signed-rank test.We check whether calibration improves AED performance on a statistically significant level. Underlined values are significant with $p < 0.05$.

Method	Text	Token	Span
Confident Learning (CL)	<u>0.021</u>	0.230	0.665
Classification Uncertainty (CU)	0.121	0.320	<u>0.003</u>
Dropout Uncertainty (DU)	0.750	0.188	0.320
Prediction Margin (PM)	0.064	0.273	0.628

For answering this question, we leverage the fact that we implemented several models of varying performance for each task. We use two complementary approaches to analyze this question. First, we measure the correlation between model and task performances for the overall score, precision, and recall. Then, we analyze which models lead to the best AED performance.

Throughout this section, scores for flaggers and scorers are coalesced; overall score corresponds to F1 and AP, precision to precision and precision@10%, recall to recall and recall@10%. For reference, model performances are shown in Figure C.1. We choose micro aggregation for measuring model performance, as we are interested in the overall scores and not the scores per class. Using macro aggregation yields qualitatively similar but less significant results.

Correlation. In order to determine whether there exists a positive or negative relationship between model and method performances, we compute Kendall's τ coefficient (Kendall 1938) for each method and dataset. The results are depicted in Table 6. We see that when the test is significant with $p < 0.05$, then there is almost always a moderate to strong monotonic relationship.[5] τ is zero or positive for classification and token labeling, hinting that there is either no relationship or a positive one. For span labeling we observe negative correlation for precision and overall. It is significant in one case only.

One issue with this test is its statistical power. In our setting, it is quite low due to the few samples available per method and task. It is therefore likely that the null hypothesis—in our case, the assumption that there is no relationship between model and method performances—is not rejected even if it should have been. Hence, we next perform additional analysis to see which models overall lead to the best model performances.

Which Models Lead to the Best Method Performances. In order to further analyze the relationship between model and AED performances, we look at which model leads to the best performance on a given dataset. In Figure 4 we show the results differentiated by overall, precision, and recall scores. We observe that in the most cases, the best or second best models lead to the best method performances. It is especially clear for token labeling, where using Flair leads to the best performance in all cases if we look at the overall and precision score. Interestingly, Flair has better performance than transformers for span labeling but the latter is preferred by most methods. Flair only leads to best method performances for most of CONLL-2003 and parts of FLIGHTS. Besides the fact that better models on average lead to better AED performance, we do

[5] $|\tau| > 0.07$ indicates a weak, $|\tau| > 0.21$ a moderate, $|\tau| > 0.35$ indicates a strong monotonic relation.

Table 6
Kendall's τ coefficient grouped by task and method measured across datasets. For p, the null hypothesis is $\tau = 0$ and the alternative hypotheses is $\tau \neq 0$. Underlined are significant p-values with $p < 0.05$. Positive correlation is highlighted 🔵, negative correlation is highlighted 🔴.

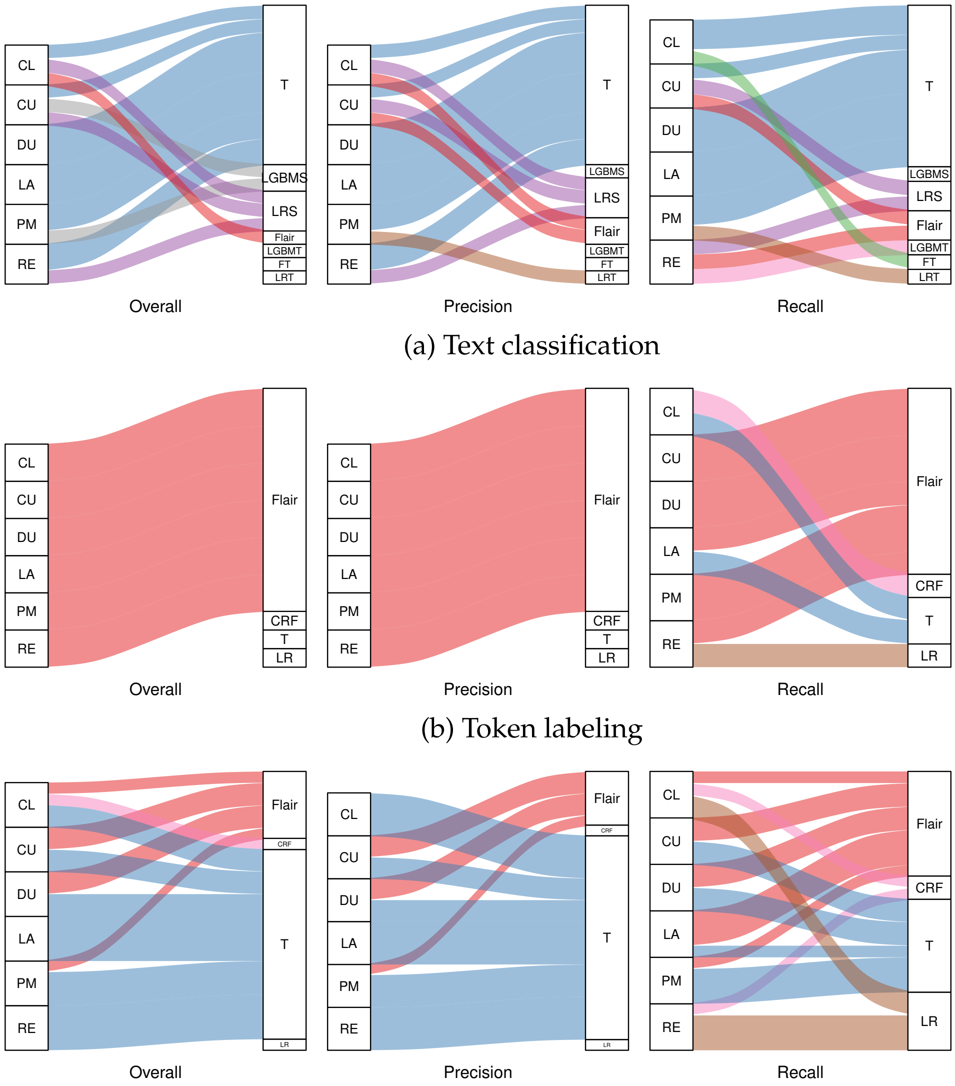
	Overall		Precision		Recall	
Method	τ	p	τ	p	τ	p
Text						
CL	+0.495	$\underline{0.002}$	+0.657	$\underline{0.000}$	−0.373	$\underline{0.018}$
CU	+0.486	$\underline{0.002}$	+0.396	$\underline{0.013}$	+0.705	$\underline{0.000}$
DU	+0.333	0.602	+0.333	0.602	+0.333	0.602
LA	+0.333	0.602	+1.000	0.117	+0.333	0.602
PM	+0.143	0.365	+0.211	0.184	+0.230	0.147
RE	+0.571	$\underline{0.000}$	+0.600	$\underline{0.000}$	+0.412	$\underline{0.011}$
Token						
CL	+0.929	$\underline{0.001}$	+0.929	$\underline{0.001}$	+0.214	0.458
CU	+0.714	$\underline{0.013}$	+0.786	$\underline{0.006}$	+0.714	$\underline{0.013}$
DU	−0.333	0.497	−0.333	0.497	+0.333	0.497
LA	+1.000	$\underline{0.042}$	+0.667	0.174	+0.667	0.174
PM	+0.000	1.000	+0.000	1.000	+0.000	1.000
RE	+0.857	$\underline{0.003}$	+0.714	$\underline{0.013}$	+0.357	0.216
Span						
CL	−0.033	0.857	−0.183	0.322	+0.101	0.588
CU	+0.017	0.928	−0.250	0.177	+0.300	0.105
DU	−0.429	0.138	−0.429	0.138	+0.286	0.322
LA	−0.357	0.216	−0.643	$\underline{0.026}$	+0.143	0.621
PM	−0.317	0.087	−0.319	0.086	+0.202	0.279
RE	−0.167	0.368	−0.217	0.242	−0.109	0.558

not see a consistent pattern that certain methods prefer certain models. A special case, however, is the recall of *Retag*. We indeed observe the assumption of Reiss et al. (2020) that the model with the lowest recall often leads to the highest AED recall (see Figure 4). This is especially pronounced for token and span labeling. For these tasks, *Retag* can use a low-recall model to flag a large fraction of tokens because the model disagrees at many positions with the input labels. This improves recall while being detrimental to precision.

To summarize, overall we see positive correlation between model and AED performances. Using a well-performing model is a good choice for most model-based AED approaches. Neural models perform especially well, although they are more expensive to train. We therefore use transformers for text classification as well as span labeling and Flair for token labeling. Using a low-recall model for *Retag* leads to higher recall for token and span labeling, as conjectured by Reiss et al. (2020). This, however, concurs with lower precision and excessive flagging and thus more annotations need to be inspected.

5.4 RQ4 – What Performance Impact Does Using (or not Using) Cross-validation Have?

Model-based AED approaches are typically used together with CV (e.g., Amiri, Miller, and Savova 2018, Larson et al. 2020, Reiss et al. 2020). Northcutt, Jiang, and Chuang (2021) explicitly state that *Confident Learning* should only be applied to out-of-sample



(a) Text classification

(b) Token labeling

(c) Span labeling

Figure 4
Model-based methods and how often which model type leads to the best method performance with respect to overall, precision, and recall score. A connection from left to right between a method and a model indicates that using that method with outputs from that model leads to the best task performance. The color of the connection indicates the chosen model, for instance, Flair is 🔴, Transformer 🔵. The model axis is presented in descending order by model performance, aggregated by Borda Count across datasets. This figure is best viewed in color.

predicted probabilities. Amiri, Miller, and Savova (2018) do not mention that they used CV for *Dropout Uncertainty*, *Label Aggregation*, or *Classification Uncertainty*.

When using AED with CV, models are trained on $k-1$ splits and then detection is done on the remaining k-th set. After all unique folds are processed, all instances are checked. CV is often used in supervised learning where the goal is to find a model configuration as well as hyperparameters that generalize on unseen data. The goal of AED, however, is to find errors in the data at hand. Resulting models are just an instrument and not used afterwards. They therefore will not be applied to unseen data and need not generalize to data other than the one to clean. Hence, the question arises whether CV is really necessary for AED, which has not been analyzed as of yet. Not using CV has the advantage of being much faster and using less energy, since using

Table 7
Performance delta of model-based methods when training models with and without CV. Negative 🔴 values indicate that not using CV performs worse than using it, positive 🔵 values the opposite. It can be seen that overall recall is strongly impacted when not using CV but precision can improve. Flagger and scorer results are separated by a gap.

	Text			Token		Span			
Method	ATIS	IMDb	SST	GUM	Plank	Comp.	CoNLL	Flights	Forex
Δ Precision									
CL	+0.51	+0.77	−0.22	+0.16	+0.27	+0.02	+0.61	+0.08	+0.27
DE	+0.41	+0.11	+0.21	+0.26	+0.23	+0.34	−0.26	+0.59	+0.36
IRT	+0.02	+0.38	+0.03	+0.00	+0.04	+0.07	−0.27	+0.03	−0.41
LA	+0.26	+0.06	+0.28	−0.00	+0.23	+0.01	−0.05	+0.29	+0.32
PE	+0.05	+0.00	+0.01	+0.01	+0.04	+0.14	+0.01	+0.22	+0.11
RE	+0.29	+0.78	+0.19	+0.15	+0.17	−0.03	−0.10	+0.19	+0.29
CU	−0.03	−0.12	−0.20	−0.00	−0.05	−0.05	−0.13	−0.08	−0.29
DU	+0.34	−0.01	+0.12	+0.04	+0.14	+0.00	−0.03	+0.07	+0.14
PM	+0.34	−0.05	+0.05	+0.03	+0.14	−0.17	−0.02	+0.05	+0.07
Δ Recall									
CL	−0.04	−0.63	−0.82	−0.07	−0.25	−0.18	−0.17	−0.24	−0.46
DE	−0.29	−0.25	−0.41	−0.10	−0.34	−0.31	−0.30	−0.39	−0.37
IRT	+0.44	+0.06	+0.63	+0.09	+0.37	+0.26	−0.31	+0.76	−0.11
LA	−0.27	−0.64	−0.83	−0.00	−0.30	−0.18	−0.27	−0.34	−0.49
PE	+0.00	+0.00	−0.00	−0.00	−0.14	−0.13	−0.06	+0.00	−0.11
RE	−0.32	−0.64	−0.82	−0.00	−0.35	−0.17	−0.30	−0.44	−0.55
CU	−0.07	−0.62	−0.40	−0.00	−0.04	−0.01	−0.24	−0.22	−0.28
DU	+0.71	−0.05	+0.25	+0.09	+0.12	+0.00	−0.06	+0.17	+0.14
PM	+0.71	−0.23	+0.09	+0.06	+0.12	−0.04	−0.04	+0.12	+0.07
Δ % Flagged									
CL	−0.02	−0.06	−0.19	−0.02	−0.07	−0.09	−0.02	−0.01	−0.06
DE	−0.05	−0.05	−0.15	−0.03	−0.10	−0.31	−0.06	−0.06	−0.11
IRT	+0.06	−0.91	+0.16	+0.03	+0.13	+0.15	−0.06	+0.07	+0.79
LA	−0.03	−0.06	−0.19	+0.00	−0.10	−0.10	−0.05	−0.03	−0.09
PE	−0.01	−0.00	−0.01	−0.00	−0.04	−0.15	−0.03	−0.08	−0.05
RE	−0.04	−0.06	−0.19	−0.02	−0.10	−0.08	−0.05	−0.03	−0.09

CV increases training time linearly with the number of folds. In the typical setup with 10-fold CV, this means an increase of training time by 10×.

To answer this question we train a single model on all instances and then predict on the very same data. Then we use the resulting outputs to rerun methods that used CV before, which are *Classification Uncertainty*, *Confident Learning*, *Diverse Ensemble*, *Dropout Uncertainty*, *Item Response Theory*, *Label Aggregation*, *Prediction Margin*, *Projection Ensemble*, and *Retag*. The results are listed in Table 7. Overall, it can be seen that not using CV massively degrades recall for model-based methods while the precision improves. This can be intuitively explained by the fact that if the underlying models have already seen all the data, then they overfit to it and hence can re-predict it well. Due to the positive relationship between model and method performances (see § 5.3) this is also reflected downstream; fewer instances are predicted differently than the original labels. This reduces recall and thereby the chance of making errors, thus increasing precision. This can be seen by the reduction in the percentage of flagged instances for flaggers. Interestingly, *Dropout Uncertainty* and *Prediction Margin* are not impacted as much and sometimes even improve when not using CV across all scores, especially for easier datasets. Recall of *Item Response Theory* also improves at the cost of more flagged items and a reduction in precision. *Prediction Ensemble* for text classification is relatively unaffected and for token and span labeling, the performance difference is around ±0.10 pp. Therefore, it might be a good tradeoff to not use CV with this method as it is already expensive due to its ensembling.

To summarize, not using CV can negatively impact performance—in particular, degrading recall. We therefore recommend the use of CV, even though it increases runtime by the number of folds (in our case, by a factor of ten). In settings where this is an issue, we recommend using methods that inherently do not need CV. These include most heuristics and well-performing approaches like *Datamap Confidence*, *Leitner Spotter*, or *Curriculum Spotter*. If precision is more important than recall, then not using CV might be taken into consideration.

6. Takeaways and Recommendations

This article has probed several questions related to annotation error detection. Our findings show that it is usually better to use well-performing models for model-based methods as they yield better detection performance on average. Using a worse model for *Retag* improves recall at the cost of lower precision. For detection, these models should be trained via cross-validation, otherwise the recall of downstream methods is heavily degraded (while the precision improves). Calibration can improve these model-based annotation error detection methods, but more research is needed to determine when exactly it can be useful. Some model-method combinations achieved relatively large gains after calibration while others did not improve.

Methods that are used frequently in practice—*Retag* and *Classification Uncertainty*—performed well in our experiments. Others did not perform particularly well, especially *Dropout Uncertainty*, *Item Response Theory*, *k-Nearest Neighbor Entropy*, *Mean Distance*, and *Prediction Margin*. For *Mean Distance* in particular, Larson et al. (2019) reported AP of > 0.6 and recall > 0.8 on corpora with artificial noise, which we could not reproduce. Experiments with *Dropout Uncertainty* disseminated in Amiri, Miller, and Savova (2018) reached similar high scores as using *Curriculum Spotter*, *Leitner Spotter*, or *Classification Uncertainty*, but we were not able to make *Dropout Uncertainty* reach similar high scores as the others. *Label Aggregation*, though, which uses the same inputs, performs exceedingly well. For the others, either no scores were reported or they were similarly low as in our experiments.

Experiments on actual corpora have shown that AED methods still have room for improvement. While looking promising on artificial corpora, there is a large performance drop when applying them in practice. Overall, the methods that worked best are *Classification Uncertainty*, *Confident Learning*, *Curriculum Spotter*, *Datamap Confidence*, *Diverse Ensemble*, *Label Aggregation*, *Leitner Spotter*, *Projection Ensemble*, and *Retag*. More complicated methods are not necessarily better. For instance, *Classification Uncertainty* and *Retag* perform well across tasks and datasets while being easy to implement. Model-based methods require *k*-fold cross-validation. Therefore, if runtime is a concern, then *Datamap Confidence* is a good alternative. It performs well while only needing to train one model instead of *k*. In case the data or its corresponding task to correct is not suitable for machine learning, methods like *Label Entropy*, *K-Nearest-Neighbor Entropy*, or *Variation n-grams* still can be applied. As the latter usually has high precision it is often worthwhile to apply it whenever the data is suitable for it; that is, if the data has sufficient surface form overlap. Individual scorer scores can be aggregated via *Borda Count* but it tremendously increases runtime. While not yielding significantly better results in our experiments, results aggregated that way were much more stable across datasets and tasks while individual scorers sometimes had performance drops in certain settings.

Manual analysis of CONLL-2003 showed that finding inconsistencies is often more difficult than finding annotation errors. While model-based methods were often quite good in the latter, they performed poorly when detecting inconsistencies. Methods that do not rely on the noisy labels but on the surface form or semantics like *k-Nearest Neighbor Entropy*, *Label Entropy*, and *Weighted Discrepancy* have shown the opposite behavior. They each have their own strengths and it can be worth combining both types of methods.

7. Conclusion

Having annotated corpora with high-quality labels is imperative for many branches of science and for the training of well-performing and generalizing models. Previous work has shown that even commonly used benchmark corpora contain non-negligible numbers of annotation errors. In order to assist human annotators with detecting and correcting these errors, many different methods for annotation error detection have been developed. To date, however, methods have not been compared, so it has been unclear what method to choose under what circumstances. To close this gap, we surveyed the field of annotation error detection, reimplemented 18 methods, collected and generated 9 datasets for text classification, token labeling, and span labeling, and evaluated method performance in different settings. Our results show that AED can already be useful in real use cases to support data cleaning efforts. But especially for more difficult datasets, the performance ceiling is far from reached yet.

In the past, the focus of most works researching or using AED was to clean data and not to develop a method. The method was only a means to achieve a cleaned corpus and not the target itself. Also, several studies proposed algorithms for different use cases and AED was one application to it just mentioned briefly at the end without in-depth evaluation, rendering it unclear how well the method performs. We therefore strongly encourage authors who introduce new AED methods to compare their method to previous work and on the same corpora to foster reproducibility and to bring the performance of new methods into context. This article surveys, standardizes, and answers several fundamental questions regarding AED so that future work has a stable footing for research. For this, we also make our implementation and datasets publicly available.

Limitations and Future Work. While we thoroughly investigated many available methods on different datasets and tasks, there are some limitations to our work. First, the datasets that we used were only in English. Therefore, it would be interesting to investigate AED on different languages. One first step could be the work by Hedderich, Zhu, and Klakow (2021), who created a corpus for NER in Estonian with natural noise patterns. Hand-curated datasets with explicitly annotated errors are rare. We therefore also used existing, clean datasets and injected random noise, similarly to previous works. These datasets with artificial errors have been shown to overestimate the ability of AED methods, but are still a good estimator for the maximal performance of methods. The next step is to create benchmark corpora that are designed from the ground up for the evaluation of annotation error detection. As creating these requires effort and is costly, a cheaper way is to aggregate raw crowdsourcing data. This is often not published along with adjudicated corpora, so we urge researchers to also publish these alongside the final corpus.

AED was evaluated on three different tasks with nine NLP datasets. The tasks were chosen based on the number of datasets and model types available to answer our research questions. Most AED methods are task-agnostic; previous work, for instance, investigated question answering (Amiri, Miller, and Savova 2018) or relation classification (Alt, Gabryszak, and Hennig 2020; Stoica, Platanios, and Poczos 2021). Hence, AED can and has been applied in different fields like computer vision (Northcutt, Athalye, and Mueller 2021). But these works are plagued by the same issues that most previous AED works have (e.g., only limited comparison to other works and quantitative analysis, code and data not available). Having several fundamental questions answered in this article, future work can now readily apply and investigate AED on many different tasks, domains, and in many different settings, while leveraging our findings (which are summarized in § 6). It would especially be interesting to evaluate and apply AED on more hierarchical and difficult tasks, such as semantic role labeling or natural language inference.

While we investigated many relevant research questions, these were mostly about model-based methods as well as flaggers. To date, scorers have been treated as a black box, so it would be worth investigating what makes a good scorer—for example, what makes *Classification Uncertainty* better than *Prediction Margin*. Also, leveraging scorers as uncertainty estimates for correction is a promising application, similar to the works of Dligach and Palmer (2011) or Angle, Mishra, and Sharma (2018).

This work also only focuses on errors, inconsistencies, and ambiguities related to instance labels. Some datasets also benefit from finding errors concerning tokenization, sentence splitting, or missing entities (e.g., Reiss et al. 2020). We also did not investigate the specific kinds of errors made. This can be useful information and could be leveraged by human annotators during manual corrections. It would be especially interesting to investigate the kinds of errors certain models and configurations were able to correct—for instance, whether using no cross-validation finds more obvious errors but with a higher precision. We leave detection of errors other than incorrect labels or error kind detection for future work because we did not find a generic way to do it across the wide range of evaluated datasets and tasks used in this article.

Finally, we implemented each method as described and performed only basic hyperparameter tuning. We did not tune them further due to the prohibitive costs for our large-scale setup. This is especially true for the considered machine learning models, where we kept the parameters mostly default for all regardless of the dataset and domain. We are sure that one can certainly improve scores for each method, but our implementations should still serve as a lower bound. However, we do not expect large gains from further optimization and no large shifts in ranking between the methods.

Appendix A. Hyperparameter Tuning and Implementation Details

In this section we briefly describe implementation details for the different AED methods used throughout this article. As the tuning data we select one corpus for each task type. For text classification we subsample 5,000 instances from the training split of AG NEWS (Zhang, Zhao, and LeCun 2015); the number of samples is chosen as it is around the same data size as our other datasets. As the corpus for token labeling we choose the English part of PARTUT (Sanguinetti and Bosco 2015) and their POS annotations and inject 5% random noise. For span labeling we use CONLL-2003 (Reiss et al. 2020) to which we apply 5% flipped label noise.

A.1 Aggregating Probabilities and Embeddings for Span Labeling

When converting BIO-tagged sequences to spans for alignment (see § 4.3) consisting only of start and end position as well as its label, the probabilities assigned to each BIO-tag representing the span need to be aggregated. The same needs to be done for creating span embeddings from token embeddings. As an example, consider NER for persons and locations with a tagset of `B-PER`, `I-PER`, `B-LOC`, `I-LOC`. It has to be aggregated so that spans have labels `PER` or `LOC`. Look at a span of two tokens that has been tagged as `B-PER, I-PER`. Then the probability for `PER` needs to be aggregated from the `B-PER` and `I-PER` tags. We evaluate our CONLL-2003 tuning data. We use a Maxent sequence tagger to evaluate *Confident Learning* with 10-fold cross-validation for this hyperparameter selection. In addition, for *k-Nearest-Neighbor Entropy* we evaluate aggregation schemes to create span embeddings from individual token embeddings. Overall, we do not observe a large difference between max, mean, or median aggregation. The results can be seen in Table A.1. We choose aggregating via arithmetic mean because it is slightly better in terms of F1 and AP than the other methods.

A.2 Method Details

In the following we describe the choices we made when implementing the various AED methods evaluated in this article.

Diverse Ensemble Our diverse ensemble uses the predictions of all different model types trained for the task and dataset, similarly to Loftsson (2009), Alt, Gabryszak, and Hennig (2020), and Barnes, Øvrelid, and Velldal (2019).

Spotter and Datamap Confidence The implementations of *Datamap Confidence* as well as *Curriculum Spotter* and *Leitner Spotter* require callbacks or a similar functionality to obtain predictions for every epoch which only HuggingFace Transformers provide. That is why we only evaluate these methods in combination with a transformer.

Dropout Uncertainty In our implementation we use mean entropy, which we observed in preliminary experiments to perform slightly better overall than the other version evaluated by Shelmanov et al. (2021).

Variation *n*-grams We follow Wisniewski (2018) for our implementation and use generalized suffix trees to find repetitions. If there are repetitions of length more than one in the surface forms that are tagged differently, we look up the respective tag sequence that occurs most often in the corpus and flag the positions of all other repetitions where

Table A.1
Impact of different aggregation functions for span alignment and embeddings.

	CL			KNN		
Aggregation	P	R	F1	AP	P@10%	R@10%
min	0.149	0.594	0.238	0.307	0.325	0.326
max	0.761	0.881	0.817	0.308	0.325	0.326
mean	0.766	0.878	0.818	0.318	0.331	0.333
median	0.765	0.876	0.817	0.316	0.330	0.332

they disagree with the majority tags. We convert tokens and sentences to lower case to slightly increase recall while slightly reducing precision. We do not flag an instance if its label is the most common label. This yields far better results as the most common label is most often correct and should not be flagged. When using *Variation n-grams* for span labeling, we use a context of one token to the left and right of the span, similarly to Larson et al. (2020).

Projection Ensemble In our implementation we flag an instance if the majority label of the ensemble disagrees with the given one.

Label Aggregation In the original work that evaluated using *Label Aggregation* for AED (Amiri, Miller, and Savova 2018), MACE (Hovy et al. 2013) was used. We use Dawid-Skene (Dawid and Skene 1979), which has similar performance as MACE (Paun et al. 2018) but many more available implementations (we use Ustalov et al. (2021)). The difference between the two (MACE modeling annotator spam) is not relevant here.

Mean Distance We compare different embedding methods and metrics for *Mean Distance*. For that we use the Sentence Transformers[6] library and evaluate S-BERT embeddings (Reimers and Gurevych 2019), Universal Sentence Encoder (Cer et al. 2018), and average GloVe embeddings (Pennington, Socher, and Manning 2014). We evaluate on our AG NEWS tuning data. As our Universal Sentence Encoder implementation we use `distiluse-base-multilingual-cased-v1` from Sentence Transformers. The Universal Sentence Encoder embeddings as used in the original implementation of *Mean Distance* (Larson et al. 2019) overall perform not better than all S-BERT embeddings. `lof` refers to Local Outlier Factor, a clustering metric proposed by Breunig et al. (2000). Using *all-mpnet-base-v2* together with Euclidean distance works best here and we use this throughout our experiments.

Item Response Theory We use the setup from Rodriguez et al. (2021), that is, a 2P IRT model that is optimized via variational inference and the original code of the authors. We optimize for 10,000 iterations. *Item Response Theory* uses the collected predictions of all models for the respective task, similarly to *Diverse Ensemble*.

Label Entropy and Weighted Discrepancy We implement equations (2) and (3) in Hollenstein, Schneider, and Webber (2016) but assign the minimum score (meaning no error) if the current label is the most common label. This yields far better results because the most common label is most often correct and should not be downranked.

K-Nearest-Neighbor Entropy To evaluate which embedding aggregation over transformer layers works best for *k-Nearest-Neighbor Entropy*, we evaluate several different configurations on our PARTUT tuning data. We chose this task and not span labeling as span labeling requires an additional aggregation step to combine token embeddings to span embeddings (see § 1.1). The transformer of choice is RoBERTa (Liu et al. 2019), as it has better performance than BERT while still being fast enough. We also compare with several non-transformer embeddings: GloVe 6B (Pennington, Socher, and Manning 2014), Byte-Pair Encoding (Heinzerling and Strube 2018), and a concatenation of both. We follow Devlin et al. (2019) regarding which configurations to try. The results can be seen in Table A.2. The best scoring embedder is RoBERTa, using only the last layer that will be the configuration used throughout this work for obtaining token and span

[6] `https://www.sbert.net/`.

Table A.2
The performance impact of using different embedding types and configurations for KNN entropy on UD ParTUT.

Embedder	AP	P@10%	R@10%
Last Hidden	0.265	0.255	0.256
Sum All Layers	0.237	0.254	0.254
First Hidden	0.230	0.269	0.270
Sum Last 4 Hidden	0.227	0.246	0.246
Second-to-Last Hidden	0.224	0.244	0.244
Concat Last 4 Hidden	0.149	0.193	0.194
Glove	0.148	0.169	0.169
Glove + BPE	0.148	0.175	0.175
BPE	0.147	0.170	0.170

Table A.3
Evaluation of scorers and their aggregation via Borda Count for text classification and span labeling. Highlighted in gray are the runs of Borda Count aggregation.

Method	AP	P@10%	R@10%
BC_{top3}	0.848	0.460	0.947
BC_{top2}	0.824	0.456	0.938
DM	0.819	0.448	0.922
BC_{top5}	0.794	0.454	0.934
LS	0.706	0.448	0.922
CU	0.521	0.426	0.877
MD	0.422	0.344	0.708
CS	0.296	0.390	0.802
BC_{all}	0.268	0.244	0.502
KNN	0.055	0.062	0.128
DU	0.055	0.062	0.128
PM	0.050	0.046	0.095

(a) Text

Method	AP	P@10%	R@10%
DM	0.963	0.932	0.934
BC_{top3}	0.897	0.863	0.865
CU	0.881	0.837	0.839
BC_{top2}	0.881	0.837	0.839
BC_{top5}	0.716	0.625	0.626
WD	0.665	0.632	0.633
LE	0.567	0.579	0.580
BC_{all}	0.350	0.378	0.379
MD	0.206	0.231	0.232
DU	0.103	0.104	0.104
PM	0.102	0.104	0.104
KNN	0.100	0.101	0.101

(b) Span

embeddings. For sentence embeddings we will use *all-mpnet-base-v2* (see § 1.2). To compute the KNN entropy, we use the code of the original authors.

Borda Count In order to evaluate which scores to aggregate via *Borda Count* we evaluate three settings on our AG NEWS tuning data. We either aggregate the five best, three best, or all scorer outputs. As the underlying model for model-based methods we use transformers, because we need repeated probabilities for *Dropout Uncertainty*. The results are listed in Table A.3. It can be seen that aggregating only the three best scores leads to far superior performance. Hence, we choose this setting when evaluating *Borda Count* aggregation during our experiments.

Appendix B. Calibration

From the most common calibration methods we select the best method by calibrating probabilities for models trained on our AG NEWS tuning data. We use 10-fold cross-validation where eight parts are used for training models, one for calibrating and one for evaluating the calibration. The results can be seen in Figure B.1. We follow Guo et al.

(2017) and use the Expected Calibration Error (ECE) (Naeini, Cooper, and Hauskrecht 2015) as the metric for calibration quality. We decide to use one method for all task types and finally choose *Logistic Calibration* (also known as Platt Scaling), which performs well across tasks. We use the implementations of Küppers et al. (2020).

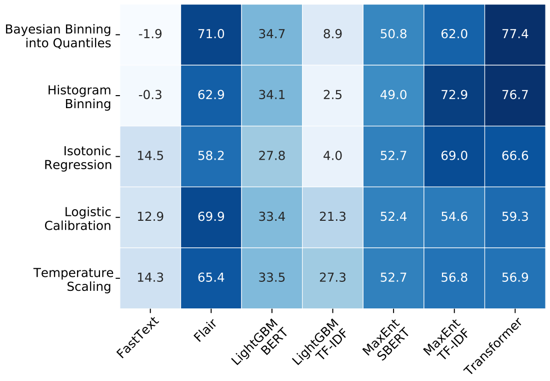


Figure B.1
Percentage decrease of the Expected Calibration Error (ECE) after calibration, when training models on our AG NEWS tuning data. Higher is better.

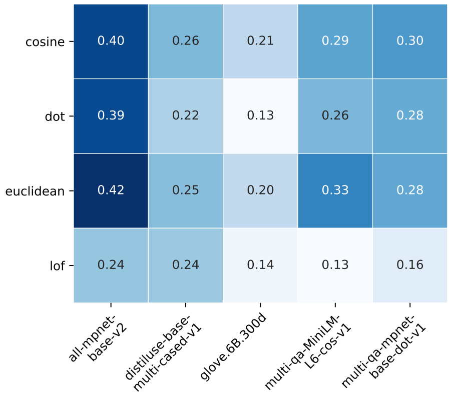
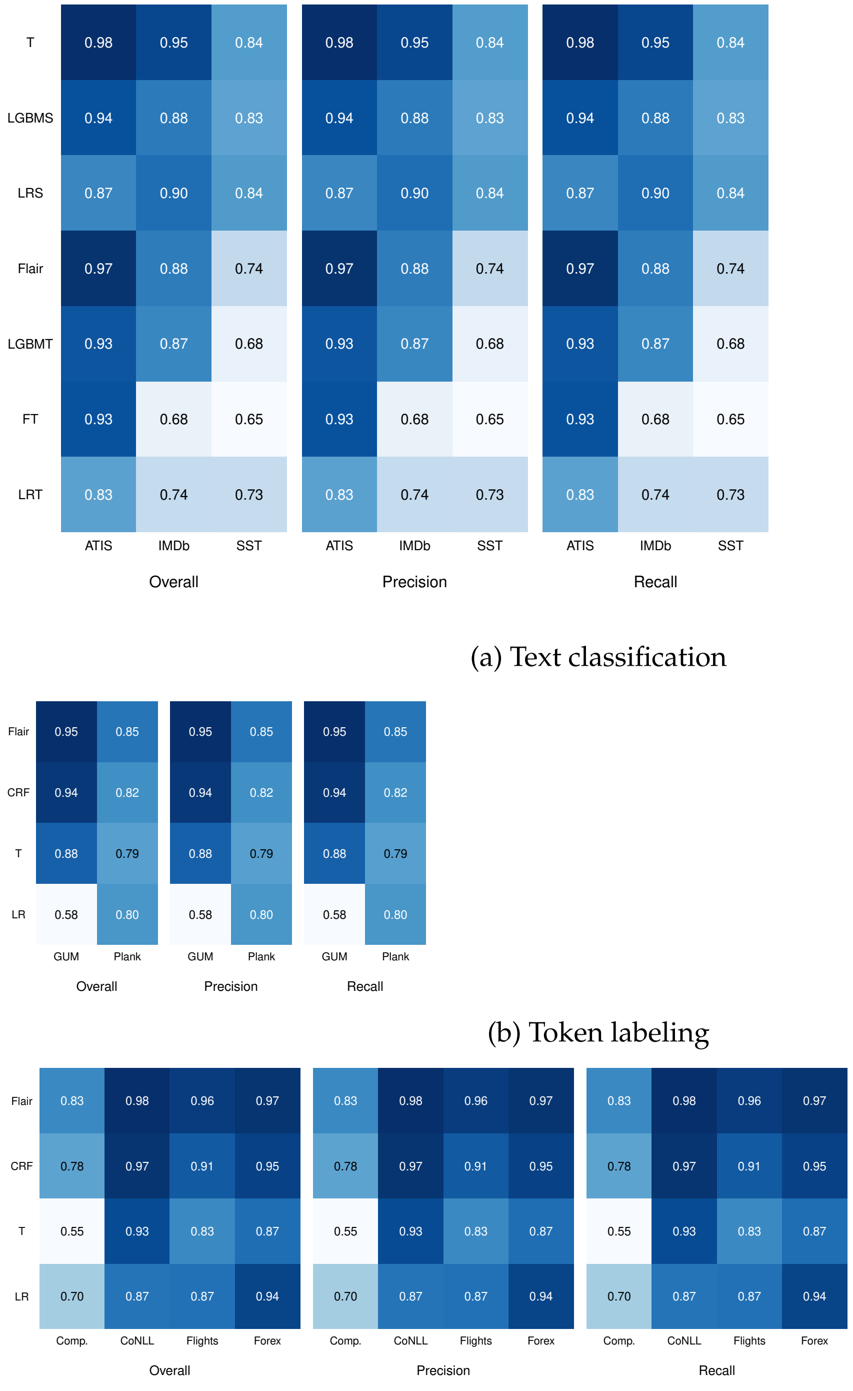


Figure B.2
Average Precision of using *Mean Distance* with different embedders and similarity metrics on our AG NEWS tuning data.

Appendix C. Best Scores



(a) Text classification

(b) Token labeling

(c) Sequence labeling

Figure C.1
Model performances across tasks and datasets. The model axis is ordered in descending order by the respective models' overall performance via Borda Count.

C	M	P	R	F1	%F
ATIS	CL	0.43	0.30	0.35	0.03
	DE	0.56	1.00	0.72	0.09
	IRT	0.00	0.00	0.00	0.91
	LA	0.71	1.00	0.83	0.07
	PE	0.37	1.00	0.54	0.13
	RE	0.67	1.00	0.81	0.07
IMDb	CL	0.23	0.63	0.33	0.06
	DE	0.19	0.73	0.30	0.08
	IRT	0.01	0.30	0.01	0.93
	LA	0.22	0.64	0.33	0.06
	PE	0.10	0.62	0.18	0.12
	RE	0.22	0.64	0.33	0.06
SST	CL	0.22	0.82	0.34	0.19
	DE	0.21	0.82	0.33	0.20
	IRT	0.01	0.16	0.02	0.82
	LA	0.22	0.84	0.35	0.19
	PE	0.22	0.84	0.34	0.19
	RE	0.21	0.82	0.34	0.19

C	M	AP	P@10%	R@10%
ATIS	BC	0.98	0.48	1.00
	CS	0.97	0.48	1.00
	CU	0.87	0.48	1.00
	DM	0.98	0.48	1.00
	DU	0.05	0.06	0.13
	KNN	0.13	0.13	0.27
	LS	0.91	0.48	1.00
	MD	0.14	0.17	0.35
	PM	0.06	0.06	0.13
IMDb	BC	0.35	0.14	0.71
	CS	0.29	0.13	0.64
	CU	0.28	0.15	0.74
	DM	0.25	0.13	0.63
	DU	0.06	0.08	0.39
	KNN	0.05	0.05	0.27
	LS	0.31	0.13	0.67
	MD	0.03	0.04	0.18
	PM	0.05	0.07	0.35
SST	BC	0.50	0.37	0.74
	CS	0.21	0.27	0.54
	CU	0.27	0.29	0.59
	DM	0.49	0.33	0.67
	DU	0.05	0.04	0.09
	KNN	0.11	0.11	0.22
	LS	0.46	0.33	0.65
	MD	0.08	0.10	0.20
	PM	0.05	0.05	0.10

(a) Text classification

C	M	P	R	F1	%F
Comp.	CL	0.83	0.35	0.50	0.18
	DE	0.57	0.58	0.57	0.43
	IRT	0.32	0.59	0.41	0.79
	LA	0.78	0.48	0.59	0.26
	PE	0.59	0.53	0.56	0.38
	RE	0.80	0.53	0.64	0.28
	VN	0.69	0.06	0.11	0.04
CoNLL	CL	0.39	0.18	0.24	0.02
	DE	0.26	0.30	0.28	0.06
	IRT	0.27	0.31	0.29	0.06
	LA	0.29	0.31	0.30	0.06
	PE	0.17	0.47	0.25	0.15
	RE	0.30	0.33	0.32	0.06
	VN	0.60	0.01	0.02	0.00
Flights	CL	0.92	0.27	0.42	0.01
	DE	0.41	0.80	0.55	0.07
	IRT	0.01	0.20	0.02	0.93
	LA	0.60	0.73	0.66	0.05
	PE	0.18	0.68	0.29	0.14
	RE	0.61	0.73	0.67	0.05
	VN	1.00	0.17	0.29	0.01
Forex	CL	0.73	0.46	0.57	0.06
	DE	0.52	0.81	0.64	0.16
	IRT	0.49	0.85	0.62	0.18
	LA	0.65	0.76	0.70	0.12
	PE	0.45	0.72	0.56	0.16
	RE	0.67	0.75	0.70	0.11
	VN	1.00	0.07	0.14	0.01

C	M	AP	P@10%	R@10%
Comp.	BC	0.68	0.83	0.20
	CU	0.70	0.87	0.21
	DM	0.66	0.82	0.19
	DU	0.43	0.50	0.12
	KNN	0.61	0.75	0.18
	LE	0.41	0.38	0.09
	MD	0.54	0.59	0.14
	PM	0.54	0.64	0.15
	WD	0.45	0.48	0.11
CoNLL	BC	0.14	0.13	0.24
	CU	0.17	0.18	0.34
	DM	0.14	0.12	0.23
	DU	0.07	0.08	0.15
	KNN	0.12	0.13	0.24
	LE	0.19	0.17	0.32
	MD	0.06	0.05	0.09
	PM	0.06	0.07	0.14
	WD	0.16	0.17	0.32
Flights	BC	0.49	0.24	0.63
	CU	0.68	0.29	0.76
	DM	0.35	0.25	0.66
	DU	0.18	0.10	0.27
	KNN	0.07	0.07	0.20
	LE	0.10	0.10	0.27
	MD	0.07	0.11	0.29
	PM	0.12	0.12	0.32
	WD	0.11	0.12	0.32
Forex	BC	0.54	0.49	0.49
	CU	0.70	0.71	0.71
	DM	0.61	0.66	0.65
	DU	0.32	0.32	0.32
	KNN	0.16	0.14	0.14
	LE	0.11	0.09	0.09
	MD	0.14	0.20	0.20
	PM	0.25	0.30	0.30
	WD	0.14	0.20	0.20

(b) Span labeling

C	M	P	R	F1	%F
GUM	CL	0.73	0.90	0.80	0.06
	DE	0.59	1.00	0.74	0.08
	IRT	0.00	0.00	0.00	0.91
	LA	0.51	1.00	0.68	0.10
	PE	0.41	1.00	0.58	0.12
	RE	0.53	1.00	0.69	0.09
	VN	0.47	0.66	0.55	0.07
Plank	CL	0.47	0.31	0.37	0.08
	DE	0.45	0.51	0.48	0.13
	IRT	0.07	0.47	0.12	0.84
	LA	0.46	0.53	0.49	0.14
	PE	0.48	0.53	0.50	0.13
	RE	0.47	0.52	0.49	0.13
	VN	0.55	0.21	0.30	0.04

C	M	AP	P@10%	R@10%
GUM	BC	0.92	0.47	0.95
	CU	0.98	0.50	1.00
	DM	0.95	0.49	0.98
	DU	0.05	0.05	0.10
	KNN	0.21	0.19	0.38
	LE	0.60	0.34	0.69
	MD	0.12	0.14	0.29
	PM	0.05	0.05	0.11
	WD	0.53	0.39	0.79
Plank	BC	0.38	0.42	0.36
	CU	0.42	0.51	0.43
	DM	0.27	0.37	0.31
	DU	0.24	0.28	0.24
	KNN	0.31	0.39	0.33
	LE	0.22	0.24	0.21
	MD	0.16	0.19	0.16
	PM	0.23	0.29	0.24
	WD	0.39	0.43	0.37

(c) Token labeling

Figure C.2
AED results achieved with using the respective best models across all flaggers 🟠 and scorers 🟣 for text classification, span, and token labeling.

Acknowledgments

We thank Andreas Rücklé, Edwin Simpson, Falko Helm, Ivan Habernal, Ji-Ung Lee, Michael Bugert, Nafise Sadat Moosavi, Nils Dycke, Richard Eckart de Castilho, Tobias Mayer, and Yevgeniy Puzikov, and our anonymous reviewers for the fruitful discussions and helpful feedback that improved this article. We are especially grateful for Michael Bugert and his implementation of the span matching via linear sum assignment and Amir Zeldes, Andreas Grivas, Beatrice Alex, Hadi Amiri, Jeremy Barnes, as well as Stefan Larson for answering our questions regarding their publications and for making datasets and code available. This research work has been funded by the "Data Analytics for the Humanities" grant by the Hessian Ministry of Higher Education, Research, Science and the Arts.

References

Ahrens, Sönke. 2017. *How to Take Smart Notes: One Simple Technique to Boost Writing, Learning and Thinking: For Students, Academics and Nonfiction Book Writers*. CreateSpace, North Charleston, SC.

Akbik, Alan, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 54–59.

Alt, Christoph, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569. `https://doi.org/10.18653/v1/2020.acl-main.142`

Ambati, Bharat Ram, Rahul Agarwal, Mridul Gupta, Samar Husain, and Dipti Misra Sharma. 2011. Error detection for treebank validation. In *Proceedings of the 9th Workshop on Asian Language Resources*, pages 23–30.

Amiri, Hadi, Timothy Miller, and Guergana Savova. 2018. Spotting spurious data with neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2006–2016. `https://doi.org/10.18653/v1/N18-1182`

Angle, Sachi, Pruthwik Mishra, and Dipti Mishra Sharma. 2018. Automated error correction and validation for POS tagging of Hindi. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, pages 11–18.

Aroyo, Lora and Chris Welty. 2015. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24. `https://doi.org/10.1609/aimag.v36i1.2564`

Barnes, Jeremy, Lilja Øvrelid, and Erik Velldal. 2019. Sentiment analysis is not solved! Assessing and probing sentiment classification. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 12–23. `https://doi.org/10.18653/v1/W19-4802`

Basile, Valerio, Michael Fell, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, Massimo Poesio, and Alexandra Uma. 2021. We need to consider disagreement in evaluation. In *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, pages 15–21. `https://doi.org/10.18653/v1/2021.bppf-1.3`

Behrens, Heike, editor. 2008. *Corpora in Language Acquisition Research: History, Methods, Perspectives*, volume 6 of *Trends in Language Acquisition Research*. John Benjamins Publishing Company, Amsterdam. `https://doi.org/10.1075/tilar.6.03beh`

Boyd, Adriane, Markus Dickinson, and W. Detmar Meurers. 2008. On detecting errors in dependency treebanks. *Research on Language and Computation*, 6(2):113–137. `https://doi.org/10.1007/s11168-008-9051-9`

Breunig, Markus M., Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104. `https://doi.org/10.1145/335191.335388`

Burkard, Rainer, Mauro Dell'Amico, and Silvano Martello. 2012. *Assignment Problems: Revised Reprint*. Society for Industrial and Applied Mathematics. `https://doi.org/10.1137/1.9781611972238`

Cer, Daniel, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in*

Natural Language Processing: System Demonstrations, pages 169–174. https://doi.org/10.18653/v1/D18-2029

Chinchor, Nancy and Beth Sundheim. 1993. MUC-5 evaluation metrics. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 69–78. https://doi.org/10.3115/1072017.1072026

Cui Zhu, H. Kitagawa, and C. Faloutsos. 2005. Example-based robust outlier detection in high dimensional datasets. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 829–832.

Davis, Jesse and Mark Goadrich. 2006. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, pages 233–240.

Dawid, A. P. and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28. https://doi.org/10.2307/2346806

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Dickinson, Markus. 2006. From detecting errors to automatically correcting them. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 265–272.

Dickinson, Markus. 2015. Detection of annotation errors in Corpora. *Language and Linguistics Compass*, 9(3):119–138. https://doi.org/10.1111/lnc3.12129

Dickinson, Markus and Chong Min Lee. 2008. Detecting errors in semantic annotation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 605–610.

Dickinson, Markus and W. Detmar Meurers. 2003a. Detecting errors in part-of-speech annotation. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 107–114. https://doi.org/10.3115/1067807.1067823

Dickinson, Markus and W. Detmar Meurers. 2003b. Detecting inconsistencies in treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, pages 1–12.

Dickinson, Markus and W. Detmar Meurers. 2005. Detecting errors in discontinuous structural annotation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics - ACL '05*, pages 322–329. https://doi.org/10.3115/1219840.1219880

Dligach, Dmitriy and Martha Palmer. 2011. Reducing the need for double annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 65–73.

Dwork, Cynthia, Ravi Kumar, Moni Naor, and D. Sivakumar. 2001. Rank aggregation methods for the Web. In *Proceedings of the Tenth International Conference on World Wide Web - WWW '01*, pages 613–622.

Fornaciari, Tommaso, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597. https://doi.org/10.18653/v1/2021.naacl-main.204

Gal, Yarin and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1050–1059.

Gimpel, Kevin, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47. https://doi.org/10.21236/ADA547371

Grivas, Andreas, Beatrice Alex, Claire Grover, Richard Tobin, and William Whiteley. 2020. Not a cute stroke: Analysis of Rule- and Neural Network-based Information Extraction Systems for Brain Radiology Reports. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 24–37. https://doi.org/10.18653/v1/2020.louhi-1.4

Guo, Chuan, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of* *the 34th International Conference on Machine Learning*, pages 1321–1330.

Gururangan, Suchin, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360. `https://doi.org/10.18653/v1/2020.acl-main.740`

Haselbach, Boris, Kerstin Eckart, Wolfgang Seeker, Kurt Eberle, and Ulrich Heid. 2012. Approximating theoretical linguistics classification in real data: The case of German "nach" particle verbs. In *Proceedings of COLING 2012*, pages 1113–1128.

Hedderich, Michael A., Dawei Zhu, and Dietrich Klakow. 2021. Analysing the noise model error for realistic noisy label data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7675–7684. `https://doi.org/10.1609/aaai.v35i9.16938`

Heinzerling, Benjamin and Michael Strube. 2018. BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 2989–2993.

Hemphill, Charles T., John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Proceedings of the Workshop on Speech and Natural Language*, pages 96–101. `https://doi.org/10.3115/116580.116613`

Hendrycks, Dan and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of International Conference on Learning Representations*, pages 1–12.

Hollenstein, Nora, Nathan Schneider, and Bonnie Webber. 2016. Inconsistency detection in semantic annotation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3986–3990.

Hovy, Dirk, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130.

Jamison, Emily and Iryna Gurevych. 2015. Noise or additional information? Leveraging crowdsource annotation item agreement for natural language tasks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 291–297. `https://doi.org/10.18653/v1/D15-1035`

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. `https://doi.org/10.18653/v1/E17-2068`

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1–9.

Kehler, A., L. Kertz, H. Rohde, and J. L. Elman. 2007. Coherence and coreference revisited. *Journal of Semantics*, 25(1):1–44. `https://doi.org/10.1093/jos/ffm018`, PubMed: 22923856

Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika*, 30(1–2):81–93. `https://doi.org/10.1093/biomet/30.1-2.81`

Khandelwal, Urvashi, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations (ICLR)*, pages 1–13.

Küppers, Fabian, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. 2020. Multivariate confidence calibration for object detection. In *2nd Workshop on Safe Artificial Intelligence for Automated Driving (SAIAD)*, pages 1–9.

Květoň, Pavel and Karel Oliva. 2002. (Semi-)automatic detection of errors in PoS-tagged corpora. In *COLING 2002: The 19th International Conference on Computational Linguistics*, pages 1–7. `https://doi.org/10.3115/1072228.1072249`

Larson, Stefan, Adrian Cheung, Anish Mahendran, Kevin Leach, and Jonathan K. Kummerfeld. 2020. Inconsistencies in crowdsourced slot-filling annotations: A typology and identification methods. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5035–5046. `https://doi.org/10.18653/v1/2020.coling-main.442`

Larson, Stefan, Anish Mahendran, Andrew Lee, Jonathan K. Kummerfeld, Parker Hill, Michael A. Laurenzano, Johann Hauswald, Lingjia Tang, and Jason Mars. 2019. Outlier detection for improved data quality and diversity in dialog systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 517–527. `https://doi.org/10.18653/v1/N19-1051`

Leitner, Sebastian. 1974. *So Lernt Man Leben [How to Learn to Live]*. Droemer-Knaur, Munich.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. arxiv preprints 11692.

Loftsson, Hrafn. 2009. Correcting a POS-Tagged corpus using three complementary methods. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 523–531. `https://doi.org/10.3115/1609067.1609125`

Lord, F. M., M. R. Novick, and Allan Birnbaum. 1968. *Statistical Theories of Mental Test Scores*. Addison-Wesley, Oxford, England.

Manning, Christopher D. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing*, volume 6608, pages 171–189. `https://doi.org/10.1007/978-3-642-19400-9_14`

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York.

Ménard, Pierre André and Antoine Mougeot. 2019. Turning silver into gold: Error-focused corpus reannotation with active learning. In *Proceedings - Natural Language Processing in a Deep Learning World*, pages 758–767. `https://doi.org/10.26615/978-954-452-056-4_088`

Naeini, Mahdi Pakdaman, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2901–2907.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043.

Northcutt, Curtis, Lu Jiang, and Isaac Chuang. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411. `https://doi.org/10.1613/jair.1.12125`

Northcutt, Curtis G., Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. In *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, pages 1–13.

Paun, Silviu, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian models of annotation. *Transactions of the Association for Computational Linguistics*, 6(0):571–585. `https://doi.org/10.1162/tacl_a_00040`

Pavlick, Ellie and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. `https://doi.org/10.3115/v1/D14-1162`

Peters, Matthew E., Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14. `https://doi.org/10.18653/v1/W19-4302`

Plank, Barbara, Dirk Hovy, and Anders Søgaard. 2014a. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 742–751.

Plank, Barbara, Dirk Hovy, and Anders Søgaard. 2014b. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511. `https://doi.org/10.3115/v1/P14-2083`

Platt, John C. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods.

Advances in Large Margin Classifiers, 10(3):1–9.

Pustejovsky, J. and Amber Stubbs. 2013. *Natural Language Annotation for Machine Learning*. O'Reilly Media, Sebastopol, CA.

Qian, Kun, Ahmad Beirami, Zhouhan Lin, Ankita De, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. 2021. Annotation inconsistency and entity bias in MultiWOZ. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 326–337.

Rehbein, Ines. 2014. POS error detection in automatically annotated corpora. In *Proceedings of LAW VIII - the 8th Linguistic Annotation Workshop*, pages 20–28.

Reimers, Nils and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990. `https://doi.org/10.18653/v1/D19-1410`

Reiss, Frederick, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger. 2020. Identifying incorrect labels in the CoNLL-2003 corpus. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 215–226. `https://doi.org/10.18653/v1/2020.conll-1.16`

Rodrigues, Filipe and Francisco Pereira. 2018. Deep learning from crowds. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1611–1618.

Rodriguez, Pedro, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503. `https://doi.org/10.18653/v1/2021.acl-long.346`

Saito, Takaya and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):1–21. `https://doi.org/10.1371/journal.pone.0118432`, PubMed: 25738806

Sanguinetti, Manuela and Cristina Bosco. 2015. PartTUT: The Turin University Parallel Treebank. In Basili, Roberto, Cristina Bosco, Rodolfo Delmonte, Alessandro Moschitti, and Maria Simi, editors, *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*, volume 589, pages 51–69. `https://doi.org/10.1007/978-3-319-14206-7_3`

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*, pages 1–5.

Schreibman, Susan, Ray Siemens, and John Unsworth, editors. 2004. *A Companion to Digital Humanities*. Blackwell Publishing Ltd, Malden, MA, USA.

Shelmanov, Artem, Evgenii Tsymbalov, Dmitri Puzyrev, Kirill Fedyanin, Alexander Panchenko, and Maxim Panov. 2021. How certain is your transformer? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1833–1840. `https://doi.org/10.18653/v1/2021.eacl-main.157`

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Song, Hwanjun, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2020. Learning from noisy labels with deep neural networks: A survey. arxiv preprint, 2007.8199. `https://doi.org/10.1109/TNNLS.2022.3152527`, PubMed: 35254993

Stoica, George, Emmanouil Antonios Platanios, and Barnabas Poczos. 2021. Re-TACRED: Addressing shortcomings of the TACRED dataset. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence 2021*, pages 13843–13850. `https://doi.org/10.1609/aaai.v35i15.17631`

Sun, Chen, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–13.

Swayamdipta, Swabha, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and

Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293. `https://doi.org/10.18653/v1/2020.emnlp-main.746`

Szpiro, George. 2010. *Numbers Rule: The Vexing Mathematics of Democracy, from Plato to the Present*. Princeton University Press. `https://doi.org/10.1515/9781400834440`

Tjong Kim Sang, Erik F. and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147. `https://doi.org/10.3115/1119176.1119195`

Ustalov, Dmitry, Nikita Pavlichenko, Vladimir Losev, Iulian Giliazev, and Evgeny Tulin. 2021. A general-purpose crowdsourcing computational quality control toolkit for Python. In the *Ninth AAAI Conference on Human Computation and Crowdsourcing: Works-in-Progress and Demonstration Track*, pages 1–4.

van Halteren, Hans. 2000. The detection of inconsistency in manually tagged text. In *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, pages 48–55.

Vlachos, Andreas. 2006. Active annotation. In *Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 64–71.

Wang, Zihan, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5153–5162. `https://doi.org/10.18653/v1/D19-1519`, PubMed: 31303768

Wilcoxon, Frank. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80. `https://doi.org/10.2307/3001968`

Wisniewski, Guillaume. 2018. Errator: A tool to help detect annotation errors in the Universal Dependencies project. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 4489–4493.

Yaghoub-Zadeh-Fard, Mohammad Ali, Boualem Benatallah, Moshe Chai Barukh, and Shayan Zamanirad. 2019. A study of incorrect paraphrases in crowdsourced user utterances. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 295–306. `https://doi.org/10.18653/v1/N19-1026`

Zadrozny, Bianca and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 609–616.

Zadrozny, Bianca and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 694–700.

Zeldes, Amir. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612. `https://doi.org/10.1007/s10579-016-9343-x`

Zhang, Xiang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, pages 649–657.

Zhang, Xin, Guangwei Xu, Yueheng Sun, Meishan Zhang, and Pengjun Xie. 2021. Crowdsourcing learning as domain adaptation: A case study on named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5558–5570. `https://doi.org/10.18653/v1/2021.acl-long.432`

Zheng, Guoqing, Ahmed Hassan Awadallah, and Susan Dumais. 2021. Meta label correction for noisy label learning. In *Proceedings of the Thirty-fifth AAAI Conference on Artificial Intelligence 2021*, pages 11053–11061. `https://doi.org/10.1609/aaai.v35i12.17319`

Chapter 9

From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains

From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains

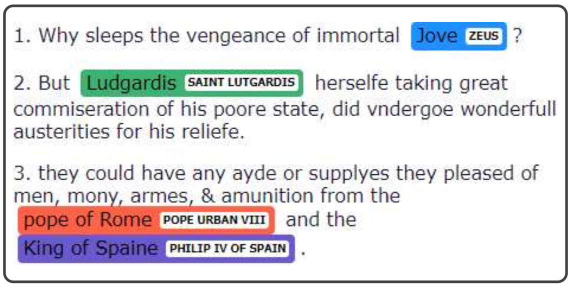
Jan-Christoph Klie **Richard Eckart de Castilho** **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science
Technical University of Darmstadt, Germany
www.ukp.tu-darmstadt.de

Abstract

Entity linking (EL) is concerned with disambiguating entity mentions in a text against knowledge bases (KB). It is crucial in a considerable number of fields like humanities, technical writing and biomedical sciences to enrich texts with semantics and discover more knowledge. The use of EL in such domains requires handling noisy texts, low resource settings and domain-specific KBs. Existing approaches are mostly inappropriate for this, as they depend on training data. However, in the above scenario, there exists hardly annotated data, and it needs to be created from scratch. We therefore present a novel domain-agnostic Human-In-The-Loop annotation approach: we use recommenders that suggest potential concepts and adaptive candidate ranking, thereby speeding up the overall annotation process and making it less tedious for users. We evaluate our ranking approach in a simulation on difficult texts and show that it greatly outperforms a strong baseline in ranking accuracy. In a user study, the annotation speed improves by 35 % compared to annotating without interactive support; users report that they strongly prefer our system. An open-source and ready-to-use implementation based on the text annotation platform INCEpTION[1] is made available[2].

1 Introduction

Entity linking (EL) describes the task of disambiguating entity mentions in a text by linking them to a knowledge base (KB), e.g. the text span *Earl of Orrery* can be linked to the KB entry *John Boyle, 5. Earl of Cork*, thereby disambiguating it. EL is highly beneficial in many fields like digital humanities, classics, technical writing or biomedical sciences for applications like search (Meij et al., 2014), semantic enrichment (Schlögl and Lejtovicz, 2017) or information extraction (Nooralahzadeh and Øvrelid, 2018). These are overwhelmingly low-resource settings: often, no data annotated exists; coverage of open-domain knowledge bases like Wikipedia or DBPedia is low. Therefore, entity linking is frequently performed against domain-specific knowledge bases (Munnelly and Lawless, 2018a; Bartsch, 2004).



1. Why sleeps the vengeance of immortal Jove ZEUS ?

2. But Ludgardis SAINT LUTGARDIS herselfe taking great commiseration of his poore state, did vndergoe wonderfull austerities for his reliefe.

3. they could have any ayde or supplyes they pleased of men, mony, armes, & amunition from the pope of Rome POPE URBAN VIII and the King of Spaine PHILIP IV OF SPAIN .

Figure 1: Difficult entity mentions with their linked entities: 1) Name variations, 2) Spelling Variation, 3) Ambiguity

In these scenarios, the first crucial step is to obtain annotated data. This data can then be either directly used by researchers for their downstream task or to train machine learning models for automatic annotation. For this initial data creation step, we developed a novel Human-In-The-Loop (HITL) annotation approach. Manual annotation is laborious and often prohibitively expensive. To improve annotation speed and quality, we therefore add interactive machine learning annotation support that helps the user find entities in the text and select the correct knowledge base entries for them. The more entities are annotated, the better the annotation support will be.

Throughout this work, we focus on texts from digital humanities, to be more precise, texts written in Early Modern English texts, including poems, biographies, novels as well as legal documents. In

[1] https://inception-project.github.io

[2] https://github.com/UKPLab/acl2020-interactive-entity-linking

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6982–6993
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

this domain, texts are noisy as they were written in times where orthography was rather incidental or due to OCR and transcription errors (see Fig. 1). Tools like named entity recognizers are unavailable or perform poorly (Erdmann et al., 2019).

We demonstrate the effectiveness of our approach with extensive simulation as well as a user study on different, challenging datasets. We implement our approach based on the open-source annotation platform INCEpTION (Klie et al., 2018) and publish all datasets and code. Our contributions are the following:

1. We present a generic, KB-agnostic annotation approach for low-resource settings and provide a ready-to-use implementation so that researchers can easily annotate data for their use cases. We validate our approach extensively in a simulation and in a user study.

2. We show that statistical machine learning models can be used in an interactive entity linking setting to improve annotation speed by over 35%.

2 Related work

In the following, we give a broad overview of existing EL approaches, annotation support and Human-In-The-Loop annotation.

Entity Linking describes the task of disambiguating mentions in a text against a knowledge base. It is typically approached in three steps: 1) *mention detection*, 2) *candidate generation* and 3) *candidate ranking* (Shen et al., 2015) (Fig. 2). Mention detection most often relies either on gazetteers or pretrained named entity recognizers. Candidate generation either uses precompiled candidate lists derived from labeled data or uses full-text search. Candidate ranking assigns each candidate a score, then the candidate with the highest score is returned as the final prediction. Existing systems rely on the availability of certain resources like a large Wikipedia as well as software tools and often are restricted in the knowledge base they can link to. Off-the-shelf systems like `Dexter` (Ceccarelli et al., 2013), `DBPedia Spotlight` (Daiber et al., 2013) and `TagMe` (Ferragina and Scaiella, 2010) most often can only link against Wikipedia or a related knowledge base like Wikidata or DBPedia. They require good Wikipedia coverage for computing frequency statistics like popularity, view count or PageRank (Guo et al., 2013). These features work very well for standard datasets due to their Zipfian distribution of entities, leading to high reported scores on state-of-the art datasets (Ilievski et al., 2018; Milne and Witten, 2008). However, these systems are rarely applied out-of-domain such as in digital humanities or classical studies. Compared to state-of-the-art approaches, only a limited amount of research has been performed on entity linking against domain-specific knowledge bases. `AGDISTIS` (Usbeck et al., 2014) developed a knowledge-base-agnostic approach based on the HITS algorithm. The mention detection relies on gazetteers compiled from resources like Wikipedia and thereby performs string matching. Brando et al. (2016) propose `REDEN`, an approach based on graph centrality to link French authors to literary criticism texts. It requires additional linked data that is aligned with the custom knowledge base–they use DBPedia. As we work in a domain-specific low resource setting, access to large corpora which can be used to compute popularity priors is limited. We do not have suitable named entity linking tools, gazetteers or a sufficient amount of labeled training data. Therefore, it is challenging to use state of the art systems.

Human-in-the-loop annotation HITL machine learning describes an interactive scenario where a machine learning (ML) system and a human work together to improve their performance. The ML system gives predictions, and the human corrects if they are wrong and helps to spot things that have been overlooked by the machine. The system uses this feedback to improve, leading to better predictions and thereby reducing the effort of the human. In natural language processing, it has been applied in scenarios like interactive text summarization (Gao et al., 2018), parsing (He et al., 2016) or data generation (Wallace et al., 2019). Regarding machine-learning assisted annotation, Yimam et al. (2014) propose an annotation editor that during annotation, interactively trains a model using annotations made by the user. They use string matching and MIRA (Crammer and Singer, 2003) as recommenders, evaluate on POS and NER annotation and show improvement in annotation speed. `TASTY` (Arnold et al., 2016) is a system that is able to perform EL against Wikipedia on the fly while typing a document. A pretrained neural sequence tagger is being used that performs mention detection. Candidates are precomputed and the candidate is chosen that has the highest text sim-

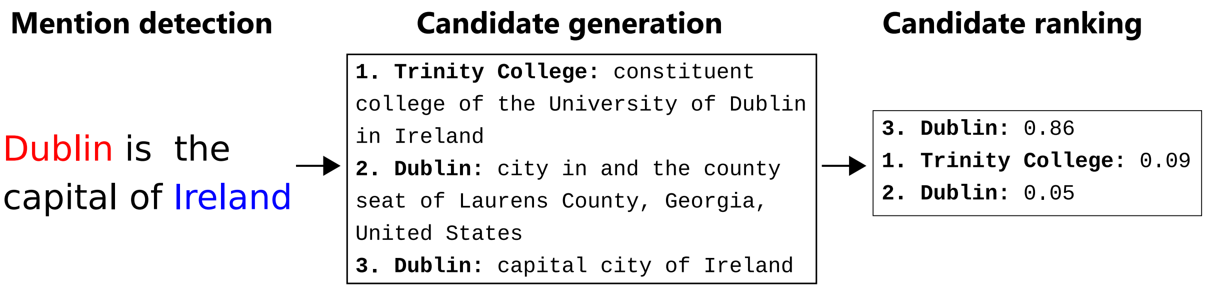

Mention detection
Candidate generation
Candidate ranking
Dublin is the capital of Ireland
1. Trinity College: constituent college of the University of Dublin in Ireland
2. Dublin: city in and the county seat of Laurens County, Georgia, United States
3. Dublin: capital city of Ireland
3. Dublin: 0.86
1. Trinity College: 0.09
2. Dublin: 0.05

Figure 2: Entity linking pipeline: First, mentions of entities in the text need to be found. Then, given a mention, candidate entities are generated. Finally, entities are ranked and the top entity is chosen.

ilarity. The system updates its suggestions after interactions such as writing, rephrasing, removing or correcting suggested entity links. Corrections are used as training data for the neural model. However, due to the following reasons, it is not yet suitable for our scenario. In order to overcome the cold start problem, it needs annotated training data in addition to a precomputed index for candidate generation. It also only links against Wikipedia.

3 Architecture

The following section describes the three components of our annotation framework, following the standard entity linking pipeline (see Fig. 2). Throughout this work, we will mainly focus on the *candidate Ranking* step. We call the text span which contains an entity the *mention* and the sentence the mention is in the *context*. Each candidate from the knowledge base is assumed to have a label and a description. For instance, in Fig. 2, one mention is *Dublin*, the context is *Dublin is the capital of Ireland*, the label of the the first candidate is *Trinity College* and its description is *constituent college of the University of Dublin in Ireland*.

Mention Detection In the annotation setting, we rely on users to mark text spans that contain annotations. As support, we provide suggestions given by different recommender models: similar to Yimam et al. (2014), we use a string matcher suggesting annotations for mentions which have been annotated before. We also propose a new Levenshtein string matcher based on Levenshtein automata (Schulz and Mihov, 2002). In contrast to the string matcher, it suggests annotations for spans within a Levenshtein distance of 1 or 2. Preliminary experiments with ML models for mention detection like using a Conditional Random Field and handcrafted features did not perform well and yielded noisy suggestions, requiring further investigation.

Candidate Generation We index the knowledge base and use full text search to retrieve candidates based on the surface form of the annotated mention. Besides, users can query this index during annotation. We use fuzzy search to help in cases where the mention and the knowledge base label are almost the same but not identical (e.g. *Dublin* vs. *Dublyn*). In the interactive setting, the user can also search the knowledge base during annotation, e.g. in cases when the gold entity is not ranked high enough or when the surface form and knowledge base label are not the same (*Zeus* vs. *Jupiter*).

Candidate Ranking We follow Zheng et al. (2010) and model candidate ranking as a learning-to-rank problem: given a mention and a list of candidates, sort the candidates so that the most relevant candidate is at the top. For training, we guarantee that the gold candidate is present in the candidate list. For evaluation, the gold candidate can be absent from the candidate list if the candidate search failed to find it.

This interaction is the core Human-in-the-loop in our approach. For training, we rephrase the task as preference learning: By selecting an entity label from the candidate list, users express that the selected one was preferred over all other candidates. These preferences are used to train state-of-the-art pairwise learning-to-rank models from the literature: the gradient boosted trees variant `LightGBM` (Ke et al., 2017), `RankSVM` (Joachims, 2002) and `RankNet` (Burges et al., 2005). Models are retrained in the background when new annotations are made, thus improving over time with an increasing number of annotations. We use a set of generic handcrafted features which are described in Table 1. These models were chosen as they can work with low data, train quickly and allow introspection. Using deep models or word embeddings as input features showed to be too slow to be interactive. We also leverage pretrained Sentence-BERT embeddings (Reimers and Gurevych, 2019) trained on Natural Language Inference data written in simple English. These are not fine-tuned by us during training. Although they come from a different domain, we conjecture that the WordPiece tokenization of BERT helps with the spelling variance of our texts in contrast to traditional word embeddings which would have many out-of-vocabulary words. For specific tasks, custom features can easily be incorporated e.g. entity type information, time information for diachronic entity linking, location information or distance for annotating geographical entities.

• Mention exactly matches label • Label is prefix/postfix of mention • Mention is prefix/postfix of label • Label is substring of mention and vice versa
• Levenshtein distance between mention and label • Levenshtein distance between context and description • Jaro-Winkler distance between mention and label • Jaro-Winkler distance between context and description • Sørensen-Dice index between context and description • Jaccard coefficient between context and description
• Exact match of Soundex encoding of mention and label • Phonetic Match Rating of mention and label
• Cosine distance between SBERT Embeddings of context and description (Reimers and Gurevych, 2019)
• Query length ∗ Query exactly matches label ∗ Query is prefix/postfix of label/mention ∗ Query is substring of mention/label ∗ Levenshtein distance between query and label • Levenshtein distance between query and mention • Jaro-Winkler distance between query and label • Jaro-Winkler distance between query and mention

Table 1: Features used for candidate ranking. Starred features were also used by Zheng et al. (2010)

4 Datasets

There are very few datasets available that can be used for EL against domain-specific knowledge bases, further stressing our point that we need more of these, thereby requiring approaches like ours to create them. We use three datasets: *AIDA-YAGO*, *Women Writers Online* (WWO) and *1641 Depositions*. *AIDA* consists of Reuters news stories. To the best of our knowledge, WWO has not been considered for automatic EL so far. The 1641 Depositions have been used in automatic EL, but only when linking against DBPedia which has a very low entity coverage (Munnelly and Lawless, 2018b). We preprocess the data, split it in sentences, tokenize and reduce noise. For WWO, we derive a RDF KB from their personography, for 1641 we derive a knowledge base from the annotations. The exact processing steps as well as example texts are described in the appendix. The resulting data sets for WWO and *1641 Depositions* are also made available in the accompanying code repository.

AIDA-YAGO: For validating our approach, we evaluate on the AIDA-YAGO state-of-the art dataset introduced by Hoffart et al. (2011). Originally, this dataset is linked against YAGO and Wikipedia. We map the Wikipedia URLs to Wikidata and link against this KB, as Wikidata is available in RDF and the official Wikidata SPARQL endpoint offers full text search: it does not offer fuzzy search though.

Women Writers Online: *Women Writers Online*[3] is a collection of texts by pre-Victorian women writers. It includes texts on a wide range of topics and from various genres including poems, plays, and novels. They represent different states of the English language between 1400 and 1850. A subset of documents has been annotated with named entities (persons, works, places) (Melson and Flanders, 2010). Persons have also been linked to create a personography, a structured representation of persons' biographies containing names, titles, time and place of birth and death. The texts are challenging to disambiguate due to spelling variance, ciphering of names and a lack of standardized orthography. Sometimes, people are not referred to by name but by rank or function, e.g. *the king*. This dataset is interesting, as it contains documents with heterogeneous topics and text genres, causing low redundancy.

1641 Depositions: The *1641 Depositions*[4] contain legal texts in form of court witness statements recorded after the Irish Rebellion of 1641. In this conflict, Irish and English Catholics revolted against English and Scottish Protestants and their colonization of Ireland. It lasted over 10 years and ended with the Irish Catholics' defeat and the foreign rule of Ireland. The depositions have been transcribed from 17th century handwriting, keeping the old language and orthography. These documents have been used to analyze the rebellion, perform cold case reviews of the atrocities committed and to gain insights into contemporary life of this era. Part of the documents have been annotated

[3] https://www.wwp.northeastern.edu/wwo

[4] http://1641.tcd.ie/

Table 2: Data statistics of the three used datasets: Total number of **D**ocuments, **T**okens, **E**ntities, average number of **E**ntities per **S**entence, % of entities that are not linked. We also report the average number of entities linked to a mention, the average number of candidates when searching for a mention in the KB and the Gini coefficient which measures how balanced the entity distribution is.

Corpus	#D	#T	#E	#E/S	%NIL	Avg. Amb.	Avg. #Cand.	Gini
AIDA	1393	301,418	34,929	1.59	20.37	1.08	6.98	0.73
WWO	74	1,461,401	14,651	0.34	7.42	1.08	16.66	0.56
1641	16	11,895	480	2.40	0.0	1.01	36.29	0.44

with named entities that are linked to DBPedia (Munnelly and Lawless, 2018b). As the coverage of DBPedia was not sufficient (only around 20% of the entities are in DBPedia), we manually created a domain specific knowledge base for this data set containing places and people mentioned. To increase difficulty and reduce overfitting, we added additional related entities from DBPedia. The number of persons increases thereby by tenfold (130 → 1383) and the number of places by twentyfold (99 → 2119). Details for that can be found in Appendix A.1. While generating a KB from gold data is not ideal, creating or completing a knowledge base during annotation is not uncommon (see e.g. Wolfe et al., 2015). The texts are difficult to disambiguate due to the same reasons as for WWO. The depositions are interesting, as they contain documents from the same domain (witness reports), but feature many different actors and events.

Table 2 contains several statistics regarding the three datasets. AIDA and 1641 contain on average at least one entity per sentence, whereas WWO, while larger, is only sparsely annotated. In contrast to the other two, 1641 contains no entities linked to NIL. This is caused by the fact that we created the KB for 1641 from the gold annotations and for entities previously NIL, new entities were created by hand ; before that, the original corpus linking to DBPedia had 77% NIL annotations. The average ambiguity, that is, how many different entities were linked to mentions with the same surface form is quite high for AIDA and WWO and quite low for 1641. We explain the latter by the extreme variance in surface form, as even mentions of the same name are often written differently (e.g. *Castlekevyn* vs. *Castlekevin*). Also, 1641 contains many hapax legomena (mentions that only occur once). The average number of candidates is comparatively larger for WWO and 1641 as we use fuzzy search for these. Finally, the distributions of assigned entities in WWO and 1641 are also more balanced, expressed by a lower Gini coefficient (Dodge, 2008). These last two aspects together with noisy texts and low resources causes entity linking to be much more difficult compared to state-of-the-art datasets like AIDA.

5 Experiments

To validate our approach, we first evaluate recommender performance. Then, non-interactive ranking performance is evaluated similarly to state-of-the-art EL. Afterwards, we simulate a user annotating corpora with our Human-In-The-Loop ranker. Finally, we conduct a user study to test it in a realistic setting. Similar to other work on EL, our main metric for ranking is accuracy. We also measure Accuracy@5, as our experiments showed that users can quickly scan and select the right entity from a list of five elements. In our annotation editor, the candidate list shows the first five elements without scrolling. As a baseline, we use the Most-Frequently Linked Entity baseline (MFLEB). It assigns, given a mention, the entity that was most often linked to it in the training data.

5.1 Automatic suggestion performance

We evaluate the performance of our Levenshtein-based recommender that suggests potential annotations to users (Table 3). We filter out suggestions consisting of $\leq$ 3 characters as these introduce too much noise. For annotation suggestions, we focus on recall: where low precision implies recommendations that are not useful, no recall results in no recommendations at all. It can be seen that for AIDA and WWO, the performance of all three recommenders is quite good (recall is about 60% and 40%) while for 1641, it is only around 20%. The Levenshtein recommender increases recall and reduces precision. The impact is most pronounced for 1641, where it improves recall upon the string matching recommender by around 50%. In summary, we suggest using the string matching rec-

Dataset	Model	P	R	F1
AIDA	String	**0.43**	**0.60**	**0.50**
	Leven@1	0.31	0.55	0.40
	Leven@2	0.19	0.57	0.28
WWO	String	**0.17**	0.38	**0.23**
	Leven@1	0.11	0.40	0.16
	Leven@2	0.04	**0.42**	0.07
1641	String	0.12	0.14	0.13
	Leven@1	**0.16**	0.19	**0.17**
	Leven@2	0.12	**0.22**	0.15

Table 3: Recommender performance in **P**recision, **R**ecall and **F1** score for **String** matching recommender and **Leven**shtein recommender with distance 1 and 2. For AIDA, we evaluate on the test set, for the other datasets, we use 10-fold cross validation.

ommender for domains where texts are clean and exhibit low spelling variance. We consider the Levenshtein recommender to be more suitable for domains with noisy texts.

5.2 Candidate ranking performance

We evaluate EL candidate ranking in a non-interactive setting first to estimate the upper bound ranking performance. As we are the first to perform EL on our version of WWO and 1641, it also serves as a difficulty comparison between AIDA as the state-of-the-art dataset and datasets from our domain-specific setting. For AIDA, we use the existing train, development and test split; for the other two corpora, we perform 10-fold cross validation as we observed high variance in score when using different train-test splits. Features related to user queries are not used in this experiment. We assume that the gold candidate always exists in training and evaluation data. The results of this experiment are depicted in Table 4. It can be seen that for AIDA, the MFLE baseline is particularly strong, being better than all trained models. For the other datasets, the baseline is weaker than all, showing that popularity is a weak feature in our setting. For AIDA, LightGBM performs best, for WWO and 1641, the RankNet is best closely followed by the RankSVM. The accuracy@5 is comparatively high as there are cases where the candidate list is relatively short. Regarding training times, LightGBM trains extremely fast with RankSVM being a close second. They are fast enough to retrain after each user annotation. The RankNet trains two to four times slower than both.

Data	Model	A@1	A@5	$\vert C\vert$	t
AIDA	MFLEB	**0.56**	0.71	31	
	LightGBM	0.44	**0.72**		9
	RankSVM	0.37	0.69		56
	RankNet	0.42	0.70		190
WWO	MFLEB	0.32	0.77	19	
	LightGBM	0.37	0.83		2
	RankSVM	0.46	0.86		15
	RankNet	**0.52**	**0.87**		37
1641	MFLEB	0.28	0.75	38	
	LightGBM	0.35	0.77		1
	RankSVM	0.48	0.80		1
	RankNet	**0.55**	**0.83**		2

Table 4: Ranking scores when using all the data. We report **A**ccuracy@1 (Gold Candidate was ranked highest, **A**ccuracy@5 (Gold Candidate was in top 5 predictions of the ranker)). $|C|$ denotes the average number of candidates found for each mention. For AIDA, we evaluate on the test set, for the other datasets, we use 10-fold cross validation. We also measure the training time t in seconds averaged over 10 runs.

Feature importance The models we chose for ranking are white-box; they allow us to introspect the importance they give to each feature, thereby explaining their scoring choice. For the RankSVM, we follow Guyon et al. (2002) and use the square of the model weights as importance. For LightGBM, we use the number of times a feature is used to make a split in a decision tree. We train RankSVM and LightGBM models on all data and report the most important and least important features in Fig. 3. We normalize the weights by the L1-norm. It can be seen that both models rely on Levenshtein distance between mention and label as well as Sentence-BERT. The other text similarity features are, while sparingly, also used. Simple features like exact match, contains or prefix and postfix seem to not have a large impact. In general, LightGBM uses more features than the RankSVM. Even though Sentence-BERT was trained on Natural Language Inference (NLI) data which contains only relatively simple sentences, it still is relied on by both models for all datasets. The high importance of Levenshtein distance between mention and label for 1641 is expected and can be explained by the fact that the knowledge base labels often were derived from the mentions in the text when creating a domain-specific knowledge base for this dataset. When trained on AIDA, the RankSVM assigns a high importance to the Jaccard distance between context and description. We attribute this to the fact that entity descriptions in Wikidata are quite short; if they are similar to the context then it is very likely a match.

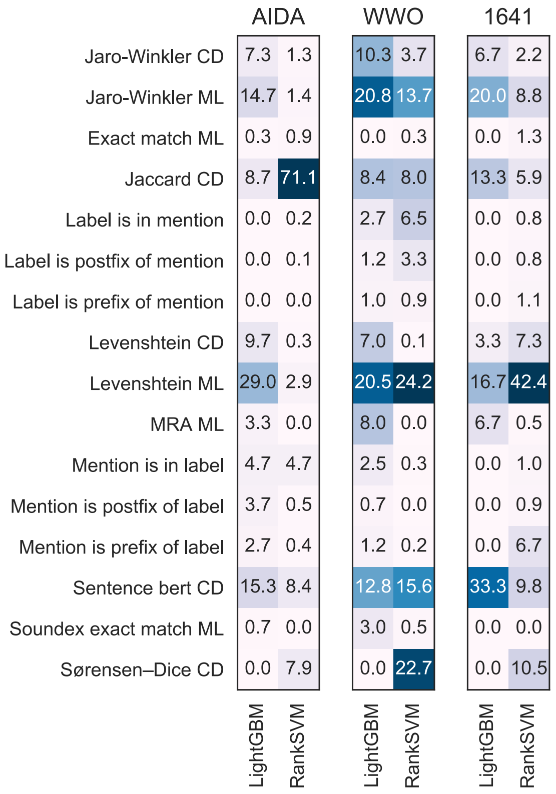


Figure 3: Feature importance of the respective models for different datasets. For the RankSVM, we use the squared weights; for LightGBM, we use the number of times a feature is used for splitting. Both are normalized to sum up to 1. ML stands for **M**ention-**L**abel, CD for **C**ontext-**D**escription.

5.3 Simulation

We simulate the Human-In-The-Loop setting by modeling a user annotating an unannotated corpus linearly. In the beginning, they annotate an initial seed of 10 entities without annotation support which are then used to bootstrap the ranker. At every step, the user annotates several entities where the ranker is used as assistance. After an annotation batch is finished, this new data is added to the training set, the ranker is retrained and evaluated. Only LightGBM and RankSVM are used as the RankNet turned out to be too slow. We do not evaluate on a holdout set. Instead, we follow Erdmann et al. (2019) and simulate annotating the complete corpus and evaluate on the very same data as we are interested in how an annotated subset helps to annotate the rest of the data, not how well the model generalizes. We assume that users annotate mention spans perfectly, i.e. we use gold spans. The candidate generation is simulated in three phases. It relies on the fact that the gold entity is given by the dataset: First, search for the mention only. If it was not found, search for the first word of the mention only. If this does not return the gold entity, search for the gold entity label. All candidates retrieved by these searches for a mention are used as training data. We also experimented with using only candidates for that the ranker assigned a higher score than the gold one. This, however, did not affect the performance. Therefore, we use all negative candidates.

Fig. 4 depicts the simulation results. All models outperform the MFLE baseline over most of the annotation process. It can be seen that both of our used models achieve high performance even if trained on very few annotations. The RankSVM handles low data better than LightGBM, but quickly reaches its peak performance due to it being a linear model with limited learning capacity. The LightGBM does not plateau that early. This potentially allows to first use a RankSVM for the cold start and when enough annotations are made, LightGBM, thereby combining the best of both models. Comparing the performance on the three datasets, we notice that the performance for AIDA is much higher. Also, the baseline rises much more steeply, hinting again that AIDA is easier and popularity there is a very strong feature. For 1641, the curve continue to rise, hinting that more data is needed to reach maximum performance.

Dataset	Phase 1	Phase 2	Phase 3
AIDA	0.20	0.00	0.80
WWO	0.26	0.27	0.47
1641	0.55	0.06	0.39

Table 5: Percentage of times the simulated user found the gold entity in the candidate list by searching for the mention (Phase 1), for the first word of the mention (Phase 2) or for the gold label (Phase 3).

Table 5 shows how the simulated user searched for the gold entities. We see that for WWO and 1641, the user often does not need to spend much effort in searching for the gold label, using the mention is in around 50% of the cases enough. We attribute this to the fuzzy search which the official Wikidata endpoint does not offer.

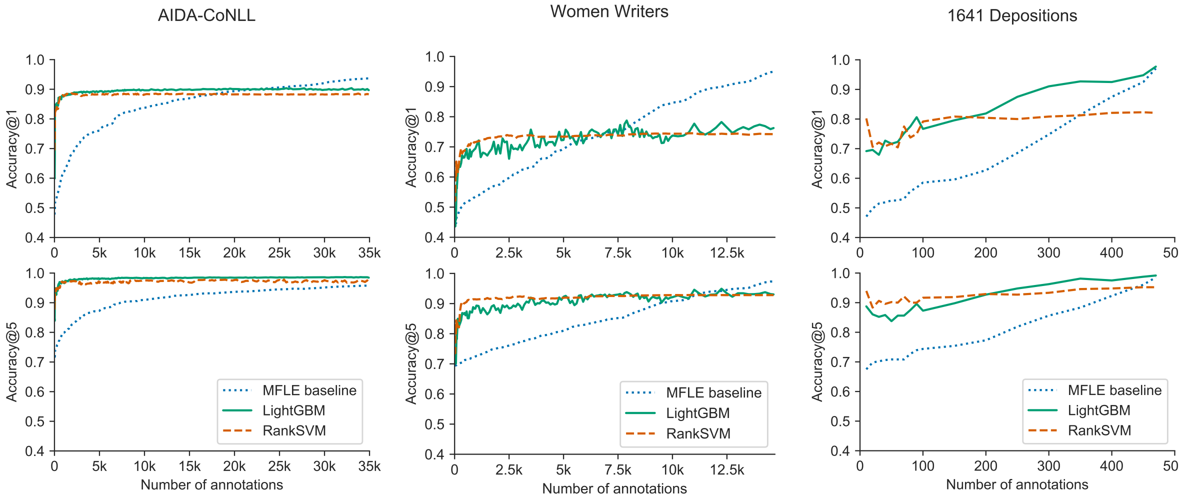


Figure 4: Human-in-the-loop simulation results for our three datasets and models. We can see that we get good Accuracy@5 with only a few annotations, especially for the RankSVM. This shows that the system is useful even at the beginning of the annotation process, alleviating the cold start problem.

5.4 User Study

In order to validate the viability of our approach in a realistic scenario, we conduct a user study. For that, we augmented the already existing annotation tool INCEpTION[5] (Klie et al., 2018) with our Human-In-The-Loop entity ranking and automatic suggestions. Fig. 5 shows a screenshot of the annotation editor itself. We let five users reannotate parts of the 1641 corpus. It was chosen as it has a high density of entity mentions while being small enough to be annotated in under one hour. Users stem from various academic backgrounds, e.g. natural language processing, computer science and digital humanities. Roughly half of them have previous experience with annotating. We compare two configurations: one uses our ranking and Levenshtein recommender, one uses the ranking of the full text search with the string matching recommender. We randomly selected eight documents which we split in two sets of four documents. To reduce bias, we assign users in four groups based on which part and which ranking they use first. Users are given detailed instructions and a warm-up document that is not used in the evaluation to get used to the annotation process. We measure annotation time, number of suggestions used and search queries performed. After the annotation is finished, we ask users to fill out a survey asking which system they prefer, how they experienced the annotation process and what suggestions they have to improve it. The evaluation of the user study shows that using our approach, users on average annotated 35% faster and needed 15% less search queries. Users positively commented on the ranking performance and the annotation suggestions for both systems. For our ranking, users reported that the gold entity often ranked first or close to top; they rarely observed that gold candidates were sorted close to the end of the candidate list.

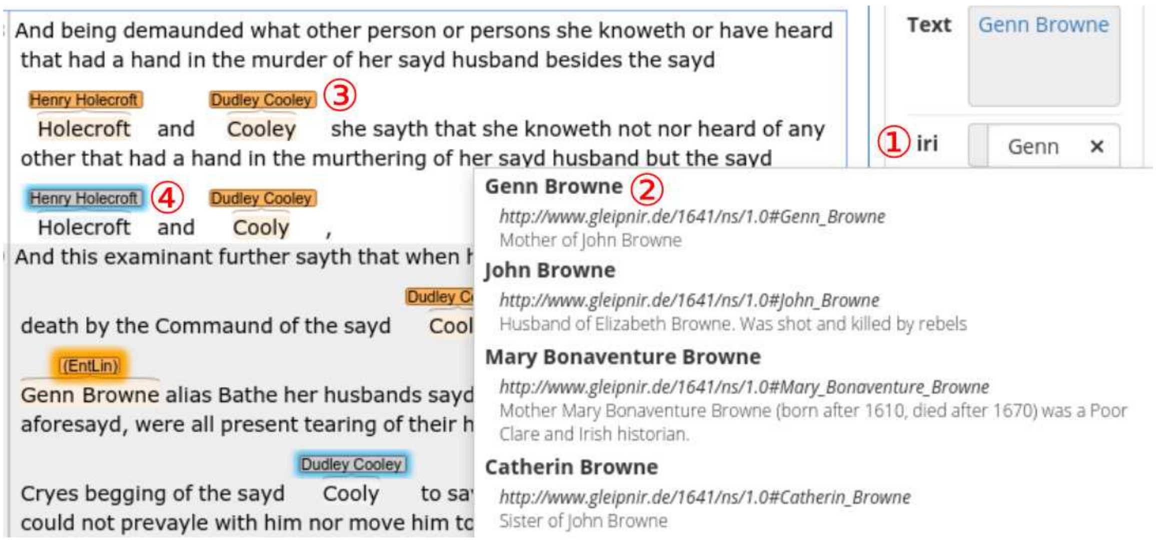
We conduct a paired sample t-test to estimate the significance of our user study. Our null-hypothesis is that the reranking system does not improve the average annotation time. Conducting the test yields the following: $t = 3.332, p = 0.029$. We therefore reject the null hypothesis with $p = 0.029 < 0.05$, meaning that we have ample evidence that our reranking speeds up annotation time.

Recommender suggestions made up around 30% of annotations. We did not measure a significant difference between string and Levenshtein recommender. About the latter, users liked that it can suggest annotations for inexact matches. However, they criticized the noisier suggestions, especially for shorter mentions (e.g. annotating *joabe* (a name) yielded suggestions for *to be*). In the future, we will address this issue by filtering out more potentially unhelpful suggestions and using annotation rejections as a blacklist.

6 Conclusion

We presented a domain-agnostic annotation approach for annotating entity linking for low-resource domains. It consists of two main com-

[5] https://inception-project.github.io



And being demaunded what other person or persons she knoweth or have heard that had a hand in the murder of her sayd husband besides the sayd Holecroft and Cooley she sayth that she knoweth not nor heard of any other that had a hand in the murthering of her sayd husband but the sayd Holecroft and Cooly ,

And this examinant further sayth that when h

death by the Commaund of the sayd Cool

Genn Browne alias Bathe her husbands sayd

aforesayd, were all present tearing of their h

Cryes begging of the sayd Cooly to sa

could not prevayle with him nor move him tc

Text: Genn Browne

iri: Genn ×

Genn Browne
http://www.gleipnir.de/1641/ns/1.0#Genn_Browne
Mother of John Browne

John Browne
http://www.gleipnir.de/1641/ns/1.0#John_Browne
Husband of Elizabeth Browne. Was shot and killed by rebels

Mary Bonaventure Browne
http://www.gleipnir.de/1641/ns/1.0#Mary_Bonaventure_Browne
Mother Mary Bonaventure Browne (born after 1610, died after 1670) was a Poor Clare and Irish historian.

Catherin Browne
http://www.gleipnir.de/1641/ns/1.0#Catherin_Browne
Sister of John Browne

Figure 5: For our user study, we extend the INCEpTION annotation framework: ① entity linking search field, ② candidate list, ③ linked named entity, ④ entity linking recommendation.

ponents: recommenders that are algorithms that suggest potential annotations to users and a ranker that, given a mention span, ranks potential entity candidates so that they show up higher in the candidate list, making it easier to find for users. Both systems are retrained whenever new annotations are made, forming the Human-In-The-Loop.

Our approach does not require the existence of external resources like labeled data, tools like named entity recognizers or large-scale resources like Wikipedia. It can be applied to any domain, only requiring a knowledge base whose entities have a label and a description. In this paper, we evaluate on three datasets: `AIDA`, which is often used to validate state-of-the-art entity linking systems as well as `WWO` and `1641` from the humanities. We show that in simulation, only a very small subset needs to be annotated (fewer than 100) for the ranker to reach high accuracy. In a user study, results show that users prefer our approach compared to the typical annotation process; annotation speed improves by around 35% when using our system relative to using no reranking support.

In the future, we want to investigate more powerful recommenders, combine interactive entity linking with knowledge base completion and use online learning to leverage deep models, despite their long training time.

Acknowledgments

We thank the anonymous reviewers and Kevin Stowe for their detailed and helpful comments. We also want to thank the Women Writers Project which made the *Women Writers Online* text collection available to us. This work was supported by the German Research Foundation under grant № EC 503/1-1 and GU 798/21-1 as well as by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 01UG1816B (CEDIFOR).

References

Sebastian Arnold, Robert Dziuba, and Alexander Löser. 2016. TASTY: Interactive Entity Linking As-You-Type. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 111–115.

Sabine Bartsch. 2004. Annotating a Corpus for Building a Domain-specific Knowledge Base. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, pages 1669–1672.

Carmen Brando, Francesca Frontini, and Jean-Gabriel Ganascia. 2016. REDEN: Named Entity Linking in Digital Literary Editions Using Linked Data Sets. *Complex Systems Informatics and Modeling Quarterly*, (7):60–80.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using Gradient Descent. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 89–96.

Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2013. Dexter. In *Proceedings of the sixth international workshop on Exploiting semantic annotations in information retrieval - ESAIR '13*, pages 17–20.

Koby Crammer and Yoram Singer. 2003. Ultraconservative Online Algorithms for Multiclass Problems. *JMLR*, 3:951–991.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems - I-SEMANTICS '13*, pages 121–124.

Yadolah Dodge. 2008. *The Concise Encyclopedia of Statistics*. Springer.

Alexander Erdmann, David Joseph Wrisley, Benjamin Allen, Christopher Brown, Sophie Cohen-Bodénès, Micha Elsner, Yukun Feng, Brian Joseph, Béatrice Joyeux-Prunel, and Marie-Catherine de Marneffe. 2019. Practical, Efficient, and Customizable Active Learning for Named Entity Recognition in the Digital Humanities. In *Proceedings of the 2019 Conference of the North*, pages 2223–2234.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*, pages 1625–1628.

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2018. APRIL: Interactively Learning to Summarise by Combining Active Preference Learning and Reinforcement Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4120–4130.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To Link or Not to Link? A Study on End-to-End Tweet Entity Linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1020–1030.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46:389–422.

Luheng He, Julian Michael, Mike Lewis, and Luke Zettlemoyer. 2016. Human-in-the-Loop Parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2337–2342.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *Proceedings of EMNLP'11*, pages 782–792.

Filip Ilievski, Piek Vossen, and Stefan Schlobach. 2018. Systematic Study of Long Tail Phenomena in Entity Linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 664–674.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, pages 133–142.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9.

Edgar Meij, Krisztian Balog, and Daan Odijk. 2014. Entity linking and retrieval for semantic search. In *Proceedings of the 7th ACM international conference on Web search and data mining - WSDM '14*, pages 683–684.

John Melson and Julia Flanders. 2010. Not Just One of Your Holiday Games: Names and Name Encoding in the Women Writers Project Textbase. White paper, Women Writers Project, Brown University.

David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*, pages 509–518.

Gary Munnelly and Séamus Lawless. 2018a. Constructing a knowledge base for entity linking on Irish cultural heritage collections. *Procedia Computer Science*, 137:199–210.

Gary Munnelly and Seamus Lawless. 2018b. Investigating Entity Linking in Early English Legal Documents. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries - JCDL '18*, pages 59–68.

Farhad Nooralahzadeh and Lilja Øvrelid. 2018. SIRIUS-LTG: An Entity Linking Approach to Fact Extraction and Verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 119–123.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3980–3990.

Matthias Schlögl and Katalin Lejtovicz. 2017. APIS - Austrian Prosopographical Information System. In *Proceedings of the Second Conference on Biographical Data in a Digital World 2017*.

Klaus U. Schulz and Stoyan Mihov. 2002. Fast string correction with Levenshtein automata. *International Journal on Document Analysis and Recognition*, 5(1):67–85.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. 2014. AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data. In *The Semantic Web – ISWC 2014*, pages 457–471.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick Me If You Can: Human-in-the-loop Generation of Adversarial Question Answering Examples. *Transactions of the Association for Computational Linguistics*, 7(0):387–401.

Travis Wolfe, Mark Dredze, James Mayfield, Paul McNamee, Craig Harman, Tim Finin, and Benjamin Van Durme. 2015. Interactive Knowledge Base Population.

Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96.

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Learning to Link Entities with Knowledge Base. In *Prooceedings of NAACL-HLT'10*, pages 483–491.

A Appendices

A.1 Dataset creation

The following section describes how we preprocess the raw texts from `WWO` and `1641`. Example texts can be found in Table 6. The respective code and datasets will be made available on acceptance.

A.1.1 Women Writers Online

We use the following checkout of the `WWO` data, which was graciously provided by the *Women Writers Project*[6].

```
Revision: 36425
Last Changed Rev: 36341
Last Changed Date: 2019-02-19
```

The texts itself are provided as TEI[7]. We use `DKPro Core`[8] to read in the TEI, split the raw text into sentences and tokenize it with the `JTokSegmenter`. When an annotation is spread over two sentences, we merge these sentences. This is mostly caused by a too eager sentence splitter. We covert the personographie which is in XML to RDF, including all properties that were encoded in there.

A.1.2 1641 Depositions

We use a subset of the 1641 depositions provided by Gary Munnelly. The raw data can be found on Github[9]. The texts itself are provided as NIF[10]. We use `DKPro Core`[11] to read in the NIF, split the raw text into sentences and tokenize it with the `JTokSegmenter`. When an annotation is spread over two sentences, we merge these sentences. This is mostly caused by a too eager sentence splitter. We use the knowledge base that comes with the NIF and create entities for all mentions that were `NIL`. We carefully deduplicate entities, e.g. `Luke Toole` and `Colonel Toole` are mapped to the same entity. In order to increase the difficulty of this dataset, we add additional entities from DBPedia: all Irish people, Irish cities and buildings in Ireland; all popes; royalities born between 1550 and 1650.

For that, we execute SPARQL queries against DBPedia for instances of `dbc:Popes`, `dbc:Royality`, `dbc:17th-century_Irish_people` and keep entries with a birth date before 1650 and a death date between 1600 and 1700. For the places, we search for `dbo:Castle`, `dbo:HistoricPlace`, `dbo:Building`, `dbc:17th-century_Irish_people` that are located in Ireland. The follwing table shows how many entities were in the original KB and how many were added:

Persons in gold data	130
Places in gold data	99
Persons added from DBPedia	1253
Places added from DBPedia	2020

[6] https://www.wwp.northeastern.edu/

[7] https://tei-c.org/

[8] https://dkpro.github.io/dkpro-core/

[9] https://github.com/munnellg/1641DepositionsCorpus

[10] https://persistence.uni-leipzig.org/nlp2rdf/

[11] https://dkpro.github.io/dkpro-core/

WWO
The following Lines occasion'd by the Marriage of Edward Herbert Esquire, and Mrs. Elizabeth Herbert. Cupid one day ask'd his Mother , When she meant that he shou'd Wed? You're too Young, my Boy, she said: Nor has Nature made another Fit to match with Cupid's Bed. Finch, Anne: Miscellany poems, on several occasions, 1713
Joseph Joice of Kisnebrasney in the kings County gentleman sworne and examined deposeth and saith That after the Rebellion was begun in the County aforesaid vizt about the xxth of November 1641 This deponent for saffty fled to the Castle of knocknamease in the same County
Deposition of Joseph Joice, 1643[12]

Table 6: Example sentences from these corpora. Linked Named entities are highlighted in yellow.

A.2 Experiments

A.2.1 Full text search

For AIDA and Wikidata, we use the official SPARQL endpoint and the Mediawiki API Query Service[13]. It does not support fuzzy search. For WWO and 1641, we host the created RDF in a Fuseki[14] instance and use the builtin functionality to index via Lucene.

A.2.2 Timing

Timing was performed on a Desktop PC with Ryzen 3600 and a GeForce RTX 2060.

[13] https://www.mediawiki.org/wiki/Wikidata_Query_Service/User_Manual/MWAPI

[14] https://jena.apache.org/documentation/fuseki2/

Chapter 10

Annotation Curricula to Implicitly Train Non-Expert Annotators

Annotation Curricula to Implicitly Train Non-Expert Annotators

Ji-Ung Lee*
UKP Lab / TU Darmstadt
lee@ukp.informatik.tu-darmstadt.de

Jan-Christoph Klie*
UKP Lab / TU Darmstadt
klie@ukp.informatik.tu-darmstadt.de

Iryna Gurevych
UKP Lab / TU Darmstadt
gurevych@ukp.informatik.tu-darmstadt.de

Annotation studies often require annotators to familiarize themselves with the task, its annotation scheme, and the data domain. This can be overwhelming in the beginning, mentally taxing, and induce errors into the resulting annotations; especially in citizen science or crowdsourcing scenarios where domain expertise is not required. To alleviate these issues, this work proposes annotation curricula, a novel approach to implicitly train annotators. The goal is to gradually introduce annotators into the task by ordering instances to be annotated according to a learning curriculum. To do so, this work formalizes annotation curricula for sentence- and paragraph-level annotation tasks, defines an ordering strategy, and identifies well-performing heuristics and interactively trained models on three existing English datasets. Finally, we provide a proof of concept for annotation curricula in a carefully designed user study with 40 voluntary participants who are asked to identify the most fitting misconception for English tweets about the Covid-19 pandemic. The results indicate that using a simple heuristic to order instances can already significantly reduce the total annotation time while preserving a high annotation quality. Annotation curricula thus can be a promising research direction to improve data collection. To facilitate future research—for instance, to adapt annotation curricula to specific tasks and expert annotation scenarios—all code and data from the user study consisting of 2,400 annotations is made available.[1]

* Equal contribution.

This article has been updated since publication. Please see the erratum here for details:
https://doi.org/10.1162/coli_x_00469

[1] https://github.com/UKPLab/annotation-curriculum.

Submission received: 7 June 2021; revised version received: 9 December 2021; accepted for publication: 21 December 2021.

https://doi.org/10.1162/coli_a_00436

1. Introduction

Supervised learning and, consequently, annotated corpora are crucial for many downstream tasks to train and develop well-performing models. Despite improvements of models trained in a semi- or unsupervised fashion (Peters et al. 2018; Devlin et al. 2019), they still substantially benefit from labeled data (Peters, Ruder, and Smith 2019; Gururangan et al. 2020). However, labels are costly to obtain and require domain experts or a large crowd of non-expert annotators (Snow et al. 2008).

Past research has mainly investigated two approaches to reduce annotation cost and effort (often approximated by annotation time); namely, **active learning** and **label suggestions**. Active learning assumes that resources for annotating data are limited and aims to reduce the number of labeled instances by only annotating those that contribute most to model training (Lewis and Gale 1994; Settles 2012). This often results in sampled instances that are more difficult to annotate, putting an increased cognitive load on annotators, and potentially leading to a lower agreement or an increased annotation time (Settles, Craven, and Friedland 2008). Label suggestions directly target annotators by providing them with suggestions from a pre-trained model. Although they are capable of effectively reducing the annotation time (Schulz et al. 2019; Klie, Eckart de Castilho, and Gurevych 2020; Beck et al. 2021, they bear the risk of biasing annotators toward the (possibly erroneous) suggested label (Fort and Sagot 2010). Both these shortcomings render existing approaches better suited for domain-expert annotators who are less burdened by difficult annotation instances and are less prone to receiving erroneous label suggestions than non-expert annotators. Overall, we can identify a lack of approaches that (1) are less distracting or biased than label suggestions and (2) can also ease the annotation process for non-expert annotators. Especially, the increasing popularity of large-scale, crowdsourced datasets (Bowman et al. 2015; Sakaguchi et al. 2021) further amplifies the need for training methods that can also be applied in non-expert annotator scenarios (Geva, Goldberg, and Berant 2019; Nie et al. 2020; Rogers 2021).

One key element that has so far not been investigated in annotation studies is the use of a curriculum to *implicitly* teach the task to annotators during annotation. The **learning curriculum** is a fundamental concept in educational research that proposes to order exercises to match a learner's proficiency (Vygotsky 1978; Krashen 1982) and has even motivated training strategies for machine learning models (Bengio et al. 2009). Moreover, Kelly (2009) showed that such learning curricula can also be used to teach learners implicitly. Similarly, the goal of **annotation curricula** (AC) is to provide an ordering of instances during annotation that is optimized for learning the task. We conjecture that a good annotation curriculum can implicitly teach the task to annotators—for instance, by showing easier annotation instances before more difficult ones—consequently reducing the cognitive strain and improving annotation speed and quality. In contrast to active learning, which may result in only sampling instances that are difficult to annotate, they explicitly emphasize the needs of a human annotator and gradually familiarize them with the annotation task. Compared to label suggestions, they are less distracting as they do not bear the risk of providing erroneous suggestions from imperfect models, making them well-suited for non-expert annotation scenarios. Furthermore, AC do not require study conductors to adapt existing annotator training processes or annotation guidelines and hence, can complement their annotation project. To provide a first assessment for the viability of such annotation curricula, we investigate the following three research questions:

RQ1. Does the order in which instances are annotated impact the annotations in terms of annotation time and quality?

RQ2. Do traditional heuristics and recent methods for assessing the reading difficulty already suffice to generate curricula that improve annotation time or quality?

RQ3. Can the generation of annotation curricula be further alleviated by interactively trained models?

We first identify and formalize two essential parts to deploy AC: (1) a "strategy" that defines how instances should be ordered (e.g., by annotation difficulty) and (2) an "estimator" that ranks them accordingly. We instantiate AC with an "easy-instances-first" strategy and evaluate heuristic and interactively trained estimators on three English datasets that provide annotation time which we use as an approximation of the annotation difficulty for evaluation. Finally, we apply our strategy and its best estimators in a carefully designed user study with 40 participants for annotating English tweets about the Covid-19 pandemic. The study results show that the ordering in which instances are annotated can have a statistically significant impact on the outcome. We furthermore find that annotators who receive the same instances in an optimized order require significantly less annotation time while retaining a high annotation quality. Our contributions are:

C1. A novel approach for training non-expert annotators that is easy to implement and is complementary to existing annotator training approaches.

C2. A formalization of AC for sentence- and paragraph-labeling tasks with a strategy that orders instances from easy to difficult, and an evaluation for three heuristics and three interactively trained estimators.

C3. A first evaluation of AC in a carefully designed user study that controls for external influences including:

a) An implementation of our evaluated annotation curriculum strategies and 2,400 annotations collected during our human evaluation study.

b) A production-ready implementation of interactive AC in the annotation framework INCEpTION (Klie et al. 2018) that can be readily deployed.

Our evaluation of different heuristics and interactively trained models further reveals additional factors—such as the data domain and the annotation task—that can influence their aptitude for AC. We thus appeal to study conductors to publish the annotation order and annotation times along with their data to allow future studies to better investigate and develop task- and domain-specific AC.

2. Related Work

Most existing approaches that help with data collection focus on either active learning or label suggestions. Other researchers also investigate tackling annotation task within the context of gamification and introduce different levels of difficulty.

Active Learning. Active learning has widely been researched in terms of model-oriented approaches (Lewis and Gale 1994), Roy and McCallum 2001; Gal, Islam, and Ghahramani 2017; Siddhant and Lipton 2018; Kirsch, van Amersfoort, and Gal 2019), data-oriented approaches (Nguyen and Smeulders 2004; Zhu et al. 2008; Huang, Jin, and Zhou 2010; Wang et al. 2017), or combinations of both (Ash et al. 2020; Yuan, Lin, and Boyd-Graber 2020). Although several works investigate annotator proficiency—which is especially important for crowdsourcing—their main concern is to identify noisy labels or erroneous annotators (Laws, Scheible, and Schütze 2011; Fang et al. 2012; Zhang and Chaudhuri 2015) or distribute tasks between workers of different proficiency (Fang, Yin, and Tao 2014; Yang et al. 2019). Despite the large amount of research in active learning, only a few studies have considered annotation time as an additional cost variable in active learning (Settles, Craven, and Friedland 2008) and even found that active learning can negatively impact annotation time (Martínez Alonso et al. 2015). Other practical difficulties for deploying active learning in real annotation studies stem from additional hyperparameters that are introduced, but seldom investigated (Lowell, Lipton, and Wallace 2019). In contrast, AC also work well with simple heuristics, allowing researchers to pre-compute the order of annotated instances.

Label Suggestions. Label suggestions have been considered for various annotation tasks in NLP, such as in part-of-speech tagging for low-resource languages (Yimam et al. 2014), interactive entity-linking (Klie, Eckart de Castilho, and Gurevych 2020), or identifying evidence in diagnostic reasoning (Schulz et al. 2019). Especially for tasks that require domain-specific knowledge such as in the medical domain, label suggestions can substantially reduce the burden on the annotator (Lingren et al. 2014). However, they also inherently pose the risk of amplifying annotation biases due to the anchoring effect (Turner and Schley 2016). Whereas domain experts may be able to reliably identify wrong suggestions and provide appropriate corrections (Fort and Sagot 2010), this cannot be assumed for non-experts. This renders label suggestions a less viable solution to ease annotations in non-expert studies where incorrect label suggestions may even distract annotators from the task. In contrast, changing the ordering in which instances are annotated by using AC is not distracting at all.

Annotation Difficulty. Although difficulty estimation is crucial in human language learning, for instance, in essay scoring (Mayfield and Black 2020) or text completion exercises (Beinborn, Zesch, and Gurevych 2014; Loukina et al. 2016; Lee, Schwan, and Meyer 2019), it is difficult to achieve in annotation scenarios due to the lack of ground truth, commonly resulting in a post-annotation analysis for model training (Beigman Klebanov and Beigman 2014; Paun et al. 2018). To consider the difficulty of annotated instances, a concept that has recently been explored for (annotation) games with a purpose, is **progression**. It allows annotators to progress through the annotation study similar to a game—by acquiring specific skills that are required to progress to the next level (Sweetser and Wyeth 2005). Although several works have shown the efficiency of progression in games with a purpose (Madge et al. 2019; Kicikoglu et al. 2020) and even in crowdsourcing (Tauchmann, Daxenberger, and Mieskes 2020), this does not necessarily benefit individual workers, as less-skilled workers are either filtered out or asked to "train" on additional instances. Moreover, implementing progression poses a substantial burden on researchers due to the inclusion of game-like elements (e.g., skills and levels), or at minimum, the separation of the data according to difficulty and, furthermore, a repeated evaluation and reassignment of workers. In contrast, reordering instances of a single set according to a given curriculum can already be achieved with low effort and can even be implemented complementary to progression.

3. Annotation Curriculum

We first specify the type of annotation tasks investigated in this work, and then formalize AC with the essential components that are required for generating appropriate annotation curricula. Finally, we instantiate an easy-instances-first strategy and define the estimators that we use to generate a respective curriculum.

3.1 Annotation Task

In this work, we focus on sentence- and paragraph-level annotation tasks that do not require any deep domain-expertise and hence are often conducted with non-expert annotators.[2] Such annotation tasks often use a simple annotation scheme limited to a small set of labels, and have been used to create datasets across various research areas, for instance, in sentiment analysis (Pak and Paroubek 2010), natural language inference (Bowman et al. 2015), and argument mining (Stab et al. 2018).

Task Formalization. We define an annotation task as being composed of a set of unlabeled instances $x \in \mathcal{U}$ that are to be annotated with their respective labels $y \in \mathcal{Y}$. We focus on instances x that are either a sentence or a paragraph and fully annotated by an annotator a. Note that for sequence labeling tasks such as named entity recognition, y is not a single label but a vector composed of the respective token-level labels. However, in such tasks, annotations are still often collected for a complete sentence or paragraph at once to provide annotators with the necessary context (Tomanek and Hahn 2009).

3.2 Approach

Figure 1 provides a general overview of AC. Given a set of unlabeled instances $x \in \mathcal{U}$, we define a strategy $\mathcal{S}$ that determines the ordering in which annotated instances should be presented (easy-instances-first). We then specify "adaptive" and "non-adaptive" estimators $f(\cdot)$ that approximate the true annotation difficulty. In this work, we focus on task-agnostic estimators that can easily be applied across a wide range of tasks and leave the investigation on task-specific estimators—which may have higher performance but also require more implementation effort from study conductors—for future work.[3] Depending on the estimator, we then order the annotated instances either beforehand (non-adaptive), or select them iteratively at each step based on the predictions of an interactively trained model (adaptive).

[2] We discuss AC strategies that may be better suited for domain experts in Section 6.

[3] We discuss some ideas for task-specific estimators in Section 6.

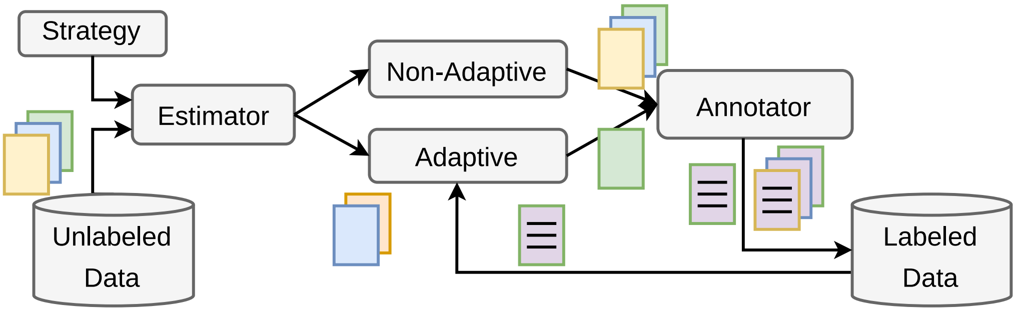


Figure 1
Annotation curricula. First, we define a strategy for ordering instances by annotation difficulty (i.e., easy-first). We then implement estimators that perform the ordering. Estimators can either be non-adaptive (e.g., heuristics) or adaptive (trained models). Finally, annotators receive instances according to the resulting curriculum.

Formalization. Ideally, an annotation curriculum that optimally introduces annotators to the task would minimize (1) annotation effort and (2) error rate (i.e., maximize annotation quality). As the annotation error can only be obtained post-study, we can only use annotation effort, approximated by annotation time, for our formalization; however, we conjecture that minimizing annotation time may also have a positive impact on annotation quality (given that the annotators remain motivated throughout their work). To further reduce noise factors during evaluation, we focus on annotation studies that involve a limited number of instances (in contrast to active learning scenarios that assume an abundance of unlabeled data). We thus formalize annotation curriculum as the task of finding the optimal curriculum $\mathcal{C}^*$ out of all possible curricula $\mathcal{C}$ (i.e., permutations of $\mathcal{U}$) for a finite set of unlabeled instances $\mathcal{U}$ that minimizes the total annotation time $\mathcal{T}$; namely, the sum of individual annotation times $t_i \in \mathbb{R}^+$ for all instances $x_i \in \mathcal{U}$ with i denoting the i-th annotated instance:

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \sum_{i=1}^{|\mathcal{U}|} a_i(x_i | x_0 \dots x_{i-1}) \tag{1}$$

where $a_i : \mathcal{U} \to \mathcal{T}$ describes the annotator after annotating $i-1$ instances.

Strategy. Due to the large number of $n!$ possible curricula $\mathcal{C}$ resulting from $n = |\mathcal{U}|$ instances, solving Equation 1 is intractable for large n even if $a(\cdot)$ was known. We can furthermore only assess the true effectiveness of a curriculum $\mathcal{C}$ post-study, making it impossible to find the optimal curriculum $\mathcal{C}^*$ beforehand. We hence require a strategy $\mathcal{S} \sim \mathcal{C}^*$ that specifies how instances of $\mathcal{U}$ should be ordered optimally. Similar to educational approaches, we rely on estimating the "difficulty" of an instance to generate our curriculum (Taylor 1953; Beinborn, Zesch, and Gurevych 2014; Lee, Schwan, and Meyer 2019). In this work, we investigate an easy-instances-first strategy that has been shown to be a reasonable strategy in previous work (Tauchmann, Daxenberger, and Mieskes 2020); thereby sorting instances in ascending order according to their difficulty. Our $\mathcal{C}^*$ is thus approximated by the ordered set $\mathcal{S} = \{x_1, \dots, x_n | \forall x_{1 \leq i \leq n} \in \mathcal{S} : f(x_i) \leq f(x_{i+1})\}$ with $f(\cdot)$ being the difficulty estimator.

Non-adaptive Estimators. We define non-adaptive estimators as heuristics or pre-trained models that are not updated interactively. The respective annotation curriculum can thus be pre-computed and does not impose any additional changes to the underlying annotation platform. To estimate the annotation difficulty, non-adaptive estimators define a scoring function $f_{\bar{a}} : \mathcal{U} \rightarrow \mathbb{R}$. In this work, we evaluate non-adaptive estimators that are commonly used in readability assessment to score the reading difficulty of a text (Xia, Kochmar, and Briscoe 2016; Deutsch, Jasbi, and Shieber 2020). Although they are not capable of capturing any task-specific difficulties, they have the advantage of being applicable to a wide range of tasks with low effort for study conductors. The following heuristics and pre-trained models are investigated to obtain difficulty estimations for the easy-instances-first curriculum:

Sentence Length (sen**)** The number of tokens in a sentence averaged across the whole document (i.e., the number of tokens for single sentence instances).

Flesch-Kincaid (FK**)** A readability score based on the number of words, syllables, and sentences (Kincaid et al. 1975).

Masked Language Modeling Loss (mlm**)** As shown in recent work, the losses of a masked language model may be used to obtain an assessment of text complexity (Felice and Buttery 2019). We use the implementation of Salazar et al. (2020).

Adaptive Estimators. While simple heuristics or annotator-unaware models allow us to pre-compute annotation curricula, they do not consider any user-specific aspect that may influence the difficulty estimation (Lee, Meyer, and Gurevych 2020). Consequently, the resulting curriculum may not provide the optimal ordering for a specific annotator. To select the instance with the most appropriate difficulty for an annotator $a_i(\cdot)$ at the i-th iteration, we use a model $\theta_i(\cdot) \sim a_i(\cdot)$ that is updated with an increasing number of annotated instances. We conjecture that using $\theta(\cdot)$ to predict the relative difficulty—in contrast to non-adaptive estimators that provide an absolute difficulty estimation—may be more robust to task-specific influences as they are inherited in all instances annotated by $a(\cdot)$. When training adaptive estimators, we use annotation time to approximate the difficulty of a specific instance due to its availability in any annotation scenario. At iteration i, we thus train the model $\theta_i : \mathcal{L} \rightarrow \mathcal{T} \subseteq \mathbb{R}^+$ to predict the annotation times $t \in \mathcal{T}$ for all labeled instances $\hat{x} \in \mathcal{L}$. Similar to active learning, we now encounter a decreasing number of unlabeled instances and an increasing number of labeled instances. The resulting model is then used to estimate the annotation time for all unlabeled instances $x \in \mathcal{U}$. The resulting scoring function is now defined as $f_a : \theta_i, \mathcal{U} \rightarrow \mathbb{R}^+$. Finally, we select instance $x^* \in \mathcal{U}$ with the minimal rank according to f_a.

$$x^* = \arg\min_{f_a} \theta_i(x) \tag{2}$$

Following our strategy $\mathcal{S}$, this results in selecting instances for annotation that have the lowest predicted annotation time. We specifically focus on regression models that can be trained efficiently in-between annotation and work robustly in low-data scenarios. We choose Ridge Regression, Gaussian Process Regression, and GBM Regression.

4. Evaluation with Existing Datasets

To identify well-performing non-adaptive and adaptive estimators, we first evaluate AC on existing datasets in an offline setting. We focus on datasets that provide annotation time which is used to approximate the annotation difficulty during evaluation (to address the lack of gold labels in actual annotation scenarios). Following Settles, Craven, and Friedland (2008), we conjecture that instances with a higher difficulty require more time to annotate. For comparison, we then compute the correlations between different orderings generated according to our easy-instances-first strategy using text difficulty heuristics (non-adaptive) and interactively trained models (adaptive) with the annotation time (approximated annotation difficulty). We evaluate our estimators in two setups:

Full We evaluate how well adaptive and non-adaptive estimators trained on the whole training set correlate with the annotation time of the respective test set (upper bound).

Adaptive We evaluate the performance of adaptive estimators in an interactive learning scenario with simulated annotators and an increasing number of training instances.

4.1 Datasets

Overall, we identify three NLP datasets that provide accurate annotation time for individual instances along with their labels:

Muc7$_T$ Tomanek and Hahn (2009) extended the MUC7 corpus that consists of annotated named entities in English Newswire articles. They reannotated the data with two annotators A and B while measuring their annotation time per sentence.

SigIE is a collection of email signatures that was tagged by Settles, Craven, and Friedland (2008) with twelve named entity types typical for email signatures such as phone number, name, and job title.

SPEC The same authors (Settles, Craven, and Friedland 2008) further annotated sentences from 100 English PubMed abstracts according to their used language (speculative or definite) with three annotators.

Table 1
Annotation task (ST for sequence tagging, Cl for classification) and the number of instances per dataset and split. $\mu_{|\mathcal{D}|}$ denotes the average instance length in characters and μ_t the average annotation time. $\sigma_{|\mathcal{D}|}$ and σ_t denotes the standard deviation, respectively. Across all datasets, annotation time is reported for annotating the whole instance (i.e., not for individual entities).

Name	Task	$\|\mathcal{D}\|$	$\|\mathcal{D}_{\textbf{train}}\|$	$\|\mathcal{D}_{\textbf{dev}}\|$	$\|\mathcal{D}_{\textbf{test}}\|$	$\mu_{\|\mathcal{D}\|}$	$\sigma_{\|\mathcal{D}\|}$	μ_t	σ_t
Muc7$_T$ A	ST	3,113	2,179	467	467	133.7	70.8	5.4	3.9
Muc7$_T$ B	ST	3,113	2,179	467	467	133.7	70.8	5.2	4.2
SigIE	ST	251	200	–	51	226.4	114.8	27.0	14.7
SPEC	Cl	850	680	–	170	160.4	64.2	22.7	12.4

Table 1 provides an overview of the used datasets. It can be seen that $\mathrm{Muc7_T}$ is the largest corpus ($|\mathcal{D}|$); however, it is also the one that consists of the shortest instances on average ($\mu_{|\mathcal{D}|}$). Furthermore, $\mathrm{Muc7_T}$ also has the lowest annotation times (μ_t) and a low standard deviation (σ_t). Comparing the number of entities per instance between $\mathrm{Muc7_T}$ (news articles) and SigIE (email signatures) shows their differences with respect to their domains with an average number of 1.3 entities ($\sigma = 1.4$) in $\mathrm{Muc7_T}$ and 5.3 entities ($\sigma = 3.0$) in SigIE. Moreover, we find that the SigIE corpus has a higher ratio of entity tokens (40.5%) than $\mathrm{Muc7_T}$ (8.4%), which may explain the long annotation time. Interestingly, the binary sentence classification task SPEC ("speculative" or "definite") also displays a substantially longer annotation time compared to $\mathrm{Muc7_T}$ (on average, more than four times), which may also indicate a higher task difficulty or less proficiency of the involved annotators.

Data splits. For $\mathrm{Muc7_T}$, we focus on the annotations of the first annotator $\mathrm{Muc7_T}$ A; using $\mathrm{Muc7_T}$ B yields similar results. For SPEC, we use ALL.DAT for our experiments. None of the aforementioned datasets provide default splits. We hence create 80-20 train-test splits of SPEC and SigIE for our experiments. To identify the best hyperparameters of our adaptive estimators, we split the largest corpus ($\mathrm{Muc7_T}$) into 70-15-15 train-dev-test splits. All splits are published along with the code and data.

4.2 Experimental Setup

Our goal is to evaluate how well the ordering generated by an estimator correlates with the annotation time provided in the respective datasets.

Evaluation Metrics. We evaluate all estimators by measuring Spearman's ρ between the true and generated orderings of all instances in the test data. We obtain the generated ordering by sorting instances according to the predicted annotation time. For our adaptive estimators that explicitly learn to predict the annotation time, we further report the mean absolute error (MAE), the rooted mean squared error (RMSE), and the coefficient of determination (R^2).

Models and Features. For an effective deployment in interactive annotation scenarios, we require models that are capable of fast training and inference. We additionally consider the amount of computational resources that a model requires as these pose further limitations for the underlying annotation platform. Consequently, fine-tuning large language models such as BERT is infeasible as they require long training times and a large amount of computational resources.[4] Instead, we utilize a combination of neural embeddings obtained from a large pre-trained language model combined with an efficient statistical model. As our goal is to predict the total time an annotator requires to annotate an instance (i.e., a sentence or a paragraph), we further require a means to aggregate token- or subtoken-level embeddings that are used in recent language models (Sennrich, Haddow, and Birch 2016). One such solution is S-BERT (Reimers and Gurevych 2019), which has shown high performance across various tasks. Moreover, Reimers and Gurevych (2019) provide S-BERT for a variety of BERT-based models, allowing future study conductors to easily extend our setup to other languages

[4] Note that using such models would require an annotation platform to either deploy its own GPU or buy additional computational resources from external providers.

Table 2
Hyperparameter tuning for adaptive estimators. We train on Muc7$_T$ A and evaluate on its development set. t denotes the total time for training and prediction on the whole dataset. Best parameters are marked by * and the best scores are highlighted in **bold**. We report the mean absolute error (MAE), the rooted mean squared error (RMSE), Spearman's ρ, and the coefficient of determination (R^2).

Name	Features	MAE	RMSE	R^2	ρ	t
RR($\alpha = 0.5$)	BOW	1.85	2.96	0.47	0.73	0.42
RR($\alpha = 0.5$)	S-BERT	1.92	2.84	0.51	0.79	**0.04**
RR($\alpha = 1$)	BOW	1.80	2.91	0.49	0.74	0.41
RR($\alpha = 1$) *	S-BERT	1.89	2.82	0.52	0.79	0.04
GP(kernel=Dot + White)	BOW	1.82	2.93	0.48	0.74	257.67
GP(kernel=Dot + White) *	S-BERT	**1.80**	**2.76**	**0.54**	**0.81**	14.35
GP(kernel=RBF(1.0)	BOW	5.33	6.71	−1.73	−0.12	300.38
GP(kernel=RBF(1.0)	S-BERT	5.33	6.71	−1.73	−0.12	32.66
GBM	BOW	2.07	3.26	0.36	0.68	0.25
GBM *	S-BERT	1.83	2.83	0.52	0.79	2.98

and specific tasks. For computational efficiency, we use the *paraphrase-distilroberta-base-v1* model, which utilizes a smaller, distilled RoBERTa model (Sanh et al. 2019). As a comparison to S-BERT, we further evaluate bag-of-words (BOW) features for all three models (cf. Table 2). For the Ridge Regression (RR), Gaussian Process Regression (GP), and GBM Regression (GBM) models, we use the implementations of Pedregosa et al. (2011) and Ke et al. (2017).

Hyperparameter Tuning. We use the full experimental setup to identify the best performing parameters for our experiments using simulated annotators. We evaluate different values for regularization strength (α) for RR and we evaluate different kernel functions for GP. To ensure that the required training of our adaptive estimators does not negatively affect the annotations due to increased loading times and can be realistically performed during annotation, we further measure the overall training time (in seconds). We use the development split of Muc7$_T$ A to tune our hyperparameters for all models used across all datasets. Considering the small number of training instances in both datasets, we do not tune SigIE- or SPEC-specific hyperparameters. All experiments were conducted using an *AMD Ryzen 5 3600*. Table 2 shows the results of our hyperparameter tuning experiments. Overall, we find that S-BERT consistently outperforms BOW in terms of Spearman's ρ. As the result of the hyperparameter tuning, we use S-BERT embeddings as input features and evaluate GP with a combined dot- and white-noise kernel and RR with $\alpha = 1$ in our adaptive experiments.

4.3 Experimental Results

We first report our experimental results for the full and adaptive setup. For conducting our experiments with simulated annotators, we use the best performing models from our hyperparameter tuning of the respective models on the Muc7$_T$ dataset and report the results of the best performing models.

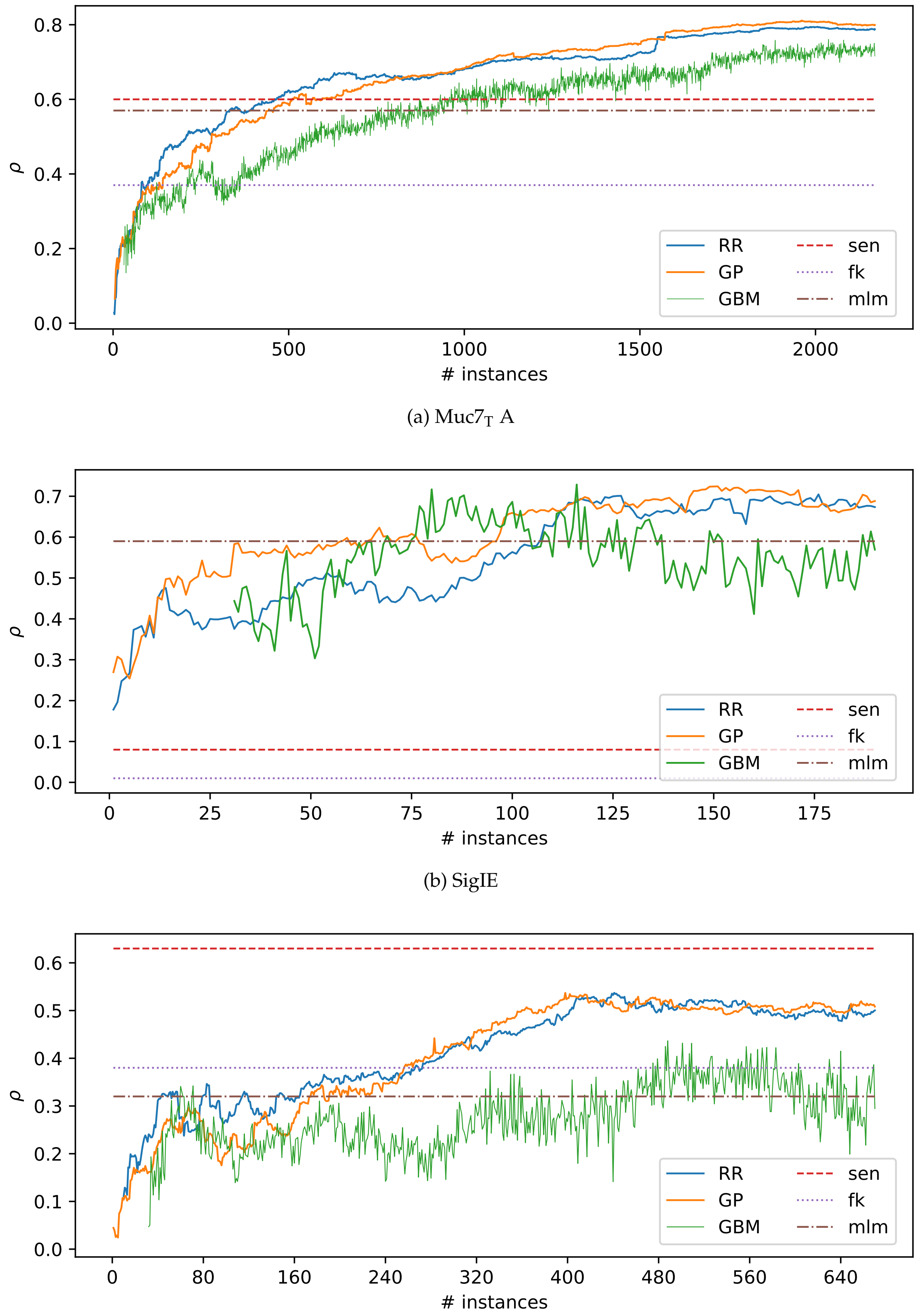
Table 3
Performance of the best performing adaptive estimators on the four datasets (Muc7$_\mathrm{T}$ provides annotation times from two different annotators A and B) trained on the respective train and evaluated on their test splits. We report the mean absolute error (MAE), the rooted mean squared error (RMSE), the coefficient of determination (R^2), and Spearman's ρ.

Name	Model	MAE	RMSE	R^2	ρ	t
Muc7$_\mathrm{T}$ A	RR	1.87	2.68	0.56	0.80	0.15
	GP	1.79	2.66	0.57	0.82	7.23
	GBM	1.95	2.97	0.47	0.75	3.40
Muc7$_\mathrm{T}$ B	RR	2.19	3.42	0.44	0.79	0.02
	GP	2.08	3.37	0.46	0.81	8.85
	LGBM	2.13	3.50	0.41	0.75	2.90
SigIE	RR	7.96	9.50	0.46	0.73	0.00
	GP	7.62	9.60	0.44	0.70	0.08
	GBM	8.22	10.84	0.29	0.55	0.14
SPEC	RR	9.63	13.86	−0.14	0.50	0.03
	GP	7.63	12.07	0.14	0.51	0.73
	GBM	8.05	12.50	0.07	0.35	1.70

Full Results. Table 4 shows the results for the heuristic estimators and regression models evaluated on the test split of each dataset. We find that heuristics that mainly consider length-based features (sen and FK) are not suited for the SigIE data that consist of email signatures. One reason for this may be the different text type of email signatures in comparison to Newswire articles and PubMed abstracts. More specifically, analyzing the ratio between non-alphabetical or numeric characters (excluding @ and .) and other characters shows that SigIE contains a substantial number of characters that are used for visually enhancing the signature (some are even used in text art). Overall, 29.9% of the characters in SigIE are non-alphabetical or numeric, in contrast to 16.7% in SPEC and 19.9% in Muc7$_\mathrm{T}$.[5] Considering that only 1.7% of them appear within named entities in SigIE (such as + in phone numbers) most of them rather introduce noise especially for length-based features such as sen and FK. On Muc7$_\mathrm{T}$ and SPEC, all three heuristics produce an ordering that correlates with annotation time to some extent. On average, mlm is the best performing and most robust heuristic across all three datasets. For our adaptive estimators, RR and GP both similarly outperform GBM in terms of Spearman's ρ. However, we can find that GP consistently outperforms RR and GBM in terms of MAE and RMSE, as well as in terms of R^2 on Muc7$_\mathrm{T}$ and SPEC. We report the extensive results in Table 3.

Adaptive Results. To evaluate the performance of adaptive estimators with increasing numbers of annotated instances, we perform experiments with simulated annotators. At each iteration, we use a model trained on the already-annotated data to select the instance with the lowest predicted annotation time (randomly in the first iteration). The simulated annotator then provides the respective gold annotation time, which is then added to the training set. Finally, the model is re-trained and evaluated on the test data. These steps are repeated until all instances are annotated. Figure 2 shows the Spearman's ρ performance of all three models after each iteration across all datasets. We

[5] The Twitter data we introduce in Section 5 consist of 20.7% non-alphabetical or numeric characters.



(a) Muc7$_T$ A

(b) SigIE

(c) SPEC

Figure 2
Experimental results of our adaptive estimators with simulated annotators. Horizontal lines show the performance of the respective non-adaptive estimators.

can observe that all models display a rather steep learning curve after training on only a few examples, despite suffering from a cold start in early iterations. Moreover, we find that GP and RR are capable of outperforming mlm consistently after 100–500 instances. GBM shows the weakest performance and is consistently outperformed by the other models for Muc7$_T$ and SPEC while being rather noisy. Although we find that non-adaptive estimators can suffice, especially in early iterations, our experiments also show the potential of adaptive estimators with an increasing number of annotations. This indicates that hybrid approaches that combine non-adaptive and adaptive estimators could be an interesting direction for future work. For instance, one may consider using non-adaptive estimators in early stages until a sufficient number of annotated instances are available to train more reliable adaptive estimators. Another approach could be to combine the rankings of different estimators, for instance, via Borda count (Szpiro 2010) or learn a weighting of the individual estimators.

5. Human Evaluation

To evaluate the effectiveness of our easy-instances-first AC with real annotators, we conduct a user study on a classification task for English tweets and analyze the resulting annotations in terms of *annotation time* and *annotation quality*. We design the study to not require domain-expertise and conduct it with citizen science volunteers.[6]

Hypothesis. We investigate the following hypothesis: Annotators who are presented with easy instances first and then with instances that gradually increase in terms of annotation difficulty require less annotation time or have improved annotation quality compared with annotators who receive the same instances in a random order.

5.1 Study Design

A careful task and data selection are essential to evaluate AC, as our goal is to measure differences that solely result from a different ordering of annotated instances. We also require instances with varying difficulty, further restricting our study design in terms of task and data.

Data Source. To avoid compromising the study results due to noisy data, we use an existing corpus that has been carefully curated and provides gold labels for evaluating the annotation quality. To involve non-expert annotators, we further require data that do not target narrow domains or require expert knowledge. As such, tasks such as identifying part-of-speech tags would substantially reduce the number of possible study participants due to the required linguistic knowledge. We identify COVIDLies (Hossain et al. 2020) as a suitable corpus due to the current relevance and the high media-coverage of the Covid-19 pandemic; ensuring a sufficient number of participants who are well-versed with the topic. The corpus consists of English tweets that have been annotated by medical experts with one out of 86 common misconceptions about the Covid-19 pandemic. Each instance consists of a tweet-misconception pair and if the tweet “agrees,” “disagrees,” or has “no stance” toward the presented misconception.

[6] We provide a statement regarding the conduct of ethical research after the conclusion.

Annotation Task. Using the COVIDLies corpus as our basis, we define a similar task that is better suited for lay people and that allows us to explicitly control the annotation difficulty. We restrict the task to identifying the most appropriate misconception out of six possible choices. Furthermore, we only include tweets that agree with a misconception (i.e., we do not ask for a stance annotation) to avoid interdependencies between stance and misconception annotations that may introduce additional noise to the results and put an unnecessary burden on the participants.[7] To exclude further sources of noise for our study, we manually check all tweets and remove all duplicates (possibly due to retweets) and hyperlinks to increase readability and avoid distractions. We also remove all tweets that were malformed (i.e., ungrammatical or containing several line breaks) or linked to misconceptions with less than five semantically similar candidates that could serve as distractors.[8] For the final selection, we choose the 60 shortest tweets.

Distractor Selection. The goal of the study is to observe effects that solely result from the ordering of instances with varying annotation difficulty. Hence, we need to ensure that annotated instances correspond to specific difficulties and are balanced equally for each participant. To control the annotation difficulty, we construct five possible sets of misconceptions for each instance that are presented to the annotator; each corresponding to a respective difficulty-level ranging from "very easy" to "very difficult." Each set consists of the expert-selected misconception and five additional misconceptions that serve as distractors which are commonly used in cloze-tests (Taylor 1953). Following existing research on automated cloze-test generation, we focus on **semantic similarity** to generate distractor subsets (Agarwal and Mannem 2011; Mostow and Jang 2012; Yeung, Lee, and Tsou 2019) and manually create one set of five semantically dissimilar and one set of five semantically similar misconceptions for each misconception.[9] As semantically dissimilar distractors are much easier to identify than semantically similar ones (Mostow and Jang 2012), we can manipulate annotation difficulty by adapting the number of semantically similar distractors; that is, starting from the set of dissimilar (very easy) misconceptions, we can gradually increase the difficulty by replacing a dissimilar misconception with a similar one until only the set of similar (very difficult) misconceptions remains. Figure 3 shows a tweet from our user study with its respective easy and difficult misconception sets. As can be seen, the difficult misconception set consists of two more semantically similar misconceptions. Especially notable is the third misconception, which states the opposite of the tweet's misconception but with a similar wording.

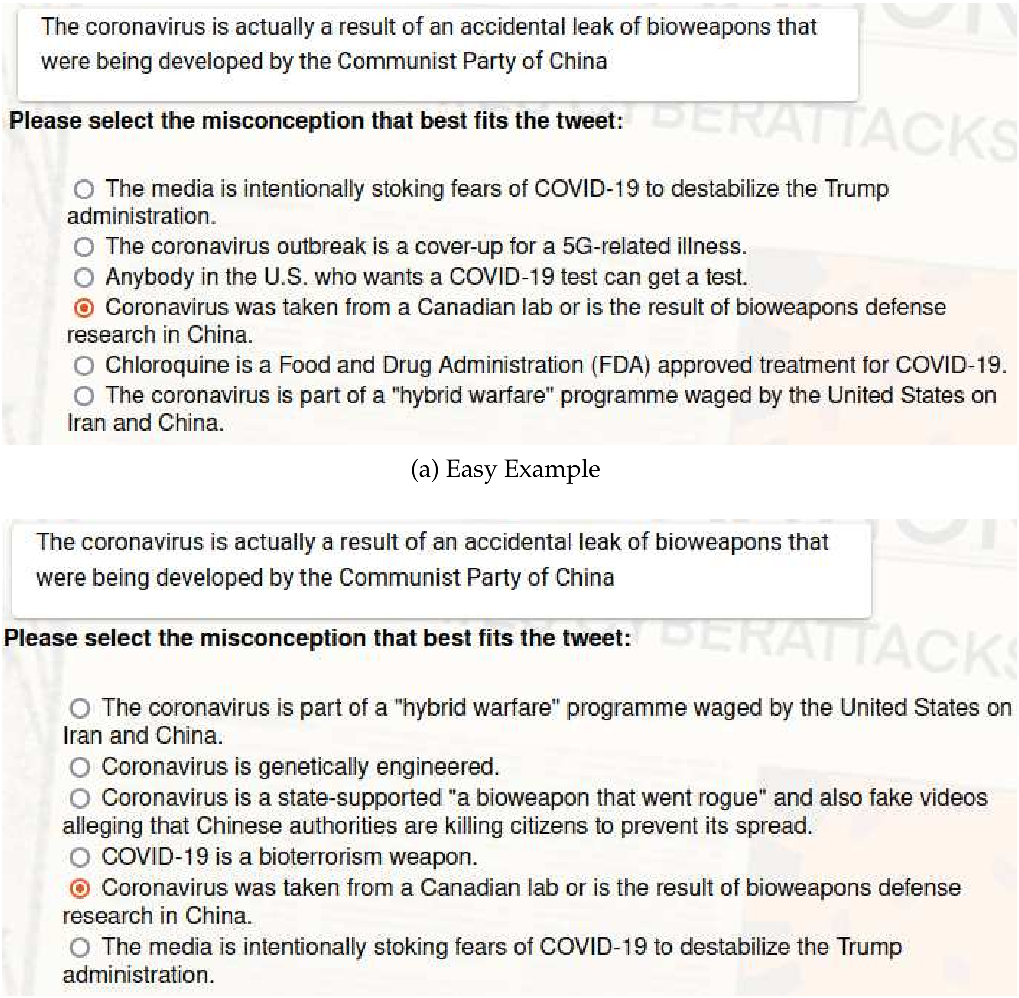
5.2 Study Setup

We set up our evaluation study as a self-hosted Web application that is only accessible during the study (one week). Participants can anonymously participate with a self-chosen, unique study key that allows them to request the deletion of their provided data at a later point. Upon registration, they are informed about the data presented and

[7] We experimented with including stance annotations (positive, negative, or neutral) during early stages of our study setup but removed them due to a substantially increased overall annotation difficulty.

[8] The sets of similar misconceptions were manually created as explained in the next paragraph.

[9] Initially, we also investigated the use of recent automated approaches to create those subsets (Gao, Gimpel, and Jensson 2020). However, the resulting subsets rather targeted syntactic instead of semantic similarity. One reason for this may be that approaches to generate cloze-tests consider only single-token gaps whereas the misconceptions consist of several words that form a descriptive statement.



The coronavirus is actually a result of an accidental leak of bioweapons that were being developed by the Communist Party of China

Please select the misconception that best fits the tweet:

○ The media is intentionally stoking fears of COVID-19 to destabilize the Trump administration.
○ The coronavirus outbreak is a cover-up for a 5G-related illness.
○ Anybody in the U.S. who wants a COVID-19 test can get a test.
● Coronavirus was taken from a Canadian lab or is the result of bioweapons defense research in China.
○ Chloroquine is a Food and Drug Administration (FDA) approved treatment for COVID-19.
○ The coronavirus is part of a "hybrid warfare" programme waged by the United States on Iran and China.

(a) Easy Example

The coronavirus is actually a result of an accidental leak of bioweapons that were being developed by the Communist Party of China

Please select the misconception that best fits the tweet:

○ The coronavirus is part of a "hybrid warfare" programme waged by the United States on Iran and China.
○ Coronavirus is genetically engineered.
○ Coronavirus is a state-supported "a bioweapon that went rogue" and also fake videos alleging that Chinese authorities are killing citizens to prevent its spread.
○ COVID-19 is a bioterrorism weapon.
● Coronavirus was taken from a Canadian lab or is the result of bioweapons defense research in China.
○ The media is intentionally stoking fears of COVID-19 to destabilize the Trump administration.

(b) Difficult Example

Figure 3
Example tweet from the user study with an easy misconception set (used in the study) and a difficult misconception set.

collected in the study, its further use, and the purpose of the study. Before collecting any data, participants are explicitly asked for their informed consent. Overall, we recruited 40 volunteers who provided their informed consent to participate in our study and annotated 60 instances each.

Participants. Our volunteers come from a variety of university majors, native languages, English proficiency, and annotation experience backgrounds. All participants provided a rather high self-assessment of English proficiency, with the lowest proficiency being intermediate (B1) provided by only one participant. Seventy percent of the participants stated an English proficiency-level of advanced (C1) or proficient (C2). Most participants have a higher level of education and are university graduates with either a Bachelor's or Master's degree; however, none of them have a medical background, which may have given them an advantage during the annotation study. Upon completing the annotations, all participants received a questionnaire including general questions about their previous annotation experience and perceived difficulty of the task (cf. Section 5.5).

Table 4
Spearman's ρ between test data and the orderings generated by the evaluated heuristics and adaptive models.

Dataset	sen	FK	mlm	RR	GP	GBM
Muc7$_T$ A	0.60	0.37	0.57	0.80	**0.82**	0.75
Muc7$_T$ B	0.60	0.38	0.55	0.79	**0.81**	0.75
SigIE	0.08	0.01	0.59	**0.73**	0.70	0.55
SPEC	**0.63**	0.38	0.32	0.50	0.51	0.35
Average	0.48	0.29	0.52	0.71	0.71	0.60

Ordering Strategy. All participants are randomly assigned to one out of four groups (ten participants per group), each corresponding to a strategy that leads to a different ordering of annotated instances. We investigate the following strategies:

Random is the control group that consists of randomly ordered instances.

AC$_{\mathbf{mlm}}$ uses the masked language modeling loss. It is a pre-computed, heuristic estimator and had (on average) the highest and most stable correlation to annotation time in our experiments with simulated annotators.

AC$_{\mathbf{GP}}$ uses a Gaussian Process that showed the highest performance on the sentence-labeling task (SPEC) in our simulated annotator experiments (cf. Table 4). It is trained interactively to predict the annotation time. We train a personalized model for each annotator using S-BERT embeddings of the presented tweet.

AC$_{\mathbf{gold}}$ consists of instances explicitly ordered from very easy to very difficult using the pre-defined distractor sets. Although such annotation difficulties are unavailable in real-world annotation studies, it provides an upper-bound for the study.

Control Instances. To provide a fair comparison between different groups, we further require participants to annotate instances that quantify the difference with respect to prior knowledge and annotation proficiency. For this, we select the first ten instances and present them in the same order for all annotators. To avoid interdependency effects between the control instances and the instances used to evaluate AC$_{\{*\}}$, we selected instances that have disjoint sets of misconceptions.

Balancing Annotation Difficulty. We generate instances of different annotation difficulties using the sets of semantically similar and dissimilar misconceptions that serve as our distractors. We randomly assign an equal number of tweet-misconception pairs to each difficulty-level ranging from very easy to very difficult. The resulting 50 instances for our final study span similar ranges in terms of length, as shown in Table 5, which is crucial to minimize the influence of reading time on our results. Overall, each of the five difficulty-levels consists of ten (two for the control instances) unique tweets that are annotated by all participants in different order.

Study Process. The final study consists of 50 instances that are ordered corresponding to the group a participant has been assigned to. Each instance consists of a tweet and six possible misconceptions (one expert-annotated and five distractors) from which the

Table 5
Average number of characters per tweet (T) and tweet and misconception (T & MC) across all difficulty-levels of annotated items.

# Chars	very easy	easy	medium	difficult	very difficult
T	219	211	183	217	194
T & MC	638	603	599	586	593

Table 6
Mean, standard deviation, and 25%, 50%, and 75% percentiles of annotation (in seconds). Σ_t denotes the total annotation time an annotator of the respective group requires to finish the study (on average).

	Σ_t	μ_t	σ_t	25%	50%	75%
Random	1,852.9	27.3	27.2	12.9	18.2	29.5
$\mathrm{AC}_{\mathrm{mlm}}$	1,273.4	23.2	19.4	**11.7**	18.6	27.4
$\mathrm{AC}_{\mathrm{GP}}$	1,324.3	26.4	19.0	14.9	20.7	30.8
$\mathrm{AC}_{\mathrm{gold}}$	**1,059.6**	**21.2**	**12.8**	12.6	**18.0**	**26.5**

participants are asked to select the most appropriate one. The lists of the six presented misconceptions are ordered randomly to prevent that participants learn to annotate a specific position. Finally, we ask each participant to answer a questionnaire that measures the perceived difficulty of the annotated instances.

5.3 General Results

In total, each of the 40 participants has provided 60 annotations, resulting in 400 annotations for the ten control instances (100 per group) and 2,000 annotations for the 50 final study instances (500 per group). In terms of annotation difficulty, each of the five difficulty-levels consists of 80 annotations for the control instances and 400 annotations for the final study. To assess the validity of $\mathrm{AC}_{\{*\}}$, we require two criteria to be fulfilled:

H1 The participant groups do not significantly differ in terms of annotation time or annotation quality for the control instances.

H2 $\mathrm{AC}_{\{*\}}$ shows a significant difference in annotation time or annotation quality compared to Random or each other.

Outliers. Across all 2,400 annotations, we identify only two cases where participants required more than ten minutes for annotation and are apparent outliers. To avoid removing annotations for evaluation, we compute the mean and standard deviation of the annotation time across all annotations (excluding the two outliers) and set the maximum value to $t_{\max} = \mu + 5\sigma = 156.39$ seconds. This results in ten annotations that are set to $t_{\max}$ for Random, three for $\mathrm{AC}_{\mathrm{mlm}}$, one for $\mathrm{AC}_{\mathrm{GP}}$, and zero for $\mathrm{AC}_{\mathrm{gold}}$. Note that this mainly favors the random control group that serves as our baseline.

Annotation Time. Table 6 shows the results of the final study in terms of annotation time per group. Overall, annotators of $\mathrm{AC}_{\mathrm{gold}}$ required on average the least amount of

Table 7
Mean, standard deviation, and 25%, 50%, and 75% percentiles of annotation quality (in percent accuracy).

	μ_{acc}	σ_{acc}	25%	50%	75%
Random	84.7	4.22	82.0	**86.0**	**88.0**
AC_{mlm}	83.6	5.32	80.0	84.0	86.0
AC_{GP}	83.6	**2.95**	82.0	**86.0**	86.0
AC_{gold}	**85.6**	3.01	**84.0**	84.0	**88.0**

time per instance and had the lowest standard deviation. We also observe a substantial decrease in the maximum annotation time, as shown in the 75th percentile for AC_{gold}. Conducting a Kruskal–Wallis test (Kruskal and Wallis 1952) on the control instances across all participant groups results in a p-value of $p = 0.200 > 0.05$.[10] Hence, we cannot reject the null-hypothesis for the control instances, and conclude that all groups initially do not show statistically significant differences in terms of annotation time for the control instances, thereby satisfying H1. Next, we conduct the same test on the evaluation instances and observe a statistically significant p-value of $p = 4.53^{-6} < 0.05$. For a more specific comparison, we further conduct pairwise Welch's t-test (Welch 1951) for each strategy with a Bonferroni-corrected p-value of $p = \frac{0.05}{6} = 0.008\overline{3}$ to account for multiple comparisons (Bonferroni 1936). Overall, AC_{gold} performs best, satisfying H2 with statistically significant improvements over Random ($p = 7.28^{-6}$) and AC_{GP} ($p = 3.79^{-7}$). Although the difference to AC_{mlm} is substantial, it is not statistically significant ($p = 0.0502$). The best performing estimator is AC_{mlm}, which performs significantly better than Random ($p = 0.0069$) and substantially better than AC_{GP} ($p = 0.0084$). Between AC_{GP} and Random, we cannot observe any statistically significant differences ($p = 0.5694$).

Annotation Quality. We evaluate annotation quality by computing the accuracy for each participant, that is, the percentage of misconceptions that they were able to correctly identify out of the six presented ones. Table 7 shows our results in terms of accuracy. Although AC_{gold} has the highest mean accuracy, the most differences lie within the range of 2% accuracy, which is equivalent to only a single wrongly annotated instance. Conducting Kruskal–Wallis tests for the control instances shows that the difference in terms of accuracy is not statistically significant ($p = 0.881$), satisfying H1. However, the same test shows no statistically significant difference for the final study ($p = 0.723$). One reason for this may be our decision to conduct the study with voluntary participants and their higher intrinsic motivation to focus on annotation quality over annotation time (Chau et al. 2020). In contrast to crowdsourcing scenarios where annotators are mainly motivated by monetary gain—trying to reduce the amount of time they spend on their annotation at the cost of quality—voluntary annotators are more motivated to invest additional time to provide correct annotations; even more so in a setup with a low number of 60 instances.

[10] In general, ANOVA (analysis of variance) is a more expressive test that does not require pairwise comparisons that are necessary for the less expressive Kruskal–Wallis test. However, we cannot apply ANOVA in our case due to violated conditions on normality and homoscedasticity of the collected data.

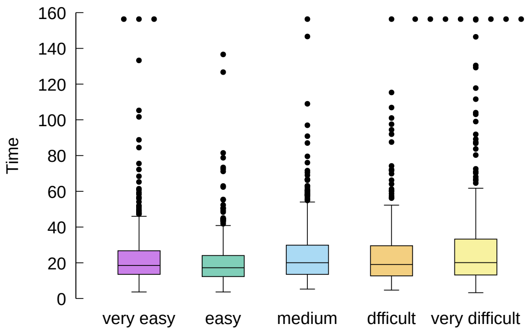


Figure 4
Annotation time (in seconds) grouped by difficulty level.

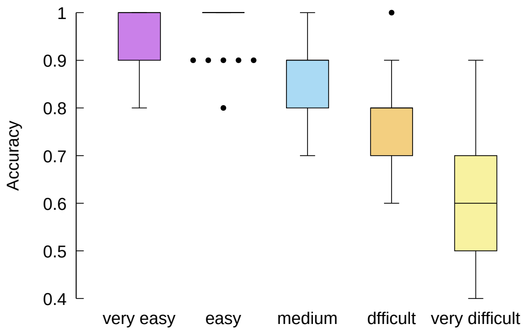
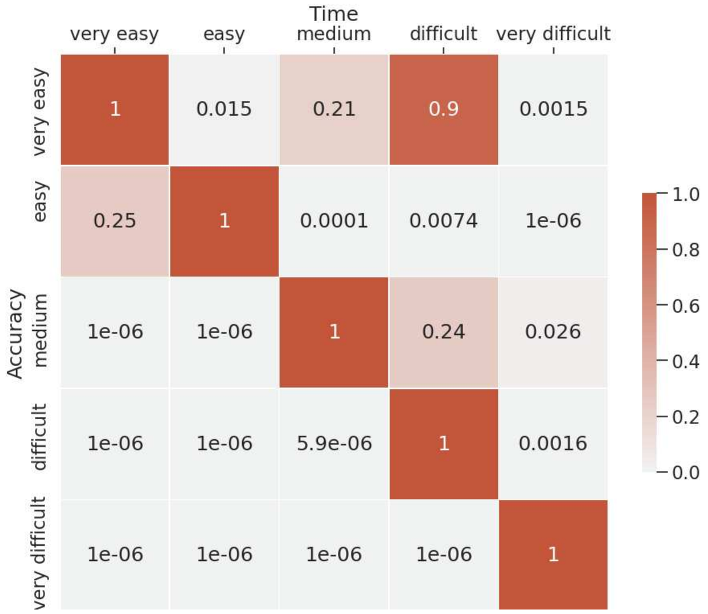


Figure 5
Accuracy per annotator grouped by difficulty level.



	very easy	easy	medium	difficult	very difficult
very easy	1	0.015	0.21	0.9	0.0015
easy	0.25	1	0.0001	0.0074	1e-06
medium	1e-06	1e-06	1	0.24	0.026
difficult	1e-06	1e-06	5.9e-06	1	0.0016
very difficult	1e-06	1e-06	1e-06	1e-06	1

Figure 6
The p-values for time (in seconds) and accuracy between different difficulty levels.

Difficulty Evaluation. To validate our generation approach with distractors, we further evaluate all annotation instances in terms of their annotation difficulty. As Figures 4 and 5 show, one can observe non-negligible differences in terms of annotation time as well as accuracy across instances of different difficulties. Conducting pairwise Welch's t-tests with a Bonferroni corrected p-value of $p = \frac{0.05}{10} = 0.005$ shows that in terms of accuracy, only very easy and easy instances do not express a statistically significant difference ($p = 0.25$), showing that participants had more trouble in identifying the correct misconception for difficult instances.[11] For all other instances, we observe p-values smaller than 1e^{-6}, as shown in Figure 6. In terms of annotation time, the differences are not as apparent as in annotation accuracy. We find statistically significant differences in only four out of ten cases showing that the annotation difficulty does not necessarily impact the annotation time. Overall, we still observe that instances express

[11] Overall, we require $\frac{n(n-1)}{2}$ pairwise comparisons, resulting in 10 comparisons with $n = 5$.

significant differences in terms of either annotation time or quality (or both), showing that our approach using distractor sets to control the annotation difficulty worked well.

5.4 Error Analysis

Model Performance. While AC_{mlm} and AC_{gold} both outperform the random baseline significantly, AC_{GP} does not. To analyze how well the used GP model performs for individual annotators, we perform leave-one-user-out cross validation experiments across all 40 participants. Table 8 shows the MAE, RMSE, the coefficient of determination (R^2), and Spearman's ρ of our experiments. Overall, we find a low correlation between the predicted and true annotation time and high standard deviations across both errors. Further analyzing the performance of AC_{GP} for interactively predicting the annotation time (cf. Figure 7) shows that the model adapts rather slowly to additional data. As can be observed, the low performance of the model (MAE between 10 and 20 seconds) results in a high variation in the annotation time of the selected instances between subsequent iterations; further experiments strongly suggest this is due to the model suffering from a cold start and the small amount of available training data as also discussed below.

Correlation with AC_{gold}. A second shortcoming of AC_{GP} becomes apparent when observing the difficulty of the sampled instances across all iterations, shown in Figure 8. We observe a low Spearman's ρ correlation to AC_{gold} of 0.005, in contrast to AC_{mlm} ($\rho = 0.22$). Only Random has a lower correlation, of $\rho = -0.15$. This shows that model adaptivity plays an important role, especially in low-data scenarios such as in early

Table 8
Leave-one-out cross validation results on annotation times, grouped by user and averaged.

	μ_t	σ_t	25%	50%	75%
MAE	12.4	6.1	8.5	10.4	14.3
RMSE	17.2	9.1	11.1	13.9	20.3
R^2	0.0	0.0	−0.1	0.0	0.0
ρ	−0.1	0.2	−0.3	−0.1	0.1

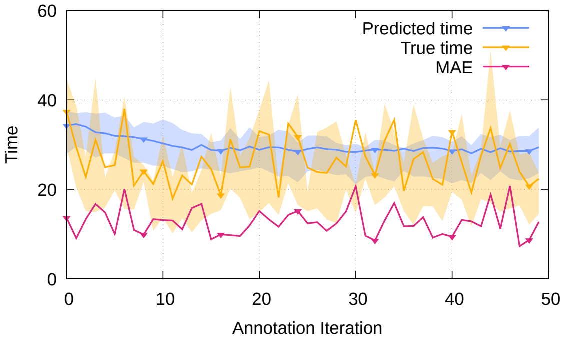


Figure 7
Mean, lower, and upper percentiles for predicted and true annotation time and the mean absolute error.

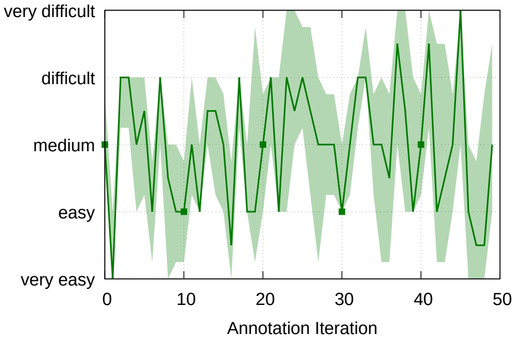


Figure 8
Median, lower, and upper percentile for instance difficulty with AC_{GP} at each iteration.

Table 9
Spearman's ρ correlation analysis for three potential confounding factors.

	CEFR		Annotator		Conductor	
	ρ	p-value	ρ	p-value	ρ	p-value
Time	−0.307	0.054	−0.134	0.409	0.085	0.600
Accuracy	0.319	0.044	−0.060	0.711	−0.211	0.191

stages during annotation studies. We plan to tackle this issue in future work using more sophisticated models and combined approaches that initially utilize heuristics and switch to interactively trained models with the availability of sufficient training data.

5.5 Participant Questionnaire

After completing the annotation study, each participant answered a questionnaire quantifying their language proficiency, previous annotation experience, and perceived difficulty of the annotation task.

Language Proficiency. In addition to their CEFR language proficiency (Council of Europe. 2001), we further asked participants to provide optional information about their first language and the number of years they have been actively speaking English. On average, our participants have been actively speaking English for more than 10 years. Overall, they stated a language proficiency of: B1 (1), B2 (11), C1 (17), and C2 (11). Most of our participants stated German as their first language (30). Other first languages include Vietnamese (4), Chinese (3), Russian (1), and Czech (1).[12]

Annotation Experience. We further collected data from our participants regarding their previous experience as study participants as well as study conductors. In general, about 50% of our participants (18) had not participated in annotation studies before. Nineteen had participated in a few (one to three) studies, and only three in more than three studies. Even more participants had not previously conducted a study (24) or only a few (12). In total, four participants stated that they had set up more than three annotation studies.

Confounding Factors. We identify the language proficiency and previous experience with annotation studies as potential confounding factors (VanderWeele and Shpitser 2013). Confounding factors are variables that are difficult to control for, but have an influence on the whole study and can lead to a misinterpretation of the results. Especially in studies that include a randomized setup such as in ours—due to the random assignment of our participants into the four groups—it is crucial to investigate the influence of potential confounding factors. In our analysis, we focus on variables for which all participants provided an answer, namely, their CEFR level and their experience as participants in and conductors of annotation studies (some of our participants were researchers). Table 9 shows the results of a Spearman's ρ correlation analysis for all three variables against annotation time and accuracy. As we can see, the participants'

[12] One participant decided not to disclose any additional information except English proficiency.

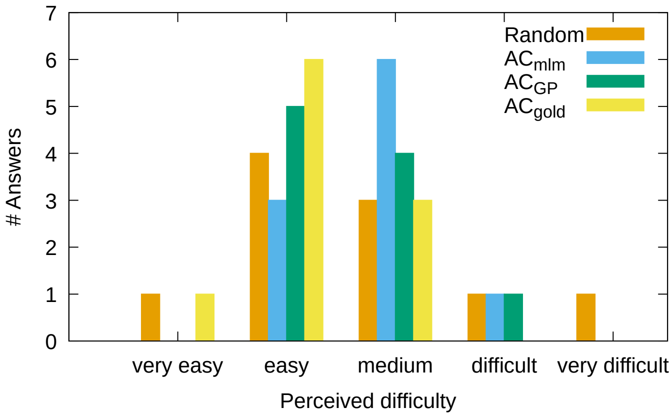


Figure 9
Accumulated perceived difficulty answers across all groups.

experiences as annotators (Annotator) or study conductors (Conductor) only yields a low, non-significant correlation with time and accuracy and, consequently, can be excluded as confounding factors. The influence of their language proficiency (CEFR) is more interesting, as it shows a small negative correlation for annotation time and a small positive correlation for annotation accuracy with p-values around 0.05, meaning that participants with a lower CEFR level required less time, but also had a lower accuracy. To investigate the influence of a participant's language proficiency on our results, we conduct a Kruskal–Wallis test for the distribution of different language proficiency levels across the four groups and find that they do not differ significantly with a p-value of $p = 0.961$. Nonetheless, we find that the CEFR level is an important confounding factor that needs to be considered in future study setups.

Perceived Difficulty. To quantify if there exists any difference between the actual difficulty and the perceived difficulty, we further asked our participants the following questions:

PQ1: How difficult did you find the overall annotation task?

PQ2: Did you notice any differences in difficulty between individual tweets?

PQ3: Would you have preferred a different ordering of the items?

Figure 9 shows the distribution of answers (from very easy to very difficult) to PQ1 across all four groups. Interestingly, whereas participants of the AC_{mlm} group did require less time during their annotation compared with AC_{GP}, more people rated the study as of medium difficulty than participants of AC_{GP}. This may be an indicator that AC_{GP} may—although not measurable in terms of annotation time—alleviate the perceived difficulty for participants, hence, still reducing the cognitive burden. We will investigate this in further studies that also include an item specific difficulty annotation, that is, by explicitly asking annotators for the perceived difficulty.[13] Overall, only four out of 40 participants (two for AC_{GP} and one for AC_{mlm} and AC_{gold} each) did state to not have noticed any differences in terms of difficulty between different instances; showing that the selected distractors resulted in instances of noticeably different annotation

[13] We excluded this additional annotation in the study as one pass already required $\sim$ 45–60 minutes.

difficulty (PQ2). For PQ3, we find that 33 participants did not wish for a different ordering of instances (but were still allowed to provide suggestions), four would have preferred an "easy-first," one a "difficult-first," and two an entirely different ordering strategy. From the 14 free-text answers and feedback via other channels, we identify three general suggestions that may be interesting for future research:

S1: Grouping by word rarity.

S2: Grouping instances by token overlap.

S3: Grouping instances by topic (tweet or alternatively, misconception) similarity.

Further analyzing the free-text answers together with the pre-defined answers ("no," "easy-first," "difficult-first," and "other") shows that the participants disagree on the preferred ordering strategy. For instance, the participants that suggested S3, disagreed if instances should be grouped by topic similarity to reduce the number of context switches or be as diverse as possible to provide some variety during annotation. Another five participants (two from Random and one from the other groups each) even explicitly supported a random ordering in the free-text answer. The disagreement upon the ordering strategy shows the importance of interactively trained estimators that are capable of providing personalized annotation curricula.

6. Limitations and Future Work

We evaluated AC with an easy-instances-first strategy in simulations as well as in a highly controlled setup using a finite, pre-annotated data set and task-agnostic estimators to minimize possible noise factors. To demonstrate the viability of AC with a sufficient number of voluntary annotators, we further chose a dataset that covers a widely discussed topic and manually controlled the annotation difficulty to make it accessible for non-experts. To evaluate AC with more generalizable results in a real-world scenario, we discuss existing limitations that should be considered beforehand that can also serve as promising research directions for future work.

Difficulty Estimators. Due to novelty of the proposed approach and the lack of well-established baselines, we focused on task-agnostic annotation difficulty estimators such as reading difficulty and annotation time, which can easily be applied to a wide range of tasks. Although our study results show that they work to some extent, our evaluation with existing datasets also shows that especially non-adaptive estimators, which approximate the absolute task-difficulty, are sensitive to the data domain and annotation task (cf. the low performance of length-based estimators on the SigIE data in Section 4). Such issues could be addressed by implementing estimators that are more *task-specific*. For named entity annotations, a general improvement may be achieved by considering the number of nouns within a sentence that can be obtained from a pre-trained part-of-speech tagger. One may even consider domain-specific word frequency lists to provide a difficulty estimate for entities. For instance, among the annotated named entities in Muc7$_T$, "U.S." (occurs 72 times) may be easier to annotate than "Morningstar" (occurs only once); simply based on a word frequency analysis. Other, more sophisticated approaches from educational research such as item response theory (Baker 2001) and scaffolding (Jackson et al. 2020) may also lead to better task-agnostic estimators. Such approaches and combinations of task-agnostic with task-specific estimators remain to be investigated in future work.

Annotation Strategies. In this work, we focused on developing and evaluating a strategy for our non-expert annotation scenario. Although it proved to be effective in our user study, we also find that our annotators disagree in their preferences with respect to the ordering of instances—which indicates that investigating *annotator-specific* strategies could be a promising line for future work. Another shortcoming of the evaluated strategy is that it does not consider an annotator's boredom or frustration (Vygotsky 1978). Especially when considering larger annotation studies, motivation may become an increasingly important factor with non-expert annotators as they further progress in a task and become more proficient. Such a strategy may also be better suited for annotation scenarios that involve domain experts to retain a high motivation by avoiding boredom—for instance, by presenting them with subsequent instances of varying difficulty or different topics. Domain experts who do not require a task-specific training may also benefit from strategies that focus on familiarizing them with the data domain early on to provide them with a good idea of what kind of instances they can expect throughout their annotations. To implement strategies that consider annotator-specific factors such as motivation and perceived difficulty, adaptive estimators may have an advantage over non-adaptive ones as they can incorporate an annotator's preference on the fly. We will investigate more sophisticated adaptive estimators (also coupled with non-adaptive ones) and strategies in future work and also plan to evaluate AC with domain expert annotators.

Larger Datasets. While using a finite set of annotated instances was necessary in our user study to ensure a proper comparability, AC is not limited to annotation scenarios with finite sets. However, deploying AC in scenarios that involve a large number of unlabeled instances requires additional consideration besides an annotator's motivation. In scenarios that only annotate a subset of the unlabeled data (similar to pool-based active learning), an easy-instances-first strategy may lead to a dataset that is imbalanced toward instances that are easy to annotate. This can hurt data diversity and consequently result in models that do not generalize well to more difficult instances. To create more diverse datasets, one may consider introducing a stopping criterion (e.g., a fixed threshold) for the annotator training phase and moving on to a different sampling strategy from active learning. Other, more sophisticated approaches would be to utilize adaptive estimators with a pacing function (Kumar, Packer, and Koller 2010) or sampling objectives that jointly consider annotator training and data diversity (Lee, Meyer, and Gurevych 2020). Such approaches are capable of monitoring the study progress and can react accordingly, which may result in more diverse datasets. However, they also face additional limitations in terms of the computational overhead that may require researchers to consider an asynchronous model training in their setup.

Implementation Overhead. Finally, to apply AC in real-world annotation studies, one needs to consider the additional effort for study conductors to implement it. Whereas the task-agnostic estimators we provide can be integrated with minimal effort, developing task- and annotator-specific estimators may not be a trivial task and requires a profound knowledge about the task, data, and annotators. Another open question is how well the time saving of approximately 8–13 minutes per annotator in our study translates to large-scale annotation studies. If so, then AC could also be helpful in annotation studies with domain experts by resulting in more annotated instances within a fixed amount of time—however, if not, this would simply lead to a trade-off between the time investment of the study conductor and annotators. Overall, we find that developing and evaluating further strategies and estimators to provide study conductors with a wide range of choices to consider for their annotation study will be an interesting task for the research community.

7. Conclusion

With annotation curricula, we have introduced a novel approach for implicitly training annotators. We provided a formalization for an easy-instances-first strategy that orders instances from easy to difficult by approximating the annotation difficulty with task-agnostic heuristics and annotation time. In our experiments with three English datasets, we identified well-performing heuristics and interactively trained models and find that the data domain and the annotation task can play an important role when creating an annotation curriculum. Finally, we evaluate the best performing heuristic and adaptive model in a user study with 40 voluntary participants who classified English tweets about the Covid-19 pandemic and show that leveraging AC can lead to a significant reduction in annotation time while preserving annotation quality.

With respect to our initial research questions (cf. Section 1), our results show that the order in which instances are annotated can have a statistically significant impact in terms of annotation time (RQ1) and that recent language models can provide a strong baseline to pre-compute a well-performing ordering (RQ2). We further find that our interactively trained regression models lack adaptivity (RQ3), as they perform well on existing datasets with hundreds or more training instances, but fall behind non-adaptive estimators in the user study.

We conclude that annotation curricula provide a promising way for more efficient data acquisition in various annotation scenarios—but that they also need further investigation with respect to task-specific estimators for annotation difficulty, annotator-specific preferences, and applicability on larger datasets. Our analysis of existing work shows that, unfortunately, the annotation ordering as well as annotation times are seldomly reported. In the face of the increasing use of AI models in high-stake domains (Sambasivan et al. 2021) and the potentially harmful impact of biased data (Papakyriakopoulos et al. 2020), we ask dataset creators to consider including individual annotation times and orderings along with a datasheet (Gebru et al. 2021) when publishing their dataset. To facilitate future research, we share all code and data and provide a ready-to-use and extensible implementation of AC in the INCEpTION annotation platform.[14]

Acknowledgments
This work has been supported by the European Regional Development Fund (ERDF) and the Hessian State Chancellery – Hessian Minister of Digital Strategy and Development under the promotional reference 20005482 (TexPrax) and the German Research Foundation under grant nos. EC 503/1-1 and GU 798/21-1 (INCEpTION). We thank Michael Bugert, Richard Eckart de Castilho, Max Glockner, Ulf Hamster, Yevgeniy Puzikov, Kevin Stowe, and the anonymous reviewers for their thoughtful comments and feedback, as well as all anonymous participants in our user study.

[14] `https://inception-project.github.io/`.

Ethics Statement

Informed Consent. Participants of our user study participated voluntarily and anonymously with a self-chosen, unique study key that allows them to request the deletion of their provided data at a later point. Upon registration, they are informed about the data presented and collected in the study, its further use, and the purpose of the study. Before collecting any data, participants are explicitly asked for their informed consent. We do not collect any personal data in our study. If participants do not provide their informed consent, their study key is deleted immediately. For publication, the study key is further replaced with a randomly generated user id.

Use of Twitter Data. The CovidLies corpus (Hossain et al. 2020) we used to generate the instances for our annotation study consists of annotated tweets. To protect the anonymity of the user who created the tweet, we only display the text (removing any links) without any metadata like Twitter user id or timestamps to our study participants. We only publish the tweet ids in our study data to conform with Twitter's terms of service and hence, all users retain their right to delete their data at any point.

References

Agarwal, Manish and Prashanth Mannem. 2011. Automatic gap-fill question generation from text books. In *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–64.

Ash, Jordan T., Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, pages 1–26.

Baker, Frank. 2001. *The Basics of Item Response Theory*. ERIC Clearinghouse on Assessment and Evaluation, College Park, MD.

Beck, Tilman, Ji-Ung Lee, Christina Viehmann, Marcus Maurer, Oliver Quiring, and Iryna Gurevych. 2021. Investigating label suggestions for opinion mining in German Covid-19 social media. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–13. https://doi.org/10.18653/v1/2021.acl-long.1

Beigman Klebanov, Beata and Eyal Beigman. 2014. Difficult cases: From data to learning, and back. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 390–396. https://doi.org/10.3115/v1/P14-2064

Beinborn, Lisa, Torsten Zesch, and Iryna Gurevych. 2014. Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–529. https://doi.org/10.1162/tacl_a_00200

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48. https://doi.org/10.1145/1553374.1553380

Bonferroni, Carlo. 1936. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. https://doi.org/10.18653/v1/D15-1075

Chau, Hung, Saeid Balaneshin, Kai Liu, and Ondrej Linda. 2020. Understanding the tradeoff between cost and quality of expert annotations for keyphrase extraction. In *Proceedings of the 14th Linguistic Annotation Workshop*, pages 74–86.

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Council for Cultural Co-operation. Education Committee. Modern Languages Division. Cambridge University Press, Strasbourg, France.

Deutsch, Tovly, Masoud Jasbi, and Stuart Shieber. 2020. "Linguistic features for readability assessment." In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17. https://doi.org/10.18653/v1/2020.bea-1.1

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Fang, Meng, Jie Yin, and Dacheng Tao. 2014. Active learning for crowdsourcing using knowledge transfer. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1809–1815.

Fang, Meng, Xingquan Zhu, Bin Li, Wei Ding, and Xindong Wu. 2012. Self-taught active learning from crowds. In *2012 IEEE 12th International Conference on Data Mining*, pages 858–863. https://doi.org/10.1109/ICDM.2012.64

Felice, Mariano and Paula Buttery. 2019. Entropy as a proxy for gap complexity in open cloze tests. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 323–327. https://doi.org/10.26615/978-954-452-056-4_037

Fort, Karën and Benoît Sagot. 2010. Influence of pre-annotation on POS-tagged corpus development. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 56–63.

Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1923–1932.

Gao, Lingyu, Kevin Gimpel, and Arnar Jensson. 2020. Distractor analysis and selection for multiple-choice cloze questions for second-language learners. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 102–114.

Gebru, Timnit, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Communications of ACM*, 64(12):86–92. https://doi.org/10.1145/3458723

Geva, Mor, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? An investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166. https://doi.org/10.18653/v1/D19-1107

Gururangan, Suchin, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740

Hossain, Tamanna, Robert L. Logan IV, Arjuna Ugarte, Yoshitomo Matsubara, Sean Young, and Sameer Singh. 2020. COVIDLies: Detecting COVID-19 misinformation on social media. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, pages 1–11. https://doi.org/10.18653/v1/2020.nlpcovid19-2.11

Huang, Sheng-Jun, Rong Jin, and Zhi-Hua Zhou. 2010. Active learning by querying informative and representative examples. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, pages 892–900.

Jackson, Corey, Carsten Østerlund, Kevin Crowston, Mahboobeh Harandi, Sarah Allen, Sara Bahaadini, Scotty Coughlin, Vicky Kalogera, Aggelos Katsaggelos, Shane Larson, et al. 2020. Teaching citizen scientists to categorize glitches using machine learning guided training. *Computers in Human Behavior*, 105:1–11. https://doi.org/10.1016/j.chb.2019.106198

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3149–3157.

Kelly, A. V. 2009. *The Curriculum: Theory and Practice*, SAGE Publications, Thousand Oaks, CA.

Kicikoglu, Osman Doruk, Richard Bartle, Jon Chamberlain, Silviu Paun, and Massimo Poesio. 2020. Aggregation driven progression system for GWAPs. In *Workshop on Games and Natural Language Processing*, pages 79–84.

Kincaid, J. Peter, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and Flesch reading ease formula) for Navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch. https://doi.org/10.21236/ADA006655

Kirsch, Andreas, Joost van Amersfoort, and Yarin Gal. 2019. BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7026–7037.

Klie, Jan Christoph, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9.

Klie, Jan Christoph, Richard Eckart de Castilho, and Iryna Gurevych. 2020. From zero to hero: Human-in-the-loop entity linking in low resource domains. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6982–6993. https://doi.org/10.18653/v1/2020.acl-main.624

Krashen, Stephen. 1982. *Principles and Practice in Second Language Acquisition*, Pergamon Press, Oxford and New York.

Kruskal, William H. and W. Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621. https://doi.org/10.1080/01621459.1952.10483441

Kumar, M., Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23, pages 1189–1197.

Laws, Florian, Christian Scheible, and Hinrich Schütze. 2011. Active learning with Amazon Mechanical Turk. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556.

Lee, Ji Ung, Christian M. Meyer, and Iryna Gurevych. 2020. Empowering active learning to jointly optimize system and user demands. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4233–4247.

Lee, Ji Ung, Erik Schwan, and Christian M. Meyer. 2019. Manipulating the difficulty of c-tests. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 360–370. https://doi.org/10.18653/v1/P19-1035

Lewis, David D. and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. https://doi.org/10.1007/978-1-4471-2099-5_1

Lingren, Todd, Louise Deleger, Katalin Molnar, Haijun Zhai, Jareen Meinzen-Derr, Megan Kaiser, Laura Stoutenborough, Qi Li, and Imre Solti. 2014. Evaluating the impact of pre-annotation on annotation speed and potential bias: Natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *Journal of the American Medical Informatics Association*, 21(3):406–413. https://doi.org/10.1136/amiajnl-2013-001837, PubMed: 24001514

Loukina, Anastassia, Su-Youn Yoon, Jennifer Sakano, Youhua Wei, and Kathy Sheehan. 2016. Textual complexity as a predictor of difficulty of listening items in language proficiency tests. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3245–3253.

Lowell, David, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 21–30. https://doi.org/10.18653/v1/D19-1003

Madge, Chris, Juntao Yu, Jon Chamberlain, Udo Kruschwitz, Silviu Paun, and Massimo Poesio. 2019. Progression in a language annotation game with a purpose. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, pages 77–85.

Martínez Alonso, Héctor, Barbara Plank, Anders Johannsen, and Anders Søgaard. 2015. Active learning for sense annotation. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 245–249.

Mayfield, Elijah and Alan W. Black. 2020. Should you fine-tune BERT for automated essay scoring? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 151–162. https://doi.org/10.18653/v1/2020.bea-1.15

Mostow, Jack and Hyeju Jang. 2012. Generating diagnostic multiple choice comprehension cloze questions. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 136–146.

Nguyen, Hieu T. and Arnold Smeulders. 2004. Active learning using pre-clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 79–87. https://doi.org/10.1145/1015330.1015349

Nie, Yixin, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901. https://doi.org/10.18653/v1/2020.acl-main.441

Pak, Alexander and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 1320–1326.

Papakyriakopoulos, Orestis, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. 2020. Bias in word embeddings. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 446–457. https://doi.org/10.1145/3351095.3372843

Paun, Silviu, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian models of annotation. *Transactions of the Association for Computational Linguistics*, 6:571–585. https://doi.org/10.1162/tacl_a_00040

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Peters, Matthew E., Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14. https://doi.org/10.18653/v1/W19-4302

Reimers, Nils and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. https://doi.org/10.18653/v1/D19-1410

Rogers, Anna. 2021. Changing the world by changing the data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2182–2194. https://doi.org/10.18653/v1/2021.acl-long.170

Roy, Nicholas and Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 441–448.

Sakaguchi, Keisuke, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial Winograd Schema Challenge at scale. *Communications of ACM*, 64(9):99–106. https://doi.org/10.1145/3474381

Salazar, Julian, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712. https://doi.org/10.18653/v1/2020.acl-main.240

Sambasivan, Nithya, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M. Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–15. https://doi.org/10.1145/3411764.3445518

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Schulz, Claudia, Christian M. Meyer, Jan Kiesewetter, Michael Sailer, Elisabeth Bauer, Martin R. Fischer, Frank Fischer, and Iryna Gurevych. 2019. Analysis of automatic annotation suggestions for hard discourse-level tasks in expert domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2761–2772. `https://doi.org/10.18653/v1/P19-1265`

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. `https://doi.org/10.18653/v1/P16-1162`

Settles, Burr. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114. `https://doi.org/10.2200/S00429ED1V01Y201207AIM018`

Settles, Burr, Mark Craven, and Lewis Friedland. 2008. Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, pages 1–10.

Siddhant, Aditya and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909. `https://doi.org/10.18653/v1/D18-1318`

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263. `https://doi.org/10.3115/1613715.1613751`

Stab, Christian, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. ArgumenText: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 21–25. `https://doi.org/10.18653/v1/N18-5005`

Sweetser, Penelope and Peta Wyeth. 2005. GameFlow: A model for evaluating player enjoyment in games. *Computers in Entertainment*, 3(3):3. `https://doi.org/10.1145/1077246.1077253`

Szpiro, George. 2010. *Numbers Rule: The Vexing Mathematics of Democracy, from Plato to the Present*. Princeton University Press. `https://doi.org/10.1515/9781400834440`

Tauchmann, Christopher, Johannes Daxenberger, and Margot Mieskes. 2020. The influence of input data complexity on crowdsourcing quality. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*, pages 71–72. `https://doi.org/10.1145/3379336.3381499`

Taylor, Wilson L. 1953. "Cloze procedure": A new tool for measuring readability. *Journalism & Mass Communication Quarterly*, 30(4):415–433. `https://doi.org/10.1177/107769905303000401`

Tomanek, Katrin and Udo Hahn. 2009. Timed annotations: Enhancing MUC7 metadata by the time it takes to annotate named entities. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 112–115. `https://doi.org/10.3115/1698381.1698399`

Turner, Brandon M. and Dan R. Schley. 2016. The anchor integration model: A descriptive model of anchoring effects. *Cognitive Psychology*, 90:1–47. `https://doi.org/10.1016/j.cogpsych.2016.07.003`, PubMed: 27567237

VanderWeele, Tyler J. and Ilya Shpitser. 2013. On the definition of a confounder. *Annals of Statistics*, 41(1):196–220. `https://doi.org/10.1214/12-AOS1058`, PubMed: 25544784

Vygotsky, Lev. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press.

Wang, Min, Fan Min, Zhi-Heng Zhang, and Yan-Xue Wu. 2017. Active learning through density clustering. *Expert Systems with Applications*, 85:305–317. `https://doi.org/10.1016/j.eswa.2017.05.046`

Welch, Bernard Lewis. 1951. On the comparison of several mean values: An alternative approach. *Biometrika*, 38(3/4):330–336. `https://doi.org/10.1093/biomet/38.3-4.330`

Xia, Menglin, Ekaterina Kochmar, and Ted Briscoe. 2016. "Text readability assessment for second language learners.” In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22. https://doi.org/10.18653/v1/W16-0502

Yang, Yinfei, Oshin Agarwal, Chris Tar, Byron C. Wallace, and Ani Nenkova. 2019. Predicting annotation difficulty to improve task routing and model performance for biomedical information extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1471–1480. https://doi.org/10.18653/v1/N19-1150

Yeung, Chak Yan, John Lee, and Benjamin Tsou. 2019. Difficulty-aware distractor generation for gap-fill items. In *Proceedings of the 17th Annual Workshop of the Australasian Language Technology Association*, pages 159–164.

Yimam, Seid Muhie, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic annotation suggestions and custom annotation layers in WebAnno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96. https://doi.org/10.3115/v1/P14-5016

Yuan, Michelle, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948. https://doi.org/10.18653/v1/2020.emnlp-main.637

Zhang, Chicheng and Kamalika Chaudhuri. 2015. Active learning from weak and strong labelers. In *Advances in Neural Information Processing Systems*, pages 703–711.

Zhu, Jingbo, Huizhen Wang, Tianshun Yao, and Benjamin K. Tsou. 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1137–1144. https://doi.org/10.3115/1599081.1599224

Chapter 11

Lessons Learned from a Citizen Science Project for Natural Language Processing

Lessons Learned from a Citizen Science Project for Natural Language Processing

Jan-Christoph Klie[1] **Ji-Ung Lee**[1] **Kevin Stowe**[1,2] **Gözde Gül Şahin**[1,3]
Nafise Sadat Moosavi[1,4] **Luke Bates**[1] **Dominic Petrak**[1]
Richard Eckart de Castilho[1] **Iryna Gurevych**[1]

[1]Ubiquitous Knowledge Processing Lab (UKP Lab)
Department of Computer Science and Hessian Center for AI (hessian.AI)
Technical University of Darmstadt
www.ukp.tu-darmstadt.de

[2]Educational Testing Service
[3]KUIS AI, Koç University
[4]Department of Computer Science, The University of Sheffield

Abstract

Many Natural Language Processing (NLP) systems use annotated corpora for training and evaluation. However, labeled data is often costly to obtain and scaling annotation projects is difficult, which is why annotation tasks are often outsourced to paid crowdworkers. Citizen Science is an alternative to crowdsourcing that is relatively unexplored in the context of NLP. To investigate whether and how well Citizen Science can be applied in this setting, we conduct an exploratory study into engaging different groups of volunteers in Citizen Science for NLP by re-annotating parts of a pre-existing crowdsourced dataset. Our results show that this can yield high-quality annotations and attract motivated volunteers, but also requires considering factors such as scalability, participation over time, and legal and ethical issues. We summarize lessons learned in the form of guidelines and provide our code and data to aid future work on Citizen Science.[1]

1 Introduction

Data labeling or *annotation* is often a difficult, time-consuming, and therefore expensive task. Annotations are typically drawn from domain experts or are crowdsourced. While experts can produce high-quality annotated data, they are expensive and do not scale well due to their relatively low number (Sorokin and Forsyth, 2008). In contrast, crowdsourcing can be relatively cheap, fast, and scalable, but is potentially less suited for more complicated annotation tasks (Drutsa et al., 2020). Another approach is using Citizen Science, which describes the participation and collaboration of volunteers from the general public with researchers to conduct science (Haklay et al., 2021). Over the past decade, Citizen Science platforms, which rely on unpaid volunteers to solve scientific problems, have been used for a wide variety of tasks requiring human annotation (Hand, 2010), e.g., classifying images of galaxies (Lintott et al., 2008) or for weather observation (Leeper et al., 2015).

While Citizen Science has been shown to produce high-quality annotations in ecological or environmental projects (Kosmala et al., 2016), its potential has so far not been investigated in depth for Natural Language Processing (NLP). Our goal in this work is to assess the practicality of undertaking annotation campaigns for NLP via Citizen Science. We analyze whether volunteers actually react to our calls and participate, how the resulting quality is compared to crowdsourcing, what the benefits and shortcomings are and what needs to be taken into account when conducting such a project. We especially are interested in differences between annotators recruited via different channels, which we investigate by advertising to different social media platforms, NLP-related mailing lists, and university courses. To explore this possibility, we use the PERSPECTRUM dataset (Chen et al., 2019, CC-BY-SA) that focuses on the task of stance detection and can be motivated by fighting misinformation and promoting accurate debate in internet discussions. We replicated a portion of the annotations in this dataset using citizen scientists instead of crowdworkers. To accomplish this goal, we designed an annotation workflow that is suitable for Citizen Science and allows us to recruit volunteers across a variety of platforms.

[1] https://github.com/UKPLab/eacl2023-citizen-science-lessons-learned

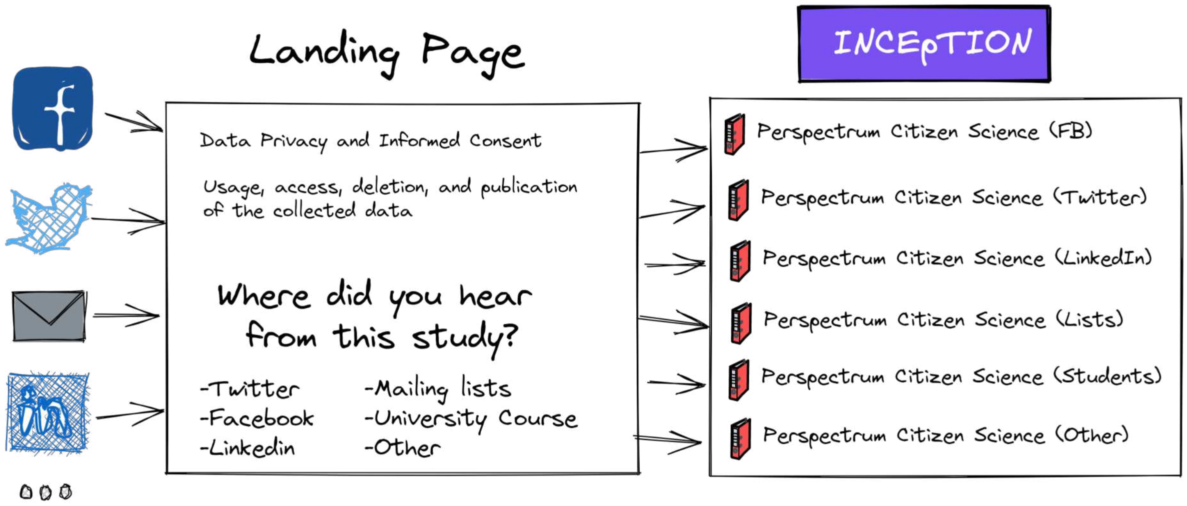

Landing Page
Data Privacy and Informed Consent
Usage, access, deletion, and publication of the collected data
Where did you hear from this study?
-Twitter
-Facebook
-Linkedin
-Mailing lists
-University Course
-Other
INCEpTION
Perspectrum Citizen Science (FB)
Perspectrum Citizen Science (Twitter)
Perspectrum Citizen Science (LinkedIn)
Perspectrum Citizen Science (Lists)
Perspectrum Citizen Science (Students)
Perspectrum Citizen Science (Other)

Figure 1: We advertised our project via various social media, mailing lists and university courses. Volunteers then are onboarded via the landing page and donated annotations via INCEpTION.

Our contributions are the following:

1. We provide a systematic study on Citizen Science across different channels and analyze turnout and quality. For this, we re-annotate parts of the PERSPECTRUM dataset using Citizen Science and compare these to the original, crowdsourced annotations.

2. We provide guidelines and recommendations on how to successfully conduct a Citizen Science project for NLP annotation and discuss critical legal and ethical aspects.

3. We provide a platform for future Citizen Science projects that handles onboarding, anonymous access, work assignment and the annotating itself.

Our results show that using Citizen Science for linguistic annotation can result in high-quality annotations, but that attracting and motivating people is critical for its success, especially in the long-term. We were able to attract 98 volunteers when conducting our Citizen Science project which resulted in 1,481 annotations over 2 months, thereby re-annotating around 10% of the original dataset. We find that annotations obtained through mailing lists and university students were of high quality when comparing them to the original, adjudicated crowdsourced data. We thus conclude that Citizen Science projects have the potential to be applied to NLP annotation if they are conceptualized well, but are best suited for creating smaller datasets.

2 Background

Prior work has developed various means and strategies for annotating large datasets. So far, annotation studies in NLP mainly use domain-experts or crowdworkers, or a mix of both (Nguyen et al., 2015). Crowdsourcing in particular has received increasing attention over the past decade (Wang et al., 2013).

Paid Experts Recruiting domain experts (e.g., linguists) for annotation studies has been a widely accepted method to generate linguistically annotated corpora. Famous examples are the Brown Corpus (Francis and Kucera, 1979) or the Penn Treebank (Marcus et al., 1993). While the resulting datasets are of the highest quality, domain experts are often few, and such annotation studies tend to be slow and expensive (Sorokin and Forsyth, 2008). Although many researchers moved on to annotation studies that recruit crowdworkers, expert annotations are still necessary in various fields, e.g., biomedical annotations (Hobbs et al., 2021).

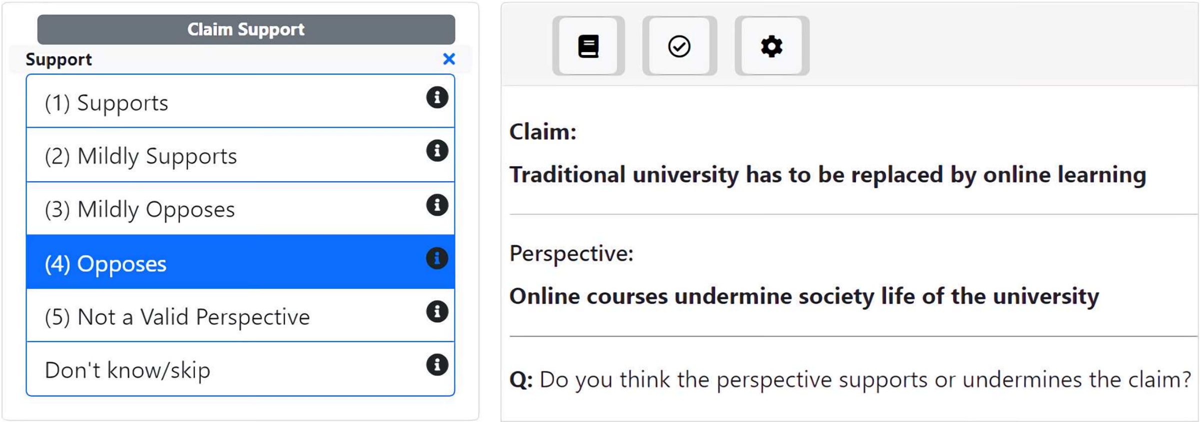
Crowdsourcing To accelerate the annotation process and reduce costs, researchers have utilized crowdsourcing as a means to annotate large corpora (Snow et al., 2008). The main idea behind crowdsourcing is that annotation tasks that do not require expert knowledge can be assigned to a large group of paid non-expert annotators. This is commonly done via crowdsourcing platforms such as Amazon Mechanical Turk (AMT) or Upwork and has been successfully used to annotate various datasets across different tasks and domains (Derczynski et al., 2016; Habernal and Gurevych, 2017). Previous work compared the quality between crowdsourcing and expert annotations, showing that many tasks can be given to crowdworkers without major impact on the quality of annotation (Snow et al., 2008; Hovy et al., 2014; De Kuthy et al., 2016).

Although crowdworkers can substantially accelerate annotation, crowdsourcing requires careful task design and is not always guaranteed to result in high quality data (Daniel et al., 2018). Moreover, as annotators are compensated not by the time they spend but rather by the number of annotated instances, they are compelled to work fast to maximize their monetary gain—which can negatively affect annotation quality (Drutsa et al., 2020) or even result in spamming (Hovy et al., 2013). It can also be difficult to find crowdworkers for the task at hand, for instance due to small worker pools for languages other than English (Pavlick et al., 2014; Frommherz and Zarcone, 2021) or because the task requires special qualifications (Tauchmann et al., 2020). Finally, the deployment of crowdsourcing remains ethically questionable due to undervalued payment (Fort et al., 2011; Cohen et al., 2016), privacy breaches, or even psychological harm on crowdworkers (Shmueli et al., 2021).

Games with a Purpose A related but different way to collect annotations from volunteers is *games with a purpose*, i.e., devising a game in which participants annotate data (Chamberlain et al., 2008; Venhuizen et al., 2013). Works propose games for different purposes and languages. For instance, anaphora annotation (PhraseDetectives, Poesio et al. 2013), dependency syntax annotation (Zombilingo, Fort et al. 2014), or collecting idioms (Eryiğit et al., 2022). It has been shown that if a task lends itself to being gamified, then it can attract a wide audience of participants and can be used to create large-scale datasets (von Ahn, 2006). Finally, Lyding et al. (2022) investigate games with a purpose in the context of (second) language learning to simultaneously crowdsource annotaions from learners as well as teachers. One such example is Substituto, a turn-based, teacher-moderated game for learning verb-particle constructions (Araneta et al., 2020). We do not consider gamification in this work, as enriching tasks with game-like elements requires considerable effort and cannot be applied to every task.

Citizen Science Citizen Science broadly describes participation and collaboration of the general public (the citizens) with researchers to conduct science (Haklay et al., 2021). Citizen Science is a popular alternative approach for dataset collection efforts, and has been successfully applied in cases of weather observation (Leeper et al., 2015), counting butterflies (Holmes, 1991) or birds (National Audubon Society, 2020), classifying images of galaxies (Lintott et al., 2008) or monitoring water quality (Addy et al., 2010). Newly-emerging technologies and platforms further allow researchers to conduct increasingly innovative Citizen Science projects, such as the prediction of influenza-like outbreaks (Lee et al., 2021) or the classification of animals from the Serengeti National Park (Swanson et al., 2015). *LanguageARC* is a Citizen Science platform for developing language resources (Fiumara et al., 2020). It is however not open yet to the public to create projects and does not easily allow conducting a Citizen Science meta-study as we do in this work. One work using LanguageARC is by Fort et al. (2022) (LD) who collected resources to evaluate bias in language models. They did not investigate the impact of using different recruitment channels which we do. Other projects using LanguageARC are still running and it is too early to derive recommendations from.

Compared to crowdsourcing, Citizen Science participants are volunteers that do not work for monetary gain. Instead, they are often motivated intrinsically. For instance, they may have a personal interest on positively impacting the environment (West et al., 2021), or in altruism (Rotman et al., 2012). Asking for unpaid work also entails various issues like finding good ways of how to attract volunteers, and ethical considerations (Resnik et al., 2015; Rasmussen and Cooper, 2019) that need to be addressed (cf. §5). Intrinsic motivation also has the potential of resulting in higher-quality annotations compared to crowdsourcing. For instance, Lee et al. (2022) find in their evaluation study with citizen scientists that their participants may have been willing to take more time annotating for the sake of higher annotation accuracy. However, as their main goal was to conduct an evaluation study for their specific setup, this finding cannot be generalized to other Citizen Science scenarios. So far, only Tsueng et al. (2016) provide a direct comparison between crowdsourcing and Cit-



Claim:

Traditional university has to be replaced by online learning

Perspective:

Online courses undermine society life of the university

Q: Do you think the perspective supports or undermines the claim?

Figure 2: Assigning a label to an instance in the INCEpTION text annotation platform.

izen Science and show that volunteers can achieve similar performance in mining medical entities in scientific texts. They recruit participants through different channels such as newspapers, Twitter, etc., but do not compute channel-specific performance, making it difficult to assess whether the quality of the resulting annotation depends on the recurrent channel. In contrast, in the present work, we explicitly consider the recruitment channel in our evaluation and furthermore provide a discussion and guidelines for future Citizen Science practitioners. Also, it attracts intrinsically (not only fiscally) motivated volunteers that are often skilled in the task and can provide high-quality annotations, thus potentially combining the advantages of expert annotations and crowdsourcing. Relying on unpaid annotators entails several issues, including attracting volunteers and ethical considerations (Resnik et al., 2015; Rasmussen and Cooper, 2019) that need to be taken into account (see §5).

3 Study Design

To study the feasibility of Citizen Science for NLP annotation, we asked volunteers recruited via various channels to re-annotate an existing, crowdsourced dataset. The general setup is described in Fig. 1. To conduct a systematic study, we identified the following four necessary steps: 1) Identifying a suitable dataset (§3.1); 2) Selecting suitable recruitment channels to advertise our project on (§3.2); 3) Building a landing page for onboarding participants that asks for informed consent and the channel from which they originated (§3.3); 4) Setting up the annotation editor to which participants are forwarded after the onboarding (§3.4).

3.1 Dataset selection

We first conducted a literature review of relevant crowdsourced NLP datasets to identify the ones that could be accurately reproduced via Citizen Science. We assessed datasets for the following two criteria: 1) **Availability**: the dataset must be publicly available to make proper comparisons in terms of annotator agreement; 2) **Reproducibility**: the annotation setup including annotation guidelines needs to be reproducible to ensure similar conditions between citizen scientists and crowdworkers. We focused on datasets that are targeted towards contributing to social good to encourage volunteers to participate. Unfortunately, many inspected datasets did not fulfill both of these requirements. Overall, we identified two main issues while screening over 20 candidate datasets. First, many datasets used Tweets which impacted reproducibility as Twitter only allows researchers to publish the tweet identifiers. This leads to irrecoverable instances when tweets were deleted. Second was the lack of precise guidelines. For instance, many considered datasets about societal biases lack explicit descriptions of what is considered a stereotype. As such biases are often also impacted by the respective cultural background of annotators, they are difficult to reproduce without specific guidelines.

In the end, we decided on the stance detection task of the PERSPECTRUM dataset (Chen et al., 2019). The task provides clear instructions, publicly available data, and is motivated by social good (fighting misinformation/promoting accurate debate in internet discussions). Each instance consists of a claim–perspective pair (cf. Fig. 2) and annotators are asked if the claim *supports*, *opposes*, *mildly-supports*, *mildly-opposes*, or is *not a valid*

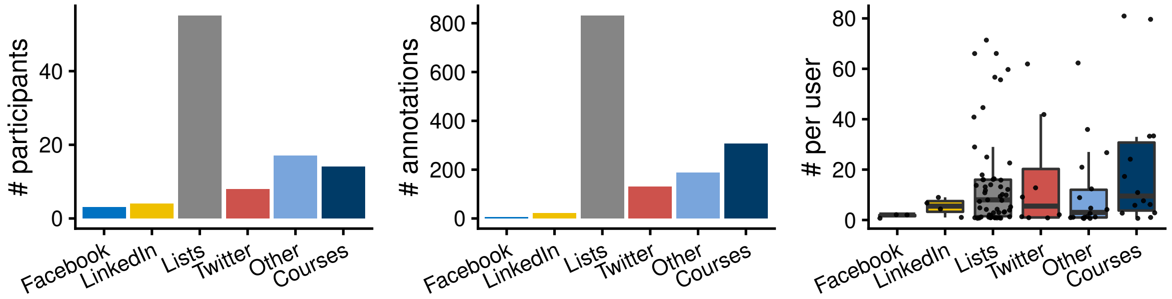


Figure 3: Participants, annotations and annotations grouped by the channel via they were recruited. It can be seen that overall, most participants and annotations were contributed by annotators recruited via mailing lists. Annotators from mailing lists and courses yielded the volunteers who contributed the most individually.

perspective. Following the original work, we also evaluated the annotations on a coarser tagset that only contains the categories for *support*, *oppose* and *not a valid perspective*. Overall, the dataset consists of 907 claims and $8,370$ different perspectives which yield $11,805$ annotated instances. In preliminary studies, we received further feedback that forcing annotators to provide an explicit label for each instance could lead to increasing frustration, especially for ambiguous or complicated instances. To lessen the burden for our voluntary annotators and keep them motivated in the annotation task, we allowed them to skip instances (*Don't know/skip*) which was not present in the original annotation editor for PERSPECTRUM.

3.2 Recruitment channels

To recruit annotators, we advertised our project on three social media platforms, namely, Twitter, LinkedIn and Facebook. Unfortunately, after creating the Facebook organization and advertising the project, the account was banned due to "violating their community standards" and has so far remained banned. One of our team members then promoted our annotation study on their personal Facebook to attract participation from this social media platform. In addition, the team members advertised the work on Twitter and in relevant LinkedIn groups such as COMPUTATIONAL LINGUISTICS and MACHINE LEARNING AND DATA SCIENCE.

We further promoted the study via two external mailing lists (i.e., CORPORA-LIST, ML-NEWS). Late in the project, we received interest from other faculty to advertise the task in their courses—an offer that we gladly accepted. For this, participation was completely voluntary and anonymous, students' grades were not affected by participation, and authors were not among the instructors. To evaluate different recruitment channels separately, we asked participants on the landing page to answer the question: "Where did you hear from this study?". We also allowed volunteers to not disclose how they found out about the study, this is referred to as "Other" or "Undisclosed" in this paper. Final participation counts are given in Fig. 3. We deliberately limited our outreach, e.g. we did not use university social media accounts or colleagues with large follower bases. Also, we made sure to not exhaust channels by posting too many calls for participation.

3.3 Landing page

We implemented a customizable landing page web application catering to the needs of Citizen Science projects. The link to such a landing page was shared via the respective recruitment channels. The landing page contained information about the study itself, its purpose, its organizers, which data we collected, and its intended use. This landing page toolbox is designed so that it can easily be adapted to future Citizen Science projects. To allow project creators to use an annotation editor of their choice, we designed the toolbox to act as an intermediary that collects a participant's consent for the actual annotation study. This ensures that only participants that have been properly informed and have explicitly provided their consent are given access to the study. For future Citizen Science projects, the tool further assists organizers through the landing page creation process to foster an ethical collection of data by asking several questions, that are listed in the appendix.

3.4 Annotation editor

INCEpTION (Klie et al., 2018) offers a configurable, web-based platform for annotating text documents at span, relation and document levels. To make it usable in Citizen Science scenarios, we extended the platform with three features, namely, (1) the ability to join a project through a link, (2) support for anonymous guest annotators, and (3) a dynamic workload manager. Allowing citizen scientists to participate in the project anonymously as guests without any sign-up process substantially reduced the entry barrier and made it easier for us to satisfy data protection policies. The same is true for the ability of joining a project through an invite link. Upon opening the link, annotators were greeted with the annotation guidelines and were directly able to start annotating. Finally, we implemented a dynamic workload manager that takes as input the desired number of annotators per document and then automatically forwards annotators directly to the document instances requiring annotation. Upon finishing annotating an instance, INCEpTION was configured to automatically load and display the next instance for annotation, similar to popular crowdsourcing platforms. We also included rules for handling other issues that may occur with voluntary annotations such as recovering instances that annotators have started to work on but then abandoned. Additionally, we modified the existing user interface to improve the annotation workflow. This mainly included implementing a dedicated labeling interface that allows users to select a single label for an instance via a radio button group. Annotation of an instance thus required two user actions: first, selecting the document label, and second, confirming the annotation, thereby moving on to the next document.

4 Results

We conducted our study between January and March 2022 and promoted the task in successive rounds across all recruitment channels. In total, we were able to recruit 98 participants who provided 1481 annotations resulting in 906 fully annotated instances. Each instance with at least one annotation has received on average 1.63 annotations. Detailed statistics are provided in the appendix.

Participation To identify promising channels for future Citizen Science studies, we report the number of annotators per channel, the total number of annotations per channel and per user (cf. Fig. 3). Overall, we find that the most effective channel for public outreach are mailing lists (55 participants). Asking students in university courses to participate was the second most effective with 14 participants. Facebook, LinkedIn, and Twitter only yielded three, four, and eight participants respectively. We further find a highly skewed distribution of annotations per user, as many annotators only provide a few annotations while a few annotators provide many annotations. For instance, the most active annotators were two students who provided ∼80 annotations as well as six participants from mailing lists who provided ∼60–80 annotations each. For Twitter and "undisclosed", only a single annotator made over 60 annotations. We also find that on average, participants from university courses provided the most annotations per person. When looking at participation over time (see Fig. 5), we observe increased activity in annotations made after the call for participation has been posted to the respective channel. For many channels, the count quickly flattens. Interestingly, Twitter sees a second spike long after the post was made. We attribute it to people sharing the post in our community quite a while after the initial release. We did not track whether individual volunteers came back for another round of annotations after their initial participation.

Coverage Overall, our 98 volunteers have provided 1,481 annotations to 906 unique instances (approximately 8% of the original dataset) over two months. This is comparable to other Citizen Science projects like Fort et al. (2022), which had 102 participants in total. They annotated three tasks and collected 2347, 2904 and 220 submissions over eight months. Table 1 shows the resulting coverage of our Citizen Science annotation study. While this still leaves room for improvement, the number of annotations collected nonetheless shows that Citizen Science can be viable in real life settings and is a promising direction to investigate in further studies, especially for creating focused and smaller-scale resources.

Quality In terms of annotation quality, we find that most channels yield annotations that highly agree with the gold labels (cf. Table 2), even though our annotations are not adjudicated yet. We further find that volunteers from university courses and mailing list show the highest accuracy, followed by Twitter and "undisclosed". Only LinkedIn yields

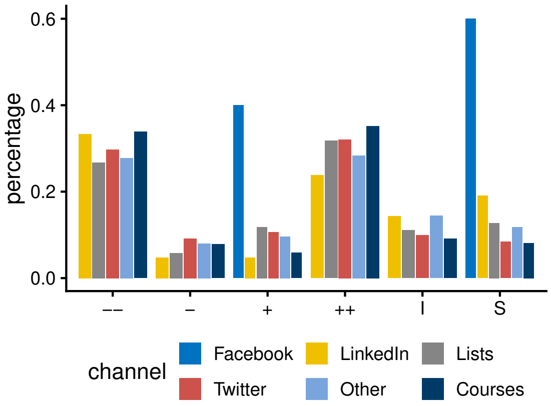


Figure 4: Label distribution grouped by channel. Labels are *supports* (++), *mildly-supports* (+), *mildly-opposes* (-), *opposes* (--), *not a valid perspective* (I) and *Skip* (S).

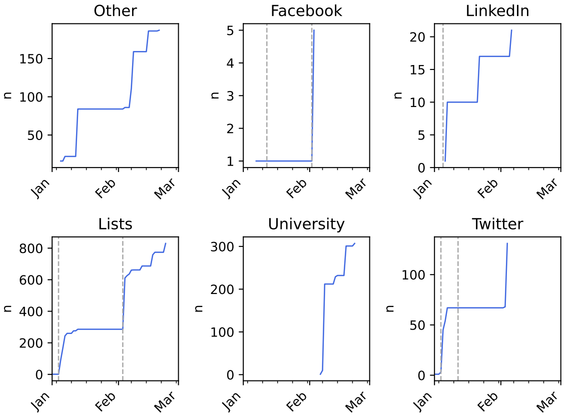


Figure 5: Annotations made over time. Vertical lines indicate when calls on the respective channels have been posted.

lower accuracy than 70% on the coarse label set.

For the majority of channels (with the exception of Facebook and LinkedIn), we only see a skip percentage of ∼10% (cf. Fig. 4). This indicates our volunteers are actually willing to spend time and effort to solve the task at hand, as adding a "Don't know/skip" option in crowdsourcing usually is an invitation for workers to speed through the tasks and not provide useful annotations. The exception is Facebook, where we find that a majority of the annotations from Facebook were labeled as *I don't know/skip* (3 out of 5). Further analysis of the label distribution grouped by channel (cf. Fig. 4) shows that all channels except for Facebook display a similar distribution in terms of annotated labels. This indicates that we can expect a rather stable annotation performance across citizen scientists recruited from different channels.

5 Discussion and Takeaways

Here we present lessons learned, discuss legal challenges and ethical considerations, as well as provide guidelines for future Citizen Science projects.

Table 1: Claims, claim clusters, and individual claim-perspective pairs that have been annotated at least once. We call the set of a claim and a perspective together with its paraphrases a claim cluster.

Name	# Annotated	# Total	% Annotated
Claims	388	907	42.78
Clusters	739	5092	14.51
Total	906	11805	7.67

Table 2: Annotation accuracy compared to the crowd-sourced and adjudicated data from PERSPECTRUM. The five annotations from Facebook (three of them were skipped) and *Don't know/skip* annotations are omitted.

Channel	Coarse	Fine
University	0.92	0.82
LinkedIn	0.69	0.62
Mailing Lists	0.90	0.82
Undisclosed	0.84	0.75
Twitter	0.85	0.73

Channel-dependent differences Our results clearly differ across recruiting channels. We find that overall, Facebook and LinkedIn have the lowest turnout and accuracy when compared to the gold labels, followed by Twitter. Our assumption for the overall low participation is that our network for these channels was not large enough. Advertising our study to NLP-related and university-internal mailing lists and university courses yielded the highest number of participants who also provided the most and best-quality annotations. Although our results show that students may outperform participants from other channels, we also acknowledge that this may not always be a viable option to recruit citizen scientists. Overall, our findings indicate that it is important to address the respective target groups that may be interested in a specific study. However, we also note that continuously advertising Citizen Science studies to the same channels may have a negative impact, as it can cause participation fatigue and lead to fewer volunteers participating. One possible solution could be the use of LanguageARC (Fiumara et al., 2020) from the LDC and centralize calls for participation.

Motivating volunteers In contrast to crowdsourcing, there is no monetary or other extrinsic motivation that could be used to attract Citizen Science annotators. Thus, annotator motivation is a crucial question for Citizen Science studies. As Fig. 5 shows, citizen scientists can be quickly motivated to participate, but can also quickly lose interest in a given annotation study. This can become an issue with a low number of participants, yet our results also indicate that we were able to find highly-motivated participants (8 out of 98 in our results).

Compared to other groups, university students in particular provided a high amount of quality annotations. Considering the findings by Phillips et al. (2018), who do not find statistical differences in terms of quality between students participating for course credit vs. no extrinsic reward—asking students to participate in such projects as part of their coursework might be another good option, but needs to ensure an ethical data collection. For instance, such an approach has been used to annotate the Georgetown University Multilayer Corpus (Zeldes, 2017). Nonetheless, one remaining question is how to keep participants motivated and participate in several sessions as our results indicate that a vast majority of our volunteers only participated in a single session and that participation quickly stops shortly after a call has been posted to the respective channel.

Finally, we want to emphasize the inclusion of a *Don't know/skip* option for Citizen Science annotators. Whereas in crowdsourcing studies, annotators may exploit such an option to increase their gain (Hovy et al., 2013), from the feedback we got during our pilot study, it is crucial to keep volunteers motivated for Citizen Science. For this work, we did not provide a survey that asks about the motivation, as we thought that this might deter potential participants We however suggest that future studies provide such a survey that is as unintrusive as possible to further analyze why participants take part in the respective annotation project.

Legal challenges One substantial challenge in implementing Citizen Science studies is the potentially wide outreach they can have and, consequently, the varying kinds of data protection regulations they have to oblige. To preempt any potential issues that can arise—especially when data that can be used to identify a person (personal data, e.g. obtained during a survey or login credentials) is involved—we recommend researchers who plan to implement a Citizen Science study consider the most strict regulations that are widely accepted.

For the GDPR (European Parliament, 2016), currently one of the strictest data protection regulations, we recommend researchers to explicitly ask voluntary participants for their informed consent when collecting personal information. This includes informing participants beforehand about (1) the purpose of the data collection, (2) the kind of personal and non-personal data collected, (3) the planned use of the data, (4) any planned anonymization processes for publication, and finally, (5) how participants can request access, change, and deletion of the data. We further recommend assigning one specific contact person for any questions and requests for access, change, or deletion of the data. This may seem like additional work when compared to crowdsourcing, but transparent and open communication is one of the key factors to build trust—which is necessary for voluntary participants to consider such studies and provide high-quality annotations. Finally, participants should be informed and agree to the annotations donated being published under a permissive license.

Ethical and economical considerations Although Citizen Science can substantially reduce annotation costs, we emphasize the importance of considering an ethical deployment that does not compromise the trust of the participants. Moreover, given increasing concerns regarding the ownership and use of collected data (Arrieta-Ibarra et al., 2018), one should grant participants full rights to access, change, delete, and share their own personal data (Jones and Tonetti, 2020). This ensures that participants are not exploited for "free labor"—in contrast to approaches like reCAPTCHA (von Ahn et al., 2008), where humans are asked to solve a task in order to gain access to services. Whereas CAPTCHAs were initially intended to block malicious bots, they are becoming increasingly problematic due to their deployment and use by monopolizing companies which raises ethical concerns (Avanesi and Teurlings, 2022) . It is especially important to take the data itself into consideration; exposing volunteers to toxic, hateful, or otherwise sensitive speech should be avoided if they are not informed about it beforehand.

Recommendations Overall, we derive the following recommendations for future Citizen Science studies. 1) our call for annotations resonated the most with the target group that is likely to benefit the most from contributing to it: NLP researchers coming from mailing lists and university students. Therefore, the target audience should be carefully selected, for instance by identifying topic-specific mailing lists or respective university courses. This further means that the purpose of data collection should be made clear and that the results should be made publicly available. 2) the research question of the study should conform to the respective ethical and legal guidelines of the potential target group which should clearly be communicated to make the project accountable. 3) participation should be easy with clearly formulated annotation guidelines and, moreover, the annotation itself should be thoroughly tested beforehand to ensure that participants do not get frustrated due to design errors or choices. For instance, in our preliminary study, we got the feedback that some instances are frustrating to annotate and hence added an option to skip. 4) analyzing participation over time shows that a Citizen Science project has to be continuously advertised in order to stay relevant and achieve high participation. Otherwise, it will be forgotten quickly. This can be done by sharing status updates or creating preliminary results. Fifth, we recommend asking about user motivation before, during or after the annotation with a survey to better understand the participants and their demographics.

6 Conclusion

In this work, we presented an exploratory annotation study for utilizing Citizen Science for NLP annotation. We developed an onboarding process that can easily be adapted to similar projects and evaluated Citizen Science annotations for re-annotating an existing dataset. Furthermore, we extended the INCEpTION platform, a well-known open-source semantic annotation platform, with a dynamic workload manager and functionality for granting access to external users without registration. This enables its usage for Citizen Science projects. We advertised the study via Twitter, Facebook, LinkedIn, mailing lists, and university courses and found that participants from mailing lists and university courses are especially capable of providing high-quality annotations. We further discuss legal and ethical challenges that need to be addressed when conducting Citizen Science projects and provide general guidelines for conducting future projects that we would like to have known before starting. Overall, we conclude that Citizen Science can be a viable and affordable alternative to crowdsourcing, but is limited by successfully keeping annotators motivated. We will make our code and data publicly available to foster more research on Citizen Science for NLP and other disciplines.

Future Work We see the following directions for further research and evaluation to better understand in which settings Citizen Science can be applicable and how to use it best. Here, we used PERSPECTRUM as the dataset to annotate and mentioned in the participation calls that it benefits the social good. Therefore, it would be interesting to conduct more projects and see which datasets are suitable as well as whether volunteers participate, even if there is no extrinsic motivation. Then, it can also be tested how annotator retention develops, especially when project are running longer. The call for participation itself could also be investigated for the impact it has on turnout, motivation and quality.

7 Limitations

Throughout this article, we analyzed whether Citizen Science applies to linguistic annotation and showed that we can attract volunteers that donate a sizeable number of high-quality annotations. This work, however, comes with limitations that should be taken into account and tackled in future work. First, we based our analysis on a single annotation campaign and dataset that we advertised as being relevant for the social good. Therefore we suggest conducting more such annotation projects, also with different kinds of tasks. Second, we did not perform a user survey that for instance asked for user motivation. This is why we can only speculate about the motivation of our participants and suggest future works to explicitly prepare such a survey. Third, using Facebook as a channel might be viable, but we were not able to properly analyze it, as our account was blocked shortly after creation and never was reinstantiated. Finally, based on participation and annotation numbers, we see Citizen Science as more of an option for annotating smaller datasets, or longer-term projects that are more actively advertised than in our study which took place over two months and for which we deliberately limited the outreach.

Acknowledgments

We thank our anonymous reviewers, Michael Bugert and Max Glockner for their detailed and helpful comments to improve this manuscript. We are especially grateful for the discussions with Michael and Anne-Kathrin Bugert regarding our study setup. Finally, we thank the many volunteers that donated annotations for this project, as this work would not have been possible without their generous participation.

This work has been funded by the German Research Foundation (DFG) as part of the Evidence project (GU 798/27-1), UKP-SQuARE (GU 798/29-1), INCEpTION (GU 798/21-1) and PEER (GU 798/28-1), and within the project "The Third Wave of AI" funded by the Hessian Ministry of Higher Education, Research, Science and the Arts (HWMK). Further, it has been funded by the German Federal Ministry of Education and Research and HMWK within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

K Addy, L Green, E Herron, and K Stepenuck. 2010. Why volunteer water quality monitoring makes sense. usdanifa volunteer water quality monitoring national facilitation project, factsheet ii. *US Department of Agriculture, Washington, DC.*

Marianne Grace Araneta, Gülşen Eryiğit, Alexander König, Ji-Ung Lee, Ana Luís, Verena Lyding, Lionel Nicolas, Christos Rodosthenous, and Federico Sangati. 2020. Substituto – A Synchronous Educational Language Game for Simultaneous Teaching and Crowdsourcing. In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 1–9, Gothenburg, Sweden.

Imanol Arrieta-Ibarra, Leonard Goff, Diego Jiménez-Hernández, Jaron Lanier, and E. Glen Weyl. 2018. Should We Treat Data as Labor? Moving Beyond "Free". *AEA Papers and Proceedings*, 108:38–42.

Vino Avanesi and Jan Teurlings. 2022. "I'm not a robot," or am I?: Micro-labor and the immanent subsumption of the social in the human computation of ReCAPTCHAs. *International Journal of Communication*, 16(0):1–19.

Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. 2008. Addressing the resource bottleneck to create large-scale annotated texts. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 375–380.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. Seeing Things from a Different Angle: Discovering Diverse Perspectives about Claims. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 542–557, Minneapolis, Minnesota.

K. Bretonnel Cohen, Karën Fort, Gilles Adda, Sophia Zhou, and Dimeji Farri. 2016. Ethical Issues in Corpus Linguistics And Annotation: Pay Per Hit Does Not Affect Effective Hourly Rate For Linguistic Resource Development On Amazon Mechanical Turk. *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)*, 2016(W40):8–12.

Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. *ACM Computing Surveys*, 51(1):1–40.

Kordula De Kuthy, Ramon Ziai, and Detmar Meurers. 2016. Focus annotation of task-based data: A comparison of expert and crowd-sourced annotation in a reading comprehension corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3928–3935, Portorož, Slovenia.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan.

Alexey Drutsa, Valentina Fedorova, Dmitry Ustalov, Olga Megorskaya, Evfrosiniya Zerminova, and Daria Baidakova. 2020. Crowdsourcing Practice for Efficient Data Labeling: Aggregation, Incremental Relabeling, and Pricing. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2623–2627, Portland, Oregon, USA.

GülŞen Eryiğit, Ali Şentaş, and Johanna Monti. 2022. Gamified crowdsourcing for idiom corpora construction. *Natural Language Engineering*, 1(1):1–33.

European Parliament. 2016. Consolidated text: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance).

James Fiumara, Christopher Cieri, Jonathan Wright, and Mark Liberman. 2020. LanguageARC: Developing Language Resources Through Citizen Linguistics. In *Proceedings of the LREC 2020 Workshop on "Citizen Linguistics in Language Resource Development"*, pages 1–6, Marseille, France.

Karën Fort, Gilles Adda, and K. Bretonnel Cohen. 2011. Amazon Mechanical Turk: Gold Mine or Coal Mine? *Computational Linguistics*, 37(2):413–420.

Karën Fort, Bruno Guillaume, and Hadrien Chastant. 2014. Creating *Zombilingo* , a game with a purpose for dependency syntax annotation. In *Proceedings of the First International Workshop on Gamification for Information Retrieval*, pages 2–6, Amsterdam, The Netherlands.

Karën Fort, Aurélie Névéol, Yoann Dupont, and Julien Bezançon. 2022. Use of a citizen science platform for the creation of a language resource to study bias in language models for French: A case study. In *Proceedings of the 2nd Workshop on Novel Incentives in Data Collection from People: Models, Implementations, Challenges and Results within LREC 2022*, pages 8–13, Marseille, France.

W. N. Francis and H. Kucera. 1979. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Yannick Frommherz and Alessandra Zarcone. 2021. Crowdsourcing Ecologically-Valid Dialogue Data for German. *Frontiers in Computer Science*, 3:1–21.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation Mining in User-Generated Web Discourse. *Computational Linguistics*, 43(1):125–179.

Mordechai Haklay, Daniel Dörler, Florian Heigl, Marina Manzoni, Susanne Hecker, and Katrin Vohland. 2021. What Is Citizen Science? The Challenges of Definition. In Katrin Vohland, Anne Land-Zandstra, Luigi Ceccaroni, Rob Lemmens, Josep Perelló, Marisa Ponti, Roeland Samson, and Katherin Wagenknecht, editors, *The Science of Citizen Science*, pages 13–33. Springer International Publishing, Cham.

Eric Hand. 2010. People power: Networks of human minds are taking citizen science to a new level. *Nature*, 466(7307):685—687.

Elizabeth T. Hobbs, Stephen M. Goralski, Ashley Mitchell, Andrew Simpson, Dorjan Leka, Emmanuel Kotey, Matt Sekira, James B. Munro, Suvarna Nadendla, Rebecca Jackson, Aitor Gonzalez-Aguirre, Martin Krallinger, Michelle Giglio, and Ivan Erill. 2021. ECO-CollecTF: A Corpus of Annotated Evidence-Based Assertions in Biomedical Manuscripts. *Frontiers in Research Metrics and Analytics*, 6:1–13.

Anthony M Holmes. 1991. *The Ontario Butterfly Atlas*. Entomologists' Association, Toronto.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 377–382, Baltimore, Maryland.

Charles I. Jones and Christopher Tonetti. 2020. Nonrivalry and the Economics of Data. *American Economic Review*, 110(9):2819–2858.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9.

Margaret Kosmala, Andrea Wiggins, Alexandra Swanson, and Brooke Simmons. 2016. Assessing data quality in citizen science. *Frontiers in Ecology and the Environment*, 14(10):551–560.

Ji-Ung Lee, Jan-Christoph Klie, and Iryna Gurevych. 2022. Annotation Curricula to Implicitly Train Non-Expert Annotators. *Computational Linguistics*, pages 1–22.

Liza Lee, Mireille Desroches, Shamir Mukhi, and Christina Bancej. 2021. FluWatchers: Evaluation of a crowdsourced influenza-like illness surveillance application for Canadian influenza seasons 2015–2016 to 2018–2019. *Canada Communicable Disease Report*, 47(09):357–363.

Ronald D. Leeper, Jared Rennie, and Michael A. Palecki. 2015. Observational Perspectives from U.S. Climate Reference Network (USCRN) and Cooperative Observer Program (COOP) Network: Temperature and Precipitation Comparison. *Journal of Atmospheric and Oceanic Technology*, 32(4):703–721.

Chris J. Lintott, Kevin Schawinski, Anže Slosar, Kate Land, Steven Bamford, Daniel Thomas, M. Jordan Raddick, Robert C. Nichol, Alex Szalay, Dan Andreescu, Phil Murray, and Jan Vandenberg. 2008. Galaxy Zoo : Morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189.

Verena Lyding, Lionel Nicolas, and Alexander König. 2022. About the applicability of combining implicit crowdsourcing and language learning for the collection of NLP datasets. In *Proceedings of the 2nd Workshop on Novel Incentives in Data Collection from People: Models, Implementations, Challenges and Results within LREC 2022*, pages 46–57, Marseille, France.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

National Audubon Society. 2020. The Christmas bird count historical results.

An Nguyen, Byron Wallace, and Matthew Lease. 2015. Combining Crowd and Expert Labels Using Decision Theoretic Active Learning. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 3(1):120–129.

Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. 2014. The Language Demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics*, 2:79–92.

Christopher Phillips, Dylan Walshe, Karen O'Regan, Ken Strong, Christopher Hennon, Ken Knapp, Conor Murphy, and Peter Thorne. 2018. Assessing Citizen Science Participation Skill for Altruism or University Course Credit: A Case Study Analysis Using Cyclone Center. *Citizen Science: Theory and Practice*, 3(1):6.

Massimo Poesio, Jon Chamberlain, Udo Kruschwitz, Livio Robaldo, and Luca Ducceschi. 2013. Phrase detectives: Utilizing collective intelligence for internet-scale language resource creation. *ACM Transactions on Interactive Intelligent Systems*, 3(1):1–44.

Lisa M. Rasmussen and Caren Cooper. 2019. Citizen Science Ethics. *Citizen Science: Theory and Practice*, 4(1):5.

David B. Resnik, Kevin C. Elliott, and Aubrey K. Miller. 2015. A framework for addressing ethical issues in citizen science. *Environmental Science & Policy*, 54:475–481.

Dana Rotman, Jenny Preece, Jen Hammock, Kezee Procita, Derek Hansen, Cynthia Parr, Darcy Lewis, and David Jacobs. 2012. Dynamic changes in motivation in collaborative citizen-science projects. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work - CSCW '12*, page 217, Seattle, Washington, USA.

Boaz Shmueli, Jan Fell, Soumya Ray, and Lun-Wei Ku. 2021. Beyond Fair Pay: Ethical Implications of NLP Crowdsourcing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3758–3769, Online.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii.

Alexander Sorokin and David Forsyth. 2008. Utility data annotation with Amazon Mechanical Turk. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, Anchorage, AK, USA.

Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer. 2015. Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Scientific Data*, 2(1):1–14.

Christopher Tauchmann, Johannes Daxenberger, and Margot Mieskes. 2020. The Influence of Input Data Complexity on Crowdsourcing Quality. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*, pages 71–72, Cagliari, Italy.

Ginger Tsueng, Steven M. Nanis, Jennifer Fouquier, Benjamin M. Good, and Andrew I. Su. 2016. Citizen Science for Mining the Biomedical Literature. *Citizen Science: Theory and Practice*, 1(2):14.

Noortje J. Venhuizen, Valerio Basile, Kilian Evang, and Johan Bos. 2013. Gamification for word sense labeling. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 397–403, Potsdam, Germany.

L. von Ahn. 2006. Games with a Purpose. *Computer*, 39(6):92–94.

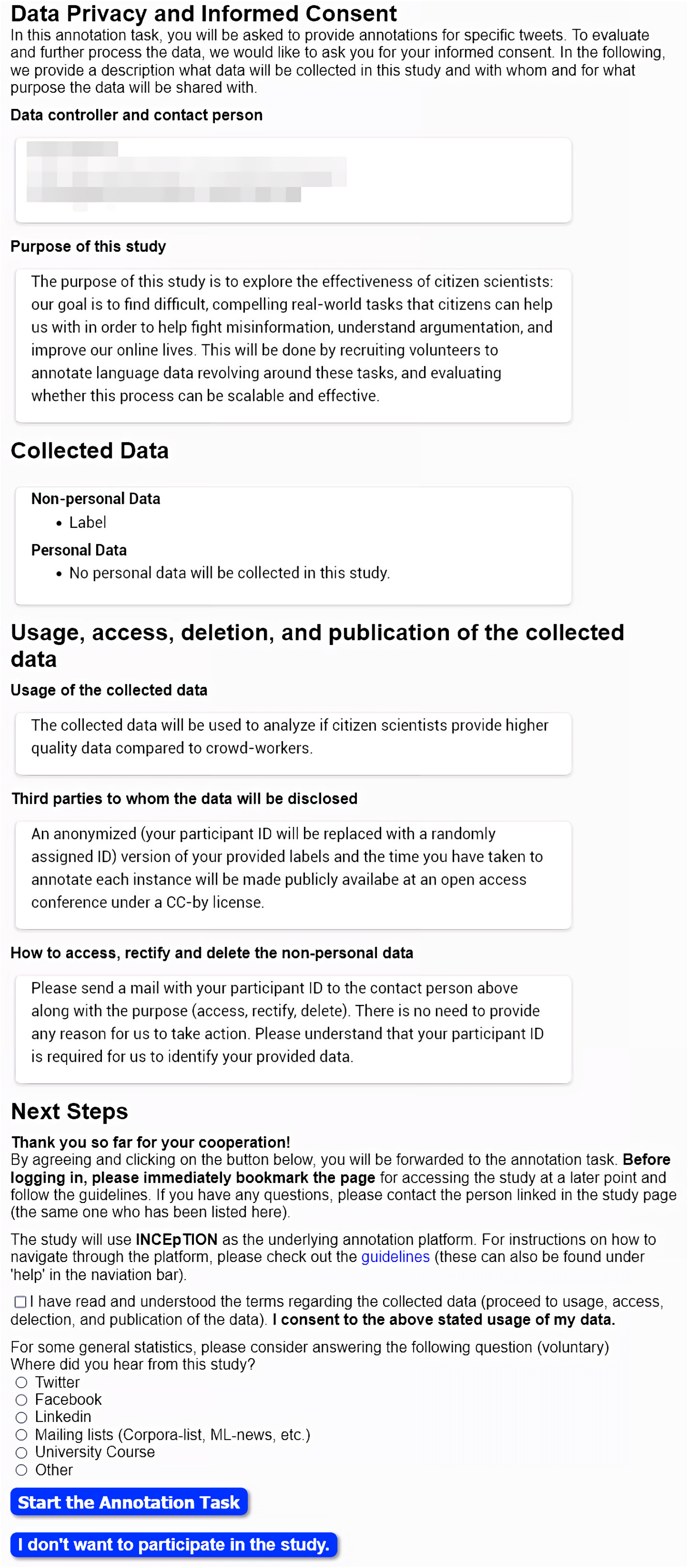
Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321(5895):1465–1468.

Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. 2013. Perspectives on crowdsourcing annotations for natural language processing. *Language Resources and Evaluation*, 47(1):9–31.

Sarah West, Alison Dyke, and Rachel Pateman. 2021. Variations in the Motivations of Environmental Citizen Scientists. *Citizen Science: Theory and Practice*, 6(1):14.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Appendix A Landing Page



Data Privacy and Informed Consent

In this annotation task, you will be asked to provide annotations for specific tweets. To evaluate and further process the data, we would like to ask you for your informed consent. In the following, we provide a description what data will be collected in this study and with whom and for what purpose the data will be shared with.

Data controller and contact person

[redacted]

Purpose of this study

The purpose of this study is to explore the effectiveness of citizen scientists: our goal is to find difficult, compelling real-world tasks that citizens can help us with in order to help fight misinformation, understand argumentation, and improve our online lives. This will be done by recruiting volunteers to annotate language data revolving around these tasks, and evaluating whether this process can be scalable and effective.

Collected Data

Non-personal Data

- Label

Personal Data

- No personal data will be collected in this study.

Usage, access, deletion, and publication of the collected data

Usage of the collected data

The collected data will be used to analyze if citizen scientists provide higher quality data compared to crowd-workers.

Third parties to whom the data will be disclosed

An anonymized (your participant ID will be replaced with a randomly assigned ID) version of your provided labels and the time you have taken to annotate each instance will be made publicly availabe at an open access conference under a CC-by license.

How to access, rectify and delete the non-personal data

Please send a mail with your participant ID to the contact person above along with the purpose (access, rectify, delete). There is no need to provide any reason for us to take action. Please understand that your participant ID is required for us to identify your provided data.

Next Steps

Thank you so far for your cooperation!
By agreeing and clicking on the button below, you will be forwarded to the annotation task. **Before logging in, please immediately bookmark the page** for accessing the study at a later point and follow the guidelines. If you have any questions, please contact the person linked in the study page (the same one who has been listed here).

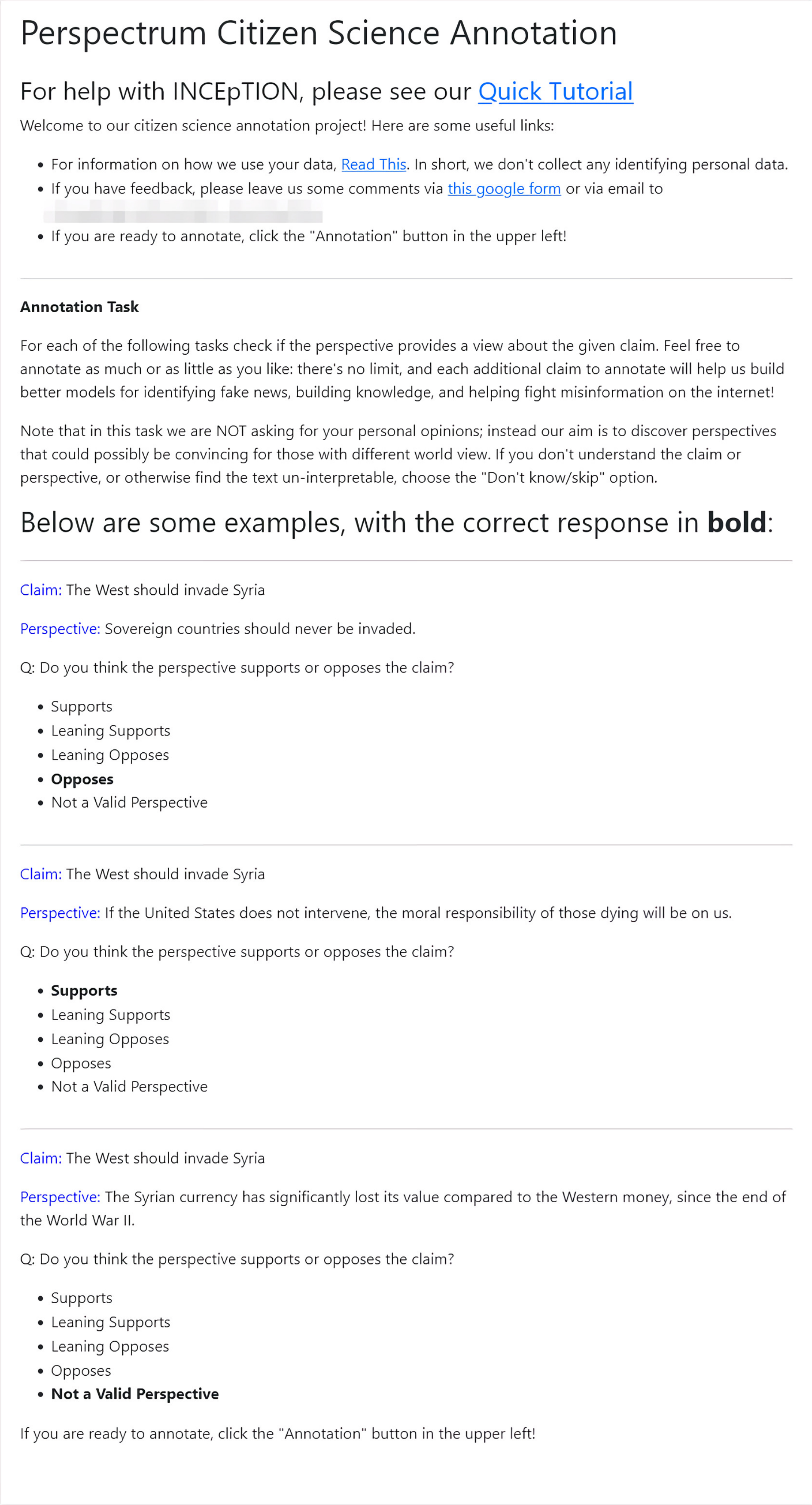
The study will use **INCEpTION** as the underlying annotation platform. For instructions on how to navigate through the platform, please check out the guidelines (these can also be found under 'help' in the naviation bar).

□ I have read and understood the terms regarding the collected data (proceed to usage, access, delection, and publication of the data). **I consent to the above stated usage of my data.**

For some general statistics, please consider answering the following question (voluntary)
Where did you hear from this study?
○ Twitter
○ Facebook
○ Linkedin
○ Mailing lists (Corpora-list, ML-news, etc.)
○ University Course
○ Other

Start the Annotation Task
I don't want to participate in the study.

Appendix B Annotation Guidelines



Perspectrum Citizen Science Annotation

For help with INCEpTION, please see our Quick Tutorial

Welcome to our citizen science annotation project! Here are some useful links:

- For information on how we use your data, Read This. In short, we don't collect any identifying personal data.
- If you have feedback, please leave us some comments via this google form or via email to [redacted]
- If you are ready to annotate, click the "Annotation" button in the upper left!

Annotation Task

For each of the following tasks check if the perspective provides a view about the given claim. Feel free to annotate as much or as little as you like: there's no limit, and each additional claim to annotate will help us build better models for identifying fake news, building knowledge, and helping fight misinformation on the internet!

Note that in this task we are NOT asking for your personal opinions; instead our aim is to discover perspectives that could possibly be convincing for those with different world view. If you don't understand the claim or perspective, or otherwise find the text un-interpretable, choose the "Don't know/skip" option.

Below are some examples, with the correct response in **bold**:

Claim: The West should invade Syria

Perspective: Sovereign countries should never be invaded.

Q: Do you think the perspective supports or opposes the claim?

- Supports
- Leaning Supports
- Leaning Opposes
- **Opposes**
- Not a Valid Perspective

Claim: The West should invade Syria

Perspective: If the United States does not intervene, the moral responsibility of those dying will be on us.

Q: Do you think the perspective supports or opposes the claim?

- **Supports**
- Leaning Supports
- Leaning Opposes
- Opposes
- Not a Valid Perspective

Claim: The West should invade Syria

Perspective: The Syrian currency has significantly lost its value compared to the Western money, since the end of the World War II.

Q: Do you think the perspective supports or opposes the claim?

- Supports
- Leaning Supports
- Leaning Opposes
- Opposes
- **Not a Valid Perspective**

If you are ready to annotate, click the "Annotation" button in the upper left!

Appendix C Questions to keep in mind for a citizen science project

- What is the purpose of the study?
- What kind of personal and non-personal data will be collected?[2]
- If there is a questionnaire involved, what questions will it involve?
- How will the data be used?
- Is a publication of the data planned and if so, which data will be published and will it be anonymized?
- How can participants request access, change, or deletion of their data?

Appendix D Project Statistics

D.1 Number of participants

In addition to the plots visualizing the number of participants (c.f. Fig. 3), we also list the raw numbers in Table 3.

Channel	Participants
Courses	14
Facebook	3
LinkedIn	4
Lists	55
Twitter	8
Undisclosed	17

Table 3: Number of participants per channel.

D.2 Annotation statistics

In addition to the plots visualizing the annotation counts and label distribution (c.f. Fig. 4), we also list the raw numbers in Table 4.

Table 4: Label distribution grouped by channel. Labels are *supports* (++), *mildly-supports* (+), *mildly-opposes* (-), *opposes* (--), *not a valid perspective* (I) and *Skip* (S).

Channel	Total	Counts						Percentage					
		+	++	-	--	I	S	+	++	-	--	I	S
Courses	307	18	108	24	104	28	25	5.86	35.18	7.82	33.88	9.12	8.14
Facebook	5	2	0	0	0	0	3	40.00	0.00	0.00	0.00	0.00	60.00
LinkedIn	21	1	5	1	7	3	4	4.76	23.81	4.76	33.33	14.29	19.05
Lists	830	98	264	48	222	92	106	11.81	31.81	5.78	26.75	11.08	12.77
Twitter	131	14	42	12	39	13	11	10.69	32.06	9.16	29.77	9.92	8.40
Undisclosed	187	18	53	15	52	27	22	9.63	28.34	8.02	27.81	14.44	11.76

[2] We provided some pre-defined suggestions such as *Name* or *IP* for personal data and *Label* for non-personal data with the possibility to add more in our landingpage module.

Part III

Epilogue

Chapter 12

Conclusion And Future Work

12.1 Conclusion

High-quality, annotated datasets are a crucial resource for many fields of science and downstream applications. Especially for machine learning, new large-scale datasets have enabled and fueled the rise of deep learning. The availability of annotated data is often a major roadblock as well as a limiting factor in performance. Creating datasets via annotation, however, requires tremendous manual effort and is expensive and time-consuming.

To alleviate these issues, with this thesis, we set out to improve two aspects of natural language dataset creation: annotation quality and annotation efficiency.

To approach this, we first introduced a novel annotation tool INCEpTION (Chapter 6). It an extensible, configurable, annotation tool that incorporates all the related tasks coming up during annotation projects into a joint web-based platform. Thus, it can provide a solid foundation for future annotation efficiency and quality experiments without re-inventing the wheel, thereby saving time and letting project creators focus on the annotations itself instead of first building an annotation editor. INCEpTION has been extensively used as the base for most of the publications that make up this thesis.

When training and evaluating machine learning models, the annotations must be of the highest quality to achieve reliable and accurate results. Recent work, however, has found that even frequently used state-of-the-art datasets still contain a non-negligible percentage of annotation errors and artifacts. Thus, as part of Chapter 7, we summarized best practices of quality management and analyzed how annotation quality management is conducted in practice. We derived best practices and recommendations for future dataset creation projects from these, for instance how to best structure the annotation process, how to select and manage annotators or how to estimate annotation quality.

Many algorithms for automatic annotation error detection have been devised over the years to reduce costs for finding annotation errors. Nevertheless, we found that methods were rarely evaluated on the same datasets and authors often used different metrics and task definitions, making comparisons difficult. To improve the situation, as part of Chapter 8, we formalized the task of automatic annotation error detection, re-implemented the most popular methods, and evaluated them on various tasks and datasets. Experiments on actual corpora have shown that AED methods still have room for improvement. While looking promising on artificial corpora, there is a large performance drop when applying them in practice.

Improving annotation efficiency, that is, reducing annotation time or cost, can enable dataset creators to either save effort or create larger datasets for the same effort. For this thesis, we proposed several ways to improve annotation efficiency, including human-in-the-loop label suggestions, interactive annotator training, and community annotation.

In Chapter 9, we proposed a novel method to reduce annotation time for entity linking in low-resource domains by leveraging human-in-the-loop label suggestions and adaptive re-ranking. Entity Linking, which means disambiguating entity mentions in a text against knowledge bases, is a complex and often tedious annotation task, especially for low-resource domains with noisy texts. In simulation and a user study, we significantly reduced annotation time when using our novel annotation support.

Dataset creation projects often require annotators to familiarize themselves with the task, its annotation scheme, and the data domain on the fly. This can be overwhelming initially, mentally taxing, and induce errors in the resulting annotations. In Chapter 10, we developed annotation curricula to train annotators implicitly while annotating. The core idea is ordering instances to be annotated according to a learning curriculum, for example, by perceived difficulty or estimated annotation time. In simulation and a user study, we showed that annotation time can be reduced compared to a random ordering without negatively impacting annotation quality.

Citizen science, the collaboration of volunteers from the general public with researchers to conduct science, is frequently used for data collection in fields like environmental science or astronomy. We adapted citizen science to natural language annotation as part of Chapter 11. By asking the community to re-annotate parts of an already existing, crowdsourced dataset, we showed that citizen science can, within limits, be a viable way for collecting annotations.

Through surveying the literature, re-implementing and evaluating existing methods, as well as introducing new methods, we were able to answer our research questions asking how annotation quality and efficiency can actually be improved. While being only a first step, this thesis already provides a solid foundation for many aspects of annotation quality and efficiency. For the future, we see the demand for high-quality datasets only rising, especially for expert annotated and narrow-domain, specialized datasets, highlighting the relevance of this thesis and the importance of future work on annotation quality and efficiency.

12.2 Future Work

We end this work by discussing interesting future research directions that can extend our work.

When analyzing annotation quality management in Chapter 7, we aimed to analyze how quality management of datasets was done in practice. Our analysis already yielded several interesting findings and common issues. We also were able to derive recommendations that future dataset creators can leverage for their own annotation campaigns. However, we did not analyze the impact these best practices have on the resulting dataset quality. Therefore, an interesting future research direction is to investigate not only what methods were used for quality management but also to quantify their impact on the resulting dataset quality. Our focus was also primarily on annotation; extending a similar analysis to text production and evaluation would certainly yield new insights and more targeted recommendations.

Label suggestions, as described in §5.1.2 and applied in Chapter 9, have been widely used to provide annotation support and thus reducing annotation time and effort. Interactive recommenders have been shown to be advantageous compared to static suggestions or pre-annotations. They are, however, limited as they usually retrain from scratch, which is expensive and time-consuming, thereby limiting responsiveness and adaptability. Using specialized continual learning or online learning methods for the task, which update recommenders instead of retraining, might provide additional benefits. The implicit feedback given by annotators using recommenders is also often unused; models are just retrained on the annotations made so far but ignore acceptance and rejection events. Another possible research direction would be learning from these events, for example, by negatively reinforcing models.

While automatic annotation error detection has already been used to find errors in noisy datasets, we see the following ways to extend their usefulness. When surveying the field as part of Chapter 8, we found that in practice, overwhelmingly flagger methods are used for finding errors. Flaggers give a dichotomous judgment of whether an instance is correct or not; their output does not require profound interpretation and gives a finite set of instances to inspect. For several reasons, scorers are only rarely leveraged. Scorers judge how likely an instance is wrong. When using scorers, it is unclear when to stop inspecting, as there is no natural stopping point. Often, a fixed percentage of the instances with the highest likeliness of being wrong are selected. Coming up with a stopping criterion similar to optimal stopping from statistics would be highly desirable. This criterion, for instance, could be based on the inspection history. If too many instances are shown to be correct once the inspection has progressed, then the error density might be too low now, and one should stop the round.

With Chapter 11, we have shown that citizen science is a potential way of creating datasets for (almost) free, but our conclusions were based on a single, relatively small study. We see the following directions for further research and evaluation to understand better which settings citizen science can be applicable and how to use it best. It would be interesting to conduct more projects and see which datasets are suitable as well as whether volunteers participate, even if there is no extrinsic motivation. Then, it can also be tested on how annotator retention develops, especially when projects run longer. The call for participation itself could also be investigated for its impact on turnout, motivation, and quality.

Finally, we find that despite the critical importance of high quality datasets, research addressing this topic is surprisingly scarce. As machine learning and data-driven approaches continue to gain prominence, ensuring the accuracy and reliability of annotations becomes increasingly crucial. Future work should prioritize investigating methods to enhance annotation quality, exploring innovative techniques, and establishing robust evaluation frameworks. The importance of annotation quality should also be further disseminated, for instance by organizing related workshops or talks. We hope that this thesis is one step in this direction. Therefore, we recommend that conference organizers and steering committees develop and adopt dataset quality management checklists to promote best practices and to better document a dataset's creation process.

Bibliography

Kelly Addy, Linda Green, Elizabeth Herron, and Kris Stepenuck. 2010. Why Volunteer Water Quality Monitoring Makes Sense. *US Department of Agriculture, Washington, DC.*

Sönke Ahrens. 2017. *How to Take Smart Notes: One Simple Technique to Boost Writing, Learning and Thinking: For Students, Academics and Nonfiction Book Writers.* CreateSpace, North Charleston, South Carolina, USA.

Bea Alex, Claire Grover, Rongzhou Shen, and Mijail Kabadjov. 2010. Agile corpus annotation in practice: An overview of manual and automatic annotation of CVs. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 29–37, Uppsala, Sweden.

Héctor Martínez Alonso and Lauren Romeo. 2014. Crowdsourcing as a preprocessing for complex semantic annotation tasks. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 229–234, Reykjavik, Iceland.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online.

Bharat Ram Ambati, Rahul Agarwal, Mridul Gupta, Samar Husain, and Dipti Misra Sharma. 2011. Error detection for treebank validation. In *Proceedings of the 9th Workshop on Asian Language Resources*, pages 23–30, Chiang Mai, Thailand.

Emmanuel Ameisen. 2020. *Building Machine Learning Powered Applications - Going from Idea to Product.* O'Reilly, Sebastopol, California, USA.

Hadi Amiri, Timothy Miller, and Guergana Savova. 2018. Spotting Spurious Data with Neural Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2006–2016, New Orleans, Louisiana, USA.

Sachi Angle, Pruthwik Mishra, and Dipti Mishra Sharma. 2018. Automated error correction and validation for POS tagging of Hindi. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, pages 11–18, Hong Kong, China.

Marianne Grace Araneta, Gülşen Eryiğit, Alexander König, Ji-Ung Lee, Ana Luís, Verena Lyding, Lionel Nicolas, Christos Rodosthenous, and Federico Sangati. 2020. Substituto – A Synchronous Educational Language Game for Simultaneous Teaching and Crowdsourcing. In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 1–9, Gothenburg, Sweden.

Lora Aroyo and Chris Welty. 2015. Truth Is a Lie: Crowd Truth and the Seven Myths of Human Annotation. *AI Magazine*, 36(1):15–24.

Imanol Arrieta-Ibarra, Leonard Goff, Diego Jiménez-Hernández, Jaron Lanier, and E. Glen Weyl. 2018. Should We Treat Data as Labor? Moving Beyond “Free”. *AEA Papers and Proceedings*, 108:38–42.

Ron Artstein and Massimo Poesio. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.

Vino Avanesi and Jan Teurlings. 2022. “I’m not a robot,” or am I?: Micro-labor and the immanent subsumption of the social in the human computation of ReCAPTCHAs. *International Journal of Communication*, 16(0):1–19.

Jeremy Barnes, Lilja Øvrelid, and Erik Velldal. 2019. Sentiment Analysis Is Not Solved! Assessing and Probing Sentiment Classification. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 12–23, Florence, Italy.

Tilman Beck, Ji-Ung Lee, Christina Viehmann, Marcus Maurer, Oliver Quiring, and Iryna Gurevych. 2021. Investigating label suggestions for opinion mining in German Covid-19 social media. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–13, Online.

Heike Behrens. 2008. *Corpora in Language Acquisition Research: History, Methods, Perspectives*, volume 6 of *Trends in Language Acquisition Research.* John Benjamins Publishing Company, Amsterdam, The Netherlands.

Emily M. Bender and Batya Friedman. 2018. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48, Montreal, Canada.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 4356–4364, Barcelona, Spain.

Kalina Bontcheva, Ian Roberts, Leon Derczynski, and Dominic Rout. 2014. The GATE Crowdsourcing Plugin: Crowdsourcing Annotated Corpora Made Easy. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 97–100, Gothenburg, Sweden.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal.

Adriane Boyd, Markus Dickinson, and W. Detmar Meurers. 2008. On Detecting Errors in Dependency Treebanks. *Research on Language and Computation*, 6(2):113–137.

Alex Brandsen, Suzan Verberne, Milco Wansleeben, and Karsten Lambers. 2020. Creating a Dataset for Named Entity Recognition in the Archaeology Domain. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4573–4577, Marseille, France.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41, Sozopol, Bulgaria.

Chris Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 286–295, Singapore.

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12, Los Angeles, California, USA.

Jean Carletta. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada.

Jhonny Cerezo, Felipe Bravo-Marquez, and Alexandre Henri Bergel. 2021. Tools Impact on the Quality of Annotations for Chat Untangling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 215–220, Online.

Jon Chamberlain, Karën Fort, Udo Kruschwitz, Mathieu Lafourcade, and Massimo Poesio. 2013. Using Games to Create Language Resources: Successes and Limitations of the Approach. In *The People's Web Meets NLP*, pages 3–44. Springer, Berlin, Heidelberg.

Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. 2008. Addressing the Resource Bottleneck to Create Large-Scale Annotated Texts. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 375–380, Venice, Italy.

Alessandro Checco, Kevin Roitero, Eddy Maddalena, Stefano Mizzaro, and Gianluca Demartini. 2017. Let's Agree to Disagree: Fixing Agreement Measures for Crowdsourcing. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, pages 11–20, San Francisco, California, USA.

Derek Chen, Yu, and Zhou Samuel R. Bowman. 2022. Clean or Annotate: How to Spend a Limited Data Collection Budget. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 152–168, Online and Seattle, Washington, USA.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2023. AlpaGasus: Training A Better Alpaca with Fewer Data. *arXiv*.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. Seeing Things from a Different Angle: Discovering Diverse Perspectives about Claims. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 542–557, Minneapolis, Minnesota, USA.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A Dataset of Numerical Reasoning over Financial Data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic.

Hao-Fei Cheng, Logan Stapleton, Anna Kawakami, Venkatesh Sivaraman, Yanghuidi Cheng, Diana Qing, Adam Perer, Kenneth Holstein, Zhiwei Steven Wu, and Haiyi Zhu. 2022. How Child Welfare Workers Reduce Racial Disparities in Algorithmic Decisions. In *CHI Conference on Human Factors in Computing Systems*, pages 1–22, New Orleans, Louisiana, USA.

Fu-Dong Chiou, David Chiang, and Martha Palmer. 2001. Facilitating Treebank Annotation Using a Statistical Parser. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–4, San Diego, California, USA.

Wen-Chi Chou, Richard Tzong-Han Tsai, Ying-Shan Su, Wei Ku, Ting-Yi Sung, and Wen-Lian Hsu. 2006. A semi-automatic method for annotating a biomedical Proposition Bank. In *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*, pages 5–12, Sydney, Australia.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Sandipan Dandapat, Priyanka Biswas, Monojit Choudhury, and Kalika Bali. 2009. Complex linguistic annotation – no easy way out! A case from Bangla and Hindi POS labeling tasks. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 10–18, Suntec, Singapore.

Sanjoy Dasgupta and Daniel Hsu. 2008. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine. Learning*, pages 208–215, Helsinki, Finland.

A. P. Dawid and A. M. Skene. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied Statistics*, 28(1):20–28.

JC de Borda. 1781. M'emoire sur les' elections au scrutin. *Histoire de l'Acad'emie Royale des Sciences.*

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, USA.

Markus Dickinson. 2005. *Error Detection and Correction in Annotated Corpora.* Ph.D. thesis, The Ohio State University.

Markus Dickinson and Chong Min Lee. 2008. Detecting errors in semantic annotation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 605–610, Marrakech, Morocco.

Markus Dickinson and W. Detmar Meurers. 2003a. Detecting Errors in Part-of-Speech Annotation. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 107–114, Budapest, Hungary.

Markus Dickinson and W. Detmar Meurers. 2003b. Detecting inconsistencies in treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, pages 1–12, Växjö, Sweden.

Markus Dickinson and W. Detmar Meurers. 2005. Detecting errors in discontinuous structural annotation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 322–329, Ann Arbor, Michigan, USA.

Dmitriy Dligach and Martha Palmer. 2011. Reducing the need for double annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 65–73, Portland, Oregon, USA.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *Proceedings of the 2019 Conference of the North*, pages 2368–2378, Minneapolis, Minnesota.

Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. 2001. Rank aggregation methods for the Web. In *Proceedings of the Tenth International Conference on World Wide Web - WWW '01*, pages 613–622, Hong Kong, China.

Robert L. Ebel. 1951. Estimation of the reliability of ratings. *Psychometrika*, 16(4):407–424.

Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan.

Hanne Martine Eckhoff and Aleksandrs Berdičevskis. 2016. Automatic parsing as an efficient pre-annotation tool for historical texts. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 62–70, Osaka, Japan.

Christopher Edwards, Heather Allen, and Crispen Chamunyonga. 2021. Correlation does not imply agreement: A cautionary tale for researchers and reviewers. *Sonography*, 8(4):185–190.

GülŞen Eryiğit, Ali Şentaş, and Johanna Monti. 2022. Gamified crowdsourcing for idiom corpora construction. *Natural Language Engineering*, 1(1):1–33.

Meng Fang, Jie Yin, and Dacheng Tao. 2014. Active learning for crowdsourcing using knowledge transfer. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1809–1815, Québec City, Québec, Canada.

Paul Felt, Eric K. Ringger, Kevin Seppi, Kristian S. Heal, Robbie A. Haertel, and Deryle Lonsdale. 2014. Evaluating Machine-Assisted Annotation in Under-Resourced Settings. *Language Resources and Evaluation*, 48(4):561–599.

Ailbhe Finnerty, Pavel Kucherbaev, Stefano Tranquillini, and Gregorio Convertino. 2013. Keep it simple: Reward and task design in crowdsourcing. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI*, pages 1–4, Trento Italy.

Roland A. Fisher. 1925. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-Scale QA-SRL Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2060, Melbourne, Australia.

James Fiumara, Christopher Cieri, Jonathan Wright, and Mark Liberman. 2020. LanguageARC: Developing Language Resources Through Citizen Linguistics. In *Proceedings of the LREC 2020 Workshop on "Citizen Linguistics in Language Resource Development"*, pages 1–6, Marseille, France.

Joseph L. Fleiss. 1971. Measuring Nominal Scale Agreement among Many Raters. *Psychological Bulletin*, 76(5):378–382.

Joseph L. Fleiss, Bruce Levin, and Myunghee Cho Paik. 2003. *Statistical Methods for Rates and Proportions*, 1 edition. Wiley Series in Probability and Statistics. Wiley.

Karën Fort, Gilles Adda, and K. Bretonnel Cohen. 2011. Amazon Mechanical Turk: Gold Mine or Coal Mine? *Computational Linguistics*, 37(2):413–420.

Karën Fort, Maud Ehrmann, and Adeline Nazarenko. 2009. Vers une méthodologie d'annotation des entités nommées en corpus ? In *Actes de la 16ème conférence sur le Traitement Automatique des Langues Naturelles. Articles longs*, pages 281–290, Senlis, France.

Karën Fort, Bruno Guillaume, and Hadrien Chastant. 2014. Creating Zombilingo, a Game With A Purpose for dependency syntax annotation. In *Proceedings of the First International Workshop on Gamification for Information Retrieval*, pages 2–6, Amsterdam, The Netherlands.

Karën Fort, Aurélie Névéol, Yoann Dupont, and Julien Bezançon. 2022. Use of a Citizen Science Platform for the Creation of a Language Resource to Study Bias in Language Models for French: A Case Study. In *Proceedings of the 2nd Workshop on Novel Incentives in Data Collection from People: Models, Implementations, Challenges and Results within LREC 2022*, pages 8–13, Marseille, France.

Karën Fort and Benoît Sagot. 2010. Influence of Pre-annotation on POS-tagged Corpus Development. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 56–63, Uppsala, Sweden.

W. N. Francis and H. Kucera. 1979. Brown Corpus Manual. Technical report, Providence, Rhode Island, USA.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1050–1059, New York City, USA.

Kuzman Ganchev, Fernando Pereira, Mark Mandel, Steven Carroll, and Peter White. 2007. Semi-Automated Named Entity Annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 53–56, Prague, Czech Republic.

Robert Greinacher and Franziska Horn. 2018. The DALPHI annotation framework & how its pre-annotations can improve annotator efficiency. *arXiv.*

Andreas Grivas, Beatrice Alex, Claire Grover, Richard Tobin, and William Whiteley. 2020. Not a cute stroke: Analysis of Rule- and Neural Network-based Information Extraction Systems for Brain Radiology Reports. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 24–37, Online.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online.

Mordechai Haklay, Daniel Dörler, Florian Heigl, Marina Manzoni, Susanne Hecker, and Katrin Vohland. 2021. What Is Citizen Science? The Challenges of Definition. In Katrin Vohland, Anne Land-Zandstra, Luigi Ceccaroni, Rob Lemmens, Josep Perelló, Marisa Ponti, Roeland Samson, and Katherin Wagenknecht, editors, *The Science of Citizen Science*, pages 13–33. Springer International Publishing, Cham.

Moritz Hardt and Benjamin Recht. 2022. *Patterns, Predictions, and Actions: Foundations of Machine Learning.* Princeton University Press, Princeton Oxford.

Christopher Harris. 2011. Youre hired! An examination of crowdsourcing incentive models in human resource tasks. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 15–18, Hong Kong, China.

Boris Haselbach, Kerstin Eckart, Wolfgang Seeker, Kurt Eberle, and Ulrich Heid. 2012. Approximating Theoretical Linguistics Classification in Real Data: The Case of German "nach" Particle Verbs. In *Proceedings of COLING 2012*, pages 1113–1128, Mumbai, India.

Andrew F. Hayes and Klaus Krippendorff. 2007. Answering the Call for a Standard Reliability Measure for Coding Data. *Communication Methods and Measures*, 1(1):77–89.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: A Chinese Machine Reading Comprehension Dataset from Real-world Applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia.

Dan Hendrycks and Kevin Gimpel. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *Proceedings of International Conference on Learning Representations*, pages 1–12, Toulon, France.

Mireille Hildebrandt. 2019. Privacy as Protection of the Incomputable Self: From Agnostic to Agonistic Machine Learning. *Theoretical Inquiries in Law*, 20(1):83–121.

Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. 2015. Incentivizing High Quality Crowdwork. In *Proceedings of the 24th International Conference on World Wide Web*, pages 419–429, Florence Italy.

Nora Hollenstein, Nathan Schneider, and Bonnie Webber. 2016. Inconsistency detection in semantic annotation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3986–3990, Portorož, Slovenia.

Anthony M Holmes. 1991. *The Ontario Butterfly Atlas*. Entomologists' Association, Toronto, Canada.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia, USA.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 377–382, Baltimore, Maryland, USA.

Eduard Hovy and Julia Lavid. 2010. Towards a 'Science' of Corpus Annotation: A New Methodological Challenge for Corpus Linguistics. *International Journal of Translation Studies*, 22:13–36.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. 2009. Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, Boulder, Colorado, USA.

Charles I. Jones and Christopher Tonetti. 2020. Nonrivalry and the Economics of Data. *American Economic Review*, 110(9):2819–2858.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher Manning. 2021. Mind Your Outliers! Investigating the Negative Impact of Outliers on Active Learning for Visual Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7265–7281, Online.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations (ICLR)*, pages 1–13, Online.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico, USA.

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2023a. Analyzing Dataset Annotation Quality Management in the Wild.

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2020. From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6982–6993, Online.

Jan-Christoph Klie, Ji-Ung Lee, Kevin Stowe, Gözde Şahin, Nafise Sadat Moosavi, Luke Bates, Dominic Petrak, Richard Eckart De Castilho, and Iryna Gurevych. 2023b. Lessons Learned from a Citizen Science Project for Natural Language Processing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3594–3608, Dubrovnik, Croatia.

Jan-Christoph Klie, Bonnie Webber, and Iryna Gurevych. 2023c. Annotation Error Detection: Analyzing the Past and Present for a More Coherent Future. *Computational Linguistics*, 49(1):157–198.

Karl J. Krahnke and Stephen D. Krashen. 1983. Principles and Practice in Second Language Acquisition. *TESOL Quarterly*, 17(2):300.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan Van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa De Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to Its Methodology.* SAGE, Los Angeles.

Klaus Krippendorff. 2004. Reliability in Content Analysis.: Some Common Misconceptions and Recommendations. *Human Communication Research*, 30(3):411–433.

Klaus Krippendorff, Yann Mathet, Stéphane Bouvry, and Antoine Widlöcher. 2016. On the reliability of unitizing textual continua: Further developments. *Quality & Quantity*, 50(6):2347–2364.

Anand Kulkarni, Matthew Can, and Björn Hartmann. 2012. Collaboratively Crowdsourcing Workflows with Turkomatic. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, pages 1003–1012, Seattle, Washington, USA.

Jonathan K. Kummerfeld. 2019. SLATE: A Super-Lightweight Annotation Tool for Experts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12, Florence, Italy.

Jonathan K. Kummerfeld. 2021. Quantifying and Avoiding Unfair Qualification Labour in Crowdsourcing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 343–349, Online.

Pavel Květoň and Karel Oliva. 2002. (Semi-)Automatic Detection of Errors in PoS-Tagged Corpora. In *COLING 2002: The 19th International Conference on Computational Linguistics*, pages 1–7, Taipei, Taiwan.

Stefan Larson, Adrian Cheung, Anish Mahendran, Kevin Leach, and Jonathan K. Kummerfeld. 2020. Inconsistencies in Crowdsourced Slot-Filling Annotations: A Typology and Identification Methods. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5035–5046, Online.

Stefan Larson, Anish Mahendran, Andrew Lee, Jonathan K. Kummerfeld, Parker Hill, Michael A. Laurenzano, Johann Hauswald, Lingjia Tang, and Jason Mars. 2019. Outlier Detection for Improved Data Quality and Diversity in Dialog Systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association* *for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 517–527, Minneapolis, Minnesota, USA.

Matthew Lease. 2011. On Quality Control and Machine Learning in Crowdsourcing. In *Proceedings of the 11th AAAI Conference on Human Computation*, AAAIWS'11-11, pages 97–102, San Francisco, California, USA.

Ji-Ung Lee, Jan-Christoph Klie, and Iryna Gurevych. 2022. Annotation Curricula to Implicitly Train Non-Expert Annotators. *Computational Linguistics*, 48(2):343–373.

Ji-Ung Lee, Christian M. Meyer, and Iryna Gurevych. 2020. Empowering Active Learning to Jointly Optimize System and User Demands. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4233–4247, Online.

Liza Lee, Mireille Desroches, Shamir Mukhi, and Christina Bancej. 2021. FluWatchers: Evaluation of a crowdsourced influenza-like illness surveillance application for Canadian influenza seasons 2015-2016 to 2018-2019. *Canada Communicable Disease Report*, 47(09):357–363.

Ronald D. Leeper, Jared Rennie, and Michael A. Palecki. 2015. Observational Perspectives from U.S. Climate Reference Network (USCRN) and Cooperative Observer Program (COOP) Network: Temperature and Precipitation Comparison. *Journal of Atmospheric and Oceanic Technology*, 32(4):703–721.

Sebastian Leitner. 1974. *So Lernt Man Leben [How to Learn to Live]*. Droemer-Knaur, Munich.

David D. Lewis and Jason Catlett. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, pages 148–156, San Francisco, California, USA.

R. Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology*, 22 140:55–55.

Christopher Lin, M Mausam, and Daniel Weld. 2016. Re-Active Learning: Active Learning with Relabeling. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1845–1852, Washington, D.C., USA.

Todd Lingren, Louise Deleger, Katalin Molnar, Haijun Zhai, Jareen Meinzen-Derr, Megan Kaiser, Laura Stoutenborough, Qi Li, and Imre Solti. 2014. Evaluating the impact of pre-annotation on annotation speed and potential bias: Natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *Journal of the American Medical Informatics Association : JAMIA*, 21(3):406–413.

Chris J. Lintott, Kevin Schawinski, Anže Slosar, Kate Land, Steven Bamford, Daniel Thomas, M. Jordan Raddick, Robert C. Nichol, Alex Szalay, Dan Andreescu, Phil Murray, and Jan Vandenberg. 2008. Galaxy Zoo : Morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189.

Hrafn Loftsson. 2009. Correcting a POS-Tagged corpus using three complementary methods. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 523–531, Athens, Greece.

Matthew Lombard, Jennifer Snyder-Duch, and Cheryl Campanella Bracken. 2002. Content Analysis in Mass Communication: Assessment and Reporting of Intercoder Reliability. *Human Communication Research*, 28(4):587–604.

F.M. Lord, M.R. Novick, and Allan Birnbaum. 1968. *Statistical Theories of Mental Test Scores*. Addison-Wesley, Oxford, England.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical Obstacles to Deploying Active Learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 21–30, Hong Kong, China.

Tingming Lu, Man Zhu, Zhiqiang Gao, and Yaocheng Gui. 2016. Evaluating ensemble based pre-annotation on named entity corpus construction in English and Chinese. In *Proceedings of the Third International Workshop on Worldwide Language Service Infrastructure and Second Workshop on Open Infrastructures and Analysis Frameworks for Human Language Technologies (WLSI/OIAF4HLT2016)*, pages 56–60, Osaka, Japan.

Yong Luo, Huaizheng Zhang, Yonggang Wen, and Xinwen Zhang. 2019. ResumeGAN: An Optimized Deep Representation Learning Framework for Talent-Job Fit via Adversarial Learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1101–1110, Beijing, China.

Verena Lyding, Lionel Nicolas, and Alexander König. 2022. About the applicability of combining implicit crowdsourcing and language learning for the collection of NLP datasets. In *Proceedings of the 2nd Workshop on Novel Incentives in Data Collection from People: Models, Implementations, Challenges and Results within LREC 2022*, pages 46–57, Marseille, France.

David J. C. MacKay. 1992. Information-Based Objective Functions for Active Data Selection. *Neural Computation*, 4(4):590–604.

Saket Maheshwary and Hemant Misra. 2018. Matching Resumes to Jobs via Deep Siamese Network. In *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*, pages 87–88, Lyon, France.

Christopher D. Manning. 2011. Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In *CICLing'11: Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I*, volume 6608, pages 171–189, Tokyo, Japan.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy.

Pierre André Ménard and Antoine Mougeot. 2019. Turning Silver into Gold: Error-Focused Corpus Reannotation with Active Learning. In *Proceedings - Natural Language Processing in a Deep Learning World*, pages 758–767, Varna, Bulgaria.

Marie-Jean Meurs, Caitlin Murphy, Nona Naderi, Ingo Morgenstern, Carolina Cantu, Shary Semarjit, Greg Butler, Justin Powlowski, Adrian Tsang, and René Witte. 2011. Towards Evaluating the Impact of Semantic Support for Curating the Fungus Scientic Literature. In *China Semantic Web Symposium Proceedings*, pages 34–39, Hangzhou, China.

Marie Mikulová, Milan Straka, Jan Štěpánek, Barbora Štěpánková, and Jan Hajic. 2022. Quality and efficiency of manual annotation: Pre-annotation bias. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2909–2918, Marseille, France.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore.

Robert Monarch. 2021. *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-Centered AI.* Manning Publications.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. Doccano: Text Annotation Tool for Human.

Julia Nantke and Frederik R. N. Schlupkothen, editors. 2020. *Annotations in Scholarly Editions and Research: Functions, Differentiation, Systematization.* De Gruyter, Boston, Massachusetts, USA.

National Audubon Society. 2020. The Christmas bird count historical results.

David F. Nettleton, Albert Orriols-Puig, and Albert Fornells. 2010. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306.

Kimberly A. Neuendorf. 2016. *The Content Analysis Guidebook.* SAGE, Thousand Oaks, California, USA.

Aurélie Névéol, Rezarta Islamaj Doğan, and Zhiyong Lu. 2011. Semi-automatic semantic annotation of PubMed queries: A study on quality, efficiency, satisfaction. *Journal of Biomedical Informatics*, 44(2):310–318.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A Reading Comprehension Dataset of Temporal Ordering Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172, Online.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. 2021a. Confident Learning: Estimating Uncertainty in Dataset Labels. *Journal of Artificial Intelligence Research*, 70:1373–1411.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. 2021b. Pervasive label errors in test sets destabilize machine learning benchmarks. In *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, pages 1–13, Online.

Philip Ogren, Guergana Savova, and Christopher Chute. 2008. Constructing Evaluation Corpora for Automated Clinical Named Entity Recognition. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 3143–2150, Marrakech, Morocco.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 1–15.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does Putting a Linguist in the Loop Improve NLU Data Collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic.

Rebecca J. Passonneau and Bob Carpenter. 2014. The Benefits of a Model of Annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326.

Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian Models of Annotation. *Transactions of the Association for Computational Linguistics*, 6(0):571–585.

Eyal Peer, Laura Brandimarte, Sonam Samat, and Alessandro Acquisti. 2017. Beyond the Turk: Alternative platforms for crowdsourcing behavioral research. *Journal of Experimental Social Psychology*, 70:153–163.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy.

Massimo Poesio, Jon Chamberlain, Udo Kruschwitz, Livio Robaldo, and Luca Ducceschi. 2013. Phrase detectives: Utilizing collective intelligence for internet-scale language resource creation. *ACM Transactions on Interactive Intelligent Systems*, 3(1):1–44.

David Powers. 2011. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

J. Pustejovsky and Amber Stubbs. 2013. *Natural Language Annotation for Machine Learning*. O'Reilly Media, Sebastopol, California, USA.

Kun Qian, Ahmad Beirami, Zhouhan Lin, Ankita De, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. 2021. Annotation inconsistency and entity bias in MultiWOZ. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 326–337, Online.

Priya Ranganathan, Cs Pramesh, and Rakesh Aggarwal. 2017. Common pitfalls in statistical analysis: Measures of agreement. *Perspectives in Clinical Research*, 8(4):187–191.

Lisa M. Rasmussen and Caren Cooper. 2019. Citizen Science Ethics. *Citizen Science: Theory and Practice*, 4(1):1–3.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282.

Ines Rehbein. 2014. POS error detection in automatically annotated corpora. In *Proceedings of LAW VIII - the 8th Linguistic Annotation Workshop*, pages 20–28, Dublin, Ireland.

Ines Rehbein, Josef Ruppenhofer, and Caroline Sporleder. 2009. Assessing the benefits of partial automatic pre-labeling for frame-semantic annotation. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 19–26, Suntec, Singapore.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3980–3990.

Frederick Reiss, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger. 2020. Identifying Incorrect Labels in the CoNLL-2003 Corpus. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 215–226, Online.

David B. Resnik, Kevin C. Elliott, and Aubrey K. Miller. 2015. A framework for addressing ethical issues in citizen science. *Environmental Science & Policy*, 54:475–481.

Eric Ringger, Marc Carmen, Robbie Haertel, Kevin Seppi, Deryle Lonsdale, Peter McClanahan, James Carroll, and Noel Ellison. 2008. Assessing the costs of machine-assisted corpus annotation through a user study. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 3318–3324, Marrakech, Morocco.

Rowena Rodrigues. 2020. Legal and human rights issues of AI: Gaps, challenges and vulnerabilities. *Journal of Responsible Technology*, 4:1–12.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation Examples are not Equally Informative: How should that change NLP Leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4486–4503, Online.

Sophie Rosset, Cyril Grouin, Thomas Lavergne, Mohamed Ben Jannet, Jérémy Leixa, Olivier Galibert, and Pierre Zweigenbaum. 2013. Automatic named entity pre-annotation for out-of-domain human annotation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 168–177, Sofia, Bulgaria.

Dana Rotman, Jenny Preece, Jen Hammock, Kezee Procita, Derek Hansen, Cynthia Parr, Darcy Lewis, and David Jacobs. 2012. Dynamic changes in motivation in collaborative citizen-science projects. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work - CSCW '12*, pages 217–226, Seattle, Washington, USA.

Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 859–866, Reykjavik, Iceland.

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Kumar Paritosh, and Lora Mois Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *SIGCHI*, pages 1–21.

Agata Savary, Jakub Waszczuk, and Adam Przepiórkowski. 2010. Towards the Annotation of Named Entities in the National Corpus of Polish. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 3622–3629, Valletta, Malta.

Daniel Schlagwein, Dubravka Cecez-Kecmanovic, and Benjamin Hanckel. 2019. Ethical norms and issues in crowdsourcing practices: A Habermasian analysis. *Information Systems Journal*, 29(4):811–837.

Susan Schreibman, Ray Siemens, and John Unsworth, editors. 2004. *A Companion to Digital Humanities.* Blackwell Publishing Ltd, Malden, Massachusetts, USA.

Claudia Schulz, Christian M. Meyer, Jan Kiesewetter, Michael Sailer, Elisabeth Bauer, Martin R. Fischer, Frank Fischer, and Iryna Gurevych. 2019. Analysis of Automatic Annotation Suggestions for Hard Discourse-Level Tasks in Expert Domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2761–2772, Florence, Italy.

William A. Scott. 1955. Reliability of content analysis: The case of nominal scale coding. *The Public Opinion Quarterly*, 19(3):321–325.

Burr Settles. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, San Rafael, California, USA.

Artem Shelmanov, Evgenii Tsymbalov, Dmitri Puzyrev, Kirill Fedyanin, Alexander Panchenko, and Maxim Panov. 2021. How Certain is Your Transformer? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1833–1840, Online.

Boaz Shmueli, Jan Fell, Soumya Ray, and Lun-Wei Ku. 2021. Beyond Fair Pay: Ethical Implications of NLP Crowdsourcing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3758–3769, Online.

Rainer Simon, Elton Barker, Leif Isaksen, and Pau de Soto. 2015. Linking early geospatial documents, one place at a time: Annotation of geographic documents with Recogito. *e-Perimetron*, 10(2):49–59.

Edwin D. Simpson and Iryna Gurevych. 2019. A Bayesian Approach for Sequence Tagging with Crowds. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1093–1104, Hong Kong, China.

Arne Skjærholt. 2011. More, Faster: Accelerated Corpus Annotation with Statistical Taggers. *Journal for Language Technology and Computational Linguistics*, 26(2):151–163.

Alisa Smirnova and Philippe Cudré-Mauroux. 2019. Relation Extraction Using Distant Supervision: A Survey. *ACM Computing Surveys*, 51(5):1–35.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, USA.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: A web-based tool for NLP-Assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France.

George Stoica, Emmanouil Antonios Platanios, and Barnabas Poczos. 2021. Re-TACRED: Addressing Shortcomings of the TACRED Dataset. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence 2021*, pages 13843–13850, Online.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A Corpus for Reasoning about Natural Language Grounded in Photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 843–852, Venice, Italy.

Simon Suster, Stephan Tulkens, and Walter Daelemans. 2017. A Short Review of Ethical Challenges in Clinical Natural Language Processing. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 80–87, Valencia, Spain.

Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer. 2015. Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Scientific Data*, 2(1):1–14.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online.

George Szpiro. 2010. *Numbers Rule: The Vexing Mathematics of Democracy, from Plato to the Present.* Princeton University Press.

Takaaki Tanaka, Francis Bond, Stephan Oepen, and Sanae Fujita. 2005. High precision treebanking: Blazing useful trees using POS information. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, pages 330–337, Ann Arbor, Michigan, USA.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, Edmonton, Alberta, Canada.

Ginger Tsueng, Steven M. Nanis, Jennifer Fouquier, Benjamin M. Good, and Andrew I. Su. 2016. Citizen Science for Mining the Biomedical Literature. *Citizen Science: Theory and Practice*, 1(2):1–11.

Morgan Ulinski, Julia Hirschberg, and Owen Rambow. 2016. Incrementally learning a dependency parser to support language documentation in field linguistics. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 440–449, Osaka, Japan.

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from Disagreement: A Survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Özlem Uzuner, Imre Solti, Fei Xia, and Eithon Cadag. 2010. Community annotation experiment for ground truth generation for the i2b2 medication challenge. *Journal of the American Medical Informatics Association*, 17(5):519–523.

Șerban Vădineanu, Daniel Pelt, Oleh Dzyubachyk, and Joost Batenburg. 2022. An Analysis of the Impact of Annotation Errors on the Accuracy of Deep Learning for Cell Segmentation. In *Proceedings of Machine Learning Research*, pages 1251–1267, Honolulu, Hawaii, USA.

Hans van Halteren. 2000. The Detection of Inconsistency in Manually Tagged Text. In *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, pages 48–55, Luxembourg.

K.J. van Stralen, F.W. Dekker, C. Zoccali, and K.J. Jager. 2012. Measuring Agreement, More Complicated Than It Seems. *Nephron Clinical Practice*, 120(3):162–167.

Vijay Vasudevan, Benjamin Caine, Raphael Gontijo-Lopes, Sara Fridovich-Keil, and Rebecca Roelofs. 2022. When does dough become a bagel? Analyzing the remaining mistakes on ImageNet. In *Proceedings of the 36th Conference on Neural Information Processing Systems*, pages 1–15, New Orleans, Louisiana, USA.

Noortje J. Venhuizen, Valerio Basile, Kilian Evang, and Johan Bos. 2013. Gamification for word sense labeling. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 397–403, Potsdam, Germany.

Marc Verhagen. 2010. The Brandeis Annotation Tool. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 3638–3643, Valletta, Malta.

Veniamin Veselovsky, Manoel Horta Ribeiro, and Robert West. 2023. Artificial Artificial Artificial Intelligence: Crowd Workers Widely Use Large Language Models for Text Production Tasks. *arXiv*.

Andreas Vlachos. 2006. Active annotation. In *Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 64–71, Trento, Italy.

L. von Ahn. 2006. Games with a Purpose. *Computer*, 39(6):92–94.

Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321(5895):1465–1468.

L. S. Vygotsky. 1980. *Mind in Society: Development of Higher Psychological Processes*. Harvard University Press.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training Named Entity Tagger from Imperfect Annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5153–5162, Hong Kong, China.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, pages 1–46, Online.

Sarah West, Alison Dyke, and Rachel Pateman. 2021. Variations in the Motivations of Environmental Citizen Scientists. *Citizen Science: Theory and Practice*, 6(1):1–18.

Mark E. Whiting, Grant Hugh, and Michael S. Bernstein. 2019. Fair Work: Crowd Work Minimum Wage with One Line of Code. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 7:197–206.

David Wilby, Twin Karmakharm, Ian Roberts, Xingyi Song, and Kalina Bontcheva. 2023. GATE Teamware 2: An open-source tool for collaborative document classification annotation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 145–151, Dubrovnik, Croatia.

Lars Wissler, Mohammed Almashraee, Dagmar Monett, and Adrian Paschke. 2014. The Gold Standard in Corpus Annotation. In *Proceedings of the 5th IEEE Germany Students Conference 2014*, pages 1–4, Passau, Germany.

Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. 2002. Building a large-scale annotated Chinese corpus. In *COLING 2002: The 19th International Conference on Computational Linguistics*, pages 1–8, Taipei, Taiwan.

Mohammad-Ali Yaghoub-Zadeh-Fard, Boualem Benatallah, Moshe Chai Barukh, and Shayan Zamanirad. 2019. A Study of Incorrect Paraphrases in Crowdsourced User Utterances. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 295–306, Minneapolis, Minnesota.

Yinfei Yang, Oshin Agarwal, Chris Tar, Byron C. Wallace, and Ani Nenkova. 2019. Predicting Annotation Difficulty to Improve Task Routing and Model Performance for Biomedical Information Extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1471–1480, Minneapolis, Minnesota, USA.

Seid Muhie Yimam, Chris Biemann, Ljiljana Majnaric, Šefket Šabanović, and Andreas Holzinger. 2016. An adaptive annotation approach for biomedical entity and relation recognition. *Brain Informatics*, 3(3):157–168.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Zhiqiang Zeng, Hua Shi, Yun Wu, and Zhiling Hong. 2015. Survey of Natural Language Processing Techniques in Bioinformatics. *Computational and Mathematical Methods in Medicine*, 2015:1–10.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction Tuning for Large Language Models: A Survey. *arXiv*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less Is More for Alignment. *arXiv*.

Appendix

A Data Handling

In accordance with the Deutsche Forschungsgemeinschaft's (German Research Foundation) "Principles for the Handling of Research Data"[1], we ensured the long-term preservation of research data and/or experimental software that has been developed as part of this dissertation. We made this data openly accessible when possible. The following software has been made available for the scientific community (see the repositories for licensing details):

- Chapter 6: https://github.com/inception-project/inception
- Chapter 7: https://github.com/UKPLab/qanno
- Chapter 8: https://github.com/UKPLab/nessie
- Chapter 9: https://github.com/UKPLab/acl2020-interactive-entity-linking
- Chapter 10: https://github.com/UKPLab/cl2022-annotation-curriculum
- Chapter 11: https://github.com/UKPLab/eacl2023-citizen-science-lessons-learned

Our dataset files are distributed and archived via TUdatalib, TU Darmstadt's research data repository under permissive licenses:

- Chapter 7: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/3939
- Chapter 8: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/3943
- Chapter 9: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2316
- Chapter 10: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2783
- Chapter 11: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/3942

All publications related to this thesis are publicly available on the ACL Anthology or on arXiv:

- Chapter 6: https://aclanthology.org/C18-2002/
- Chapter 7: https://arxiv.org/abs/2307.08153
- Chapter 8: https://aclanthology.org/2023.cl-1.4/

[1] https://www.dfg.de/download/pdf/foerderung/grundlagen_dfg_foerderung/forschungsdaten/leitlinien_forschungsdaten.pdf

- Chapter 9: `https://aclanthology.org/2020.acl-main.624/`
- Chapter 10: `https://aclanthology.org/2022.cl-2.4/`
- Chapter 11: `https://aclanthology.org/2023.eacl-main.261/`

Moreover, all research results of the aforementioned publications are documented in the present thesis, which is archived by the Universitäts- und Landesbibliothek Darmstadt.