# Robustness of Pre-trained Language Models for Natural Language Understanding

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

**Dissertation**

zur Erlangung des akademischen Grades Dr.-Ing.

vorgelegt von
**Prasetya Ajie Utama**
geboren in Bogor, Indonesia

| | |
|---|---|
| Tag der Einreichung: | 28. Juli 2023 |
| Tag der Disputation: | 24. Oktober 2023 |
| Referent: | Prof. Dr. Iryna Gurevych |
| Korreferenten: | Prof. Dr. Roy Schwartz<br>Dr. Nafise Sadat Moosavi |

Darmstadt 2023
D17

**Utama, Prasetya Ajie**

Robustness of Pre-trained Language Models for Natural Language Understanding

Darmstadt, Technische Universität Darmstadt, 2024

Date of the viva voce: 24.10.2023

Jahr der Veröffentlichung der Dissertation auf TUprints: 2024

# Ehrenwörtliche Erklärung [1]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades “Dr.-Ing.” mit dem Titel “Robustness of Pre-trained Language Models for Natural Language Understanding” selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, den 28. Juli 2023

Prasetya Ajie Utama

[1] Gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

# Wissenschaftlicher Werdegang des Verfassers[2]

09/11–05/15 Bachelor of Science (B.Sc.) in Computer Science, University of Indonesia, Jakarta, Indonesia.

09/16–05/18 Master of Science (B.Sc.) in Computer Science, Brown University, Providence, Rhode Island, USA.

09/18–11/21 Doktorand, Ubiquitous Knowledge Processing (UKP-Lab), Techniche Universität Darmstadt, Darmstadt, DE.

[2] Gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt

# Abstract

Recent advances in neural network architectures and large-scale language model pre-training have enabled Natural Language Understanding (NLU) systems to surpass human-level performance on various benchmark datasets. However, a large body of work has revealed that NLU models are brittle against examples from outside of the training data distribution, which consequently limits their real-world application. This brittleness is mainly attributed to models exploiting spurious correlations in the training dataset. That is, models learn to use cues or shortcuts rather than robust features that are representative of the underlying task. In this thesis, we present several methods to alleviate the effect of spurious correlation on the resulting NLU models.

We attempt to improve the robustness against spurious correlation from several directions. Firstly, we address the issues in modeling methods that "debias" NLU models by reducing the incentives to learn non-robust features. We introduce a regularization method that uses the existing knowledge about spurious features' characteristics to improve the out-of-distribution generalization without degrading the original performance on the standard evaluation. We further propose a strategy to maintain the effectiveness of the debiasing methods when the required prior knowledge is not available. Specifically, we introduce a self-debiasing framework that allows the identification of potentially biased examples that models should be disincentivized to exploit. Next, we also look at the inherent robustness that language models acquire during the pre-training on large text corpora. We show how task-specific fine-tuning can be destructive to such robustness and propose a novel regularizing approach to alleviate the degradation. Lastly, we tackle the issue of data augmentation approaches that aim to improve the robust performance of NLU models over downstream application tasks. We present a method to automatically generate diverse and naturalistic examples from which models can reliably learn the task.

In all task settings, we present in this thesis, models are evaluated against out-of-distribution examples designed to penalize the reliance on spurious correlations. We measure the improvement in robustness by showing the increase in performance on these examples without the degradation of the existing standard evaluation. Overall, the work in this thesis demonstrate that we can still obtain robust NLU models using improved modeling and augmentation despite the presence of spurious correlations in the existing training resources.

# Zusammenfassung

Jüngste Fortschritte bei neuronalen Netzwerkarchitekturen und dem Vortraining von Sprachmodellen in großem Maßstab haben es NLU-Systemen (Natural Language Understanding) ermöglicht, bei verschiedenen Benchmark-Datensätzen die menschliche Leistung zu übertreffen. Eine Vielzahl von Arbeiten hat jedoch gezeigt, dass NLU-Modelle gegenüber Beispielen außerhalb der Trainingsdatenverteilung anfällig sind, was folglich ihre praktische Anwendung einschränkt. Diese Sprödigkeit wird hauptsächlich auf Modelle zurückgeführt, die falsche Korrelationen im Trainingsdatensatz ausnutzen. Das heißt, Modelle lernen, Hinweise oder Verknüpfungen anstelle robuster Funktionen zu verwenden, die repräsentativ für die zugrunde liegende Aufgabe sind.

In dieser Dissertation stellen wir mehrere Methoden vor, um den Effekt der falschen Korrelation auf die resultierenden NLU-Modelle zu mildern. Wir versuchen, die Robustheit gegenüber Störkorrelationen aus mehreren Richtungen zu verbessern. Zunächst befassen wir uns mit den Problemen bei Modellierungsmethoden, die NLU-Modelle „verzerren", indem sie die Anreize zum Erlernen nicht robuster Funktionen verringern. Wir führen eine Regularisierungsmethode ein, die das vorhandene Wissen über die Eigenschaften von Störmerkmalen nutzt, um die Verallgemeinerung außerhalb der Verteilung zu verbessern, ohne die ursprüngliche Leistung bei der Standardbewertung zu beeinträchtigen. Wir schlagen außerdem eine Strategie vor, um die Wirksamkeit der Debiasing-Methoden aufrechtzuerhalten, wenn das erforderliche Vorwissen nicht verfügbar ist. Konkret führen wir ein Framework zur Selbstentzerrung ein, das die Identifizierung potenziell voreingenommener Beispiele ermöglicht, für deren Nutzung die Modelle keinen Anreiz haben sollten. Als nächstes betrachten wir auch die inhärente Robustheit, die Sprachmodelle während des Vortrainings an großen Textkorpora erwerben. Wir zeigen, wie eine aufgabenspezifische Feinabstimmung diese Robustheit zerstören kann und schlagen einen neuartigen Regularisierungsansatz vor, um die Verschlechterung zu mildern. Abschließend befassen wir uns mit der Frage der Datenerweiterungsansätze, die darauf abzielen, die robuste Leistung von NLU-Modellen gegenüber nachgelagerten Anwendungsaufgaben zu verbessern. Wir stellen eine Methode vor, um automatisch vielfältige und naturalistische Beispiele zu generieren, aus denen Modelle die Aufgabe zuverlässig lernen können.

In allen Aufgabenstellungen, die wir in dieser Arbeit vorstellen, werden Modelle anhand von Beispielen außerhalb der Verteilung bewertet, die darauf abzielen, die Abhängigkeit von falschen Korrelationen zu bestrafen. Wir messen die Verbesserung der Robustheit, indem wir die Leistungssteigerung an diesen Beispielen zeigen, ohne die Verschlechterung der bestehenden Standardbewertung. Insgesamt zeigt die Arbeit in dieser Arbeit, dass wir mit verbesserter Modellierung und Erweiterung immer noch robuste NLU-Modelle erhalten können, obwohl in den vorhandenen Trainingsressourcen falsche Korrelationen vorhanden sind.

# Acknowledgments

My Ph.D. has been a life-changing experience for me, and I would like to take this opportunity to thank everyone who helped me along the way and make it possible.

I would like to express my heartfelt gratitude to my supervisor Prof. Iryna Gurevych, whose guidance, and encouragement have been instrumental in shaping my research and academic journey. I also extend my deepest gratitude to my co-supervisor Nafise Sadat Moosavi whose unwavering support, encouragement, and understanding have been invaluable to my success. Their dedication to my research and their willingness to devote their time and energy to brainstorming research ideas and providing insightful feedback has pushed me to become a better researcher and scholar.

I am also grateful to my colleagues in the UKP Lab and AIPHES research group, who have provided a stimulating and collaborative environment that has enriched my research experience. In particular, I would like to thank the following people (in random order): Tobias Falke, Markus Zopf, Wei Zhao, Andreas Hanselowski, Leonardo Ribeiro, Fabrizio Ventola, Yevgeniy Puzikov, Shweta Mahajan, Aissatou Diallo, Jonas Pfeiffer, Michael Bugert, Debjit Paul, Ji-Ung Lee. Their camaraderie, intellectual curiosity, and willingness to lend a helping hand have made this journey all the more enjoyable.

During my time as a Ph.D. student, I had the privilege to do a research internship with Bloomberg AI team in London. I later joined the team as a full-time researcher while I was still writing this thesis. I would like to thank my colleagues and friends at Bloomberg AI, Joshua Bambrick, Marco Ponza, Edgar Meij, Diego Ceccarelli, Elliot Gunton, and Thomas Doppiere for all the support and friendships throughout the last few years.

My utmost gratitude goes to the faculties at the Computer Science Department in Brown University where I studied prior my Ph.D. In particular, I would like to thank James Tompkin, Carsten Binnig, and Ugur Cetintemel for their mentorship and believe in me which helped me pave the path of my academic journey.

I would like to also express my gratitude and appreciation to my family and friends without whom I would never be in this position. I thank my wonderful parents, Darna and Fatimah, for their constant support, encouragement, and their hard work to raise their children.

Last but certainly not least, I am deeply indebted to my wife Rumaysha Milhania, whose love, patience, and support have sustained me throughout this challenging and rewarding journey. Your unwavering belief in me and your encouragement have been my constant source of inspiration and motivation. You are the biggest reason why I kept going. I could never thank you enough for your sacrifices and for willing to go through this journey with me from the very beginning when we had nothing to fall back on.

I could not have completed this journey without the guidance, support, and encouragement of these wonderful people. I am truly grateful for their contribution to my academic and personal growth.

# Contents

# Part I

# Synopsis

# Publications

This thesis is written based on four scientific publications that I co-authored with my advisor Iryna Gurevych, my co-advisor Nafise Sadat Moosavi and other external collaborators: Victor Sanh (Huggingface), Joshua Bambrick (Bloomberg). I thank all the co-authors for their contributions and successful collaborations for the work of this thesis. In what follows I discuss the details of my individual contributions to each publication:

**Chapter 5 corresponds to the following publication:**

Utama, P. A., Moosavi, N. S., and Gurevych, I. (2020a). Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics

As the first author of the paper, upon studying the subject of the robustness of NLU models, I identified the drawbacks of the existing debiasing methods and started formulating the research ideas to address the limitations. I developed the new regularization method which I then implemented and extensively experimented with. I regularly discussed the result and the writing of the paper with Nafise Sadat Moosavi and Iryna Gurevych who assisted me in improving the draft and the subsequent versions.

**Chapter 6 corresponds to the following publication:**

Utama, P. A., Moosavi, N. S., and Gurevych, I. (2020b). Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics

As the first and the second author of the paper, Nafise Sadat Moosavi and I first identified the existing limitation of our previously proposed debiasing strategy in handling unknown bias types. My subsequent investigation and systematic analysis resulted in the novel strategy that I propose in this publication. I implemented the newly proposed approach and performed the relevant analyses and experiments. I discussed regularly with my advisor, Iryna Gurevych, who assisted me and suggested additional evaluation to support the main results.

**Chapter 7 corresponds to the following publication:**

Utama, P., Moosavi, N. S., Sanh, V., and Gurevych, I. (2021). Avoiding inference heuristics in few-shot prompt-based finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9063–9074, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics

I conceived the main research contributions while studying the topic of low-resource and efficient NLU modeling with the recent fine-tuning paradigm. I identified the limitation of the existing prompt-based fine-tuning approach and performed the implementation the proposed solution. I then wrote the first draft of the paper and performed the subsequent revisions. I regularly discussed the results with my advisors, Nafise Sadat Moosavi and Iryna Gurevych who assisted me in improving the draft. Additionally, I also regularly had discussions with Victor Sanh who identified several additional experiments that are necessary to support the main findings of the paper.

**Chapter 8 corresponds to the following publication:**

> Utama, P., Bambrick, J., Moosavi, N., and Gurevych, I. (2022). Falsesum: Generating document-level NLI examples for recognizing factual inconsistency in summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2763–2776, Seattle, United States. Association for Computational Linguistics

I formulated the research problem and developed the approaches to address it during my research internship at Bloomberg AI. I had regular discussions with Joshua Bambrick who helped me with various things including setting up the infrastructures and brainstorming the improvement to the proposed methods based on the preliminary results. I also had regular discussions with my advisors, Iryna Gurevych and Nafise Sadat Moosavi, who provided assistance in improving the draft of the publication. I wrote the draft of the paper and together with Joshua Bambrick performed the subsequent revisions to improve it.

Finally, in the interest of completeness, I provided references to other publications for which I collaborated with other researchers during my PhD. While these publications are not included in my thesis, they still fit into the direction of improving the robustness of NLU models.

> Falke, T., Ribeiro, L. F. R., Utama, P. A., Dagan, I., and Gurevych, I. (2019). Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics

In the publication above, Tobias Falke, Leonardo Ribeiro, and I first started the project by brainstorming the idea of applying textual entailment models for summarization. I performed the implementation of entailment models and run several experiments using the data that we collected.

> Stowe, K., Utama, P., and Gurevych, I. (2022). IMPLI: Investigating NLI models' performance on figurative language. In *Proceedings of the*

*60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5375–5388, Dublin, Ireland. Association for Computational Linguistics

In the publication above, Kevin Stowe and I started with the discussion to study the textual entailment models on figurative language phenomena. We brainstormed the plans for collecting the data and running the experiments. I performed the implementation of the model and the necessary experimentation. I assisted the writing of the draft and performed the subsequent corrections.

Moosavi, N. S., de Boer, M., Utama, P. A., and Gurevych, I. (2020). Improving robustness by augmenting training sentences with predicate-argument structures. *arXiv preprint arXiv:2010.12510*

Moosavi, N. S., Utama, P. A., Rücklé, A., and Gurevych, I. (2019). Improving generalization by incorporating coverage in natural language inference. *arXiv preprint arXiv:1909.08940*

In the two preprints above, I assisted with the implementation of the models. Additionally, I performed some experiments to evaluate the proposed methods. I also assisted the writing and performed several corrections to improve the papers.

The source code and reference to the datasets used to reproduce the results in the publications above are available publicly at https://github.com/ukplab.

# Chapter 1

# Introduction

## 1.1 Problem Formulation

Natural Language Understanding (NLU) is a branch of artificial intelligence that aims to develop computer programs that are able to analyze and derive meaningful interpretation from natural language. A large body of work within the area of Natural Language Processing (NLP) has extensively studied NLU under a wide range of tasks including Question Answering (QA) (Sgall, 1982; Rajpurkar et al., 2016; McCann et al., 2018) or Recognizing Textual Entailment (RTE) (Dagan et al., 2006; Bowman et al., 2015; Williams et al., 2018). The progress in NLU has had a significant impact on various technologies such as search engines, automatic text summarization, or virtual personal assistants.

Recently, the emergence of Pre-trained Language Models (PLM) has led to a substantial breakthrough in NLU. The prevailing new paradigm behind PLMs leverages large text corpora to train neural network architectures using unsupervised language modeling objectives. The idea is that the large-scale pre-training extracts useful features in language that are transferable to downstream NLU tasks via supervised fine-tuning. The evaluation using various benchmark datasets shows that the resulting models have pushed the state-of-the-art by a significant margin, and often surpassed the established human performance baselines (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020; Raffel et al., 2020).

This seemingly remarkable performance can be a justification to attribute their success to models' capabilities to better capture the semantics of the text and perform different types of reasoning. However, a growing number of studies have cast doubts on the capabilities that models have truly acquired to perform well on NLU tasks. Researchers have revealed that datasets used to evaluate the models contain spurious correlations that can be exploited to make correct predictions without learning the underlying task. For instance, Gururangan et al. (2018); Poliak et al. (2018b) reported that in many popular datasets for RTE task, also known as Natural Language Inference (NLI), specific linguistic phenomenon such as negation is strongly correlated with certain inference labels, e.g., *contradiction*. While some correlations can be useful to learn, it becomes a problem when the correlations are spurious, i.e., they are idiosyncratic to specific collected datasets rather than representative of the underlying tasks.

Standard evaluation practices that split the collected data into train and test sets randomly often conceal models' reliance on spurious correlation. Since train and test examples are drawn from the same data distribution, utilizing spurious correlation learned from the train set still leads to high-performance score in the test set. Researchers, therefore, proposed a novel evaluation framework by collecting out-of-distribution (OOD) test datasets designed such that predictions based on spurious correlation are incorrect. A prominent example of such evaluation is performed by McCoy et al. (2019) which diagnoses NLI models' reliance on the correlation between lexical overlap with the entailment label. They demonstrate that models are indeed lacking robustness as indicated by their poor performance on the counterexamples where the correlation does not hold.

Mitigating such reliance on non-robust features is critical to transfer the success on the benchmark evaluation to real-world applications. This challenge has thus attracted increasing attention from the NLP community to tackle to the problem from multiple directions. Researchers first looked at the existing NLU datasets to characterize the non-robust features which emerge as artifacts of the data annotation process. Once identified, the information about these features is then used for improving the dataset quality or the modeling methods. On the dataset collection side, such information is useful to filter out examples that exhibit spurious correlation or to obtain counterexamples through iterative annotation (Kaushik et al., 2020) or synthetic data generation (Min et al., 2020). On the modeling side, proposed methods utilize the prior information to augment the training objective, e.g., using regularization terms to suppress the effect of non-robust features on the models.

In this thesis, we aim to develop several possible approaches for mitigating the reliance on spurious correlation and ultimately improve the OOD generalization of NLU models. We study three dimensions of the robustness improvement effort within the recent PLM modeling paradigm: *pre-training*, *fine-tuning*, and *data quality*. We also look at the application of NLU models on more practical downstream tasks and study the benefit from improved robustness. Overall, the work presented in this thesis aims to answer the following research questions:

1. **Fine-tuning**: how to fine-tune models on NLU tasks without suffering from the pitfall of spurious correlation while also maintaining good predictive performance in the training data distribution?

2. **Data**: how to identify unknown spurious correlation in the training data? Specifically, how can we characterize training instances that support the spurious correlation without prior knowledge about their surface feature specifics?

3. **Pre-training**: how is the spurious correlation learning attributed to the pre-training? If PLM acquires more robust features during pre-training, how to train models to preferably use these features for the downstream tasks?

4. **Application**: how to improve the OOD generalization of NLU models to directly improve the performance on the downstream application tasks?

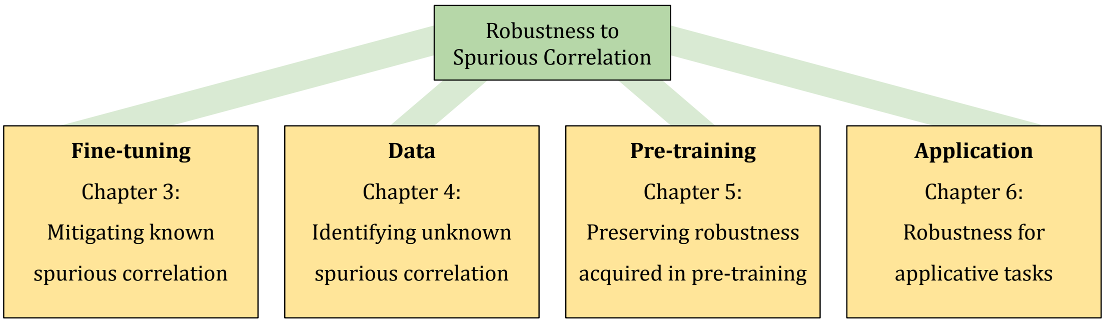

Robustness to Spurious Correlation
Fine-tuning
Chapter 3:
Mitigating known spurious correlation
Data
Chapter 4:
Identifying unknown spurious correlation
Pre-training
Chapter 5:
Preserving robustness acquired in pre-training
Application
Chapter 6:
Robustness for applicative tasks

Figure 1.1: An overview of the contributions and areas of NLU robustness addressed by each chapter in this thesis.

In Section 1.2, we provide a detailed overview of our proposed methods that address the above research questions. In all of our studies, we perform our evaluation around a set of text pair classification tasks, which represent the diverse range of knowledge and reasoning capabilities required in NLU. The improvement in robustness and OOD generalization is measured by performance on challenging test sets where reliance on spurious correlation degrades performance. We compare our methods with various existing improvement methods as well as the vanilla fine-tuning approach. We then show improved performance on different tasks, such as textual entailment, fact-checking, and paraphrase identification, as well as in settings where training resource is low. Overall, this thesis suggests that while spurious correlation exists in the training datasets, efforts from multiple directions can be applied to obtain models that generalize better to out-of-distribution data.

## 1.2 Approach Overview

The structure of this thesis follows the order of publication of the proposed approaches listed in Publications page. We summarize the overall thesis structure in Figure 1.1. In what follows, we briefly discuss the details of each published work from this thesis:

### Improving Robustness against Known Spurious Features

In Chapter 5, we first consider a setting where the characteristics of the non-robust features are known a-priori. This knowledge is typically obtained from task-specific analysis and the intuition of the researchers. For the sentence-pair NLU tasks, we focus on the two most common spurious correlations that affect the robustness of models' performance. Firstly, Gururangan et al. (2018); Poliak et al. (2018b); Tsuchiya (2018) have shown that popular NLI datasets such as SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) contain hypothesis sentences with indicative keywords that allow models to make correct predictions without properly using both of the input sentences. For instance, negation words such as “no” or “never” are strongly correlated with the "contradiction" label, while verb like "sleeping" is correlated with the "neutral" label. Beside this "hypothesis-only" features, McCoy et al. (2019) also found that models also rely on simple heuristics based on the lexical overlap features between the sentence pair. More specifically, models learn a fallible assumption that sentence pairs that highly overlap in words are classified as "entailment". Due to the bias in both the training and test sets, this heuristic works well on the majority of the test cases, but fails on simple counterexamples, e.g., "*cat chased the mouse*" does not entail "*the mouse chased the cat*".

This insight has motivated researchers to develop methods for disincentivizing models to capture spurious patterns. These methods are commonly referred to as "debiasing". Ensemble-based debiasing strategies (Clark et al., 2019; He et al., 2019; Karimi Mahabadi et al., 2020) have gained prominence and improved models' performance on the OOD diagnostic datasets substantially. They work by first training a simple hand-crafted model that relies solely on spurious features, e.g., predicting relationship labels only based on partial input. The main model is then trained in an ensemble with the biased model, which reduces the incentive for the model to learn from the examples that the biased model has predicted correctly. The model is, in turn, encouraged to focus on "harder" examples where spurious features are not sufficient to make correct predictions. While OOD evaluation shows the effectiveness of these approaches for the intended aim, it also results in a trade-off with respect to the *in-distribution* performance. Namely, the performance on the standard test set, which contains a wider range of inference phenomena, is substantially degraded.

In this work, we address this limitation by introducing a regularization method which penalizes the overconfidence of models in "easier" examples where spurious features give away the correct labels. We show that this leads to models being less likely to pick up simple cues, while also still being able to learn to make the correct prediction on these examples. The evaluation shows that models achieve OOD improvement while preserving the in-distribution performance.

### Addressing Unidentified Spurious Features

Next, in Chapter 6, we consider a scenario where the prior knowledge about the characteristics of the spurious correlation is unknown, which then limits the applicability of the existing debiasing approaches. For these approaches to work, "easy" examples that contribute to the spurious correlation should be identified automatically without hand-crafting a simple model. To this end, we look at the training dynamics of the recent pre-trained language models during fine-tuning and found that models initially capture simple patterns before gradually learning more complex inference rules. Interestingly, we observe that the simple patterns in this low data regime correspond to the non-robust features that harm the resulting OOD generalization. The early training models behave similarly to the manual hand-crafted models used to identify examples that support spurious correlation.

Based on this newly found insight, we propose a novel framework within which two identical pre-trained models are fine-tuned with significantly different amounts of training examples. The first model trained on a small fraction of fine-tuning data is expected to be a shallow model that heavily utilizes non-robust features. The shallow model is then used to reweight the training examples via various mechanisms that are introduced by previous debiasing methods. The main model, which is trained on the reweighted training examples, is expected to learn non-overlapping inference strategies from the shallow model, resulting in improved robustness against spurious correlation. During test time, the shallow model is no longer needed and only the main model is then used. Our evaluation shows that this self-identifying framework gains equally high improvement as its counterpart that utilizes the prior knowledge about the spurious features.

### Robustness in Low Resource Learning Settings

Our finding on models' reliance on spurious patterns during the earlier training phase elicits a new question on the robustness of pre-trained language models. More specifically, we look at whether language models obtained from large-scale pre-training are still inherently reliant on surface features that do not generalize well. This can severely limit the application of models in various low-resource settings, where models only have access to a few examples to avoid learning spurious patterns. To this end, in Chapter 7, we study a more recent prompting paradigm to adapt pre-trained language models to perform the downstream tasks with small to no training examples (commonly referred to as few- and zero-shot learning), which allows us to investigate models' robustness on low data regimes.

The prompting paradigm reformulates downstream task examples as *masked* language modeling instances using textual prompts. Models, which are pre-trained using similar formulations on a large corpus of text, can then make a "fill in the blank" textual prediction that can be mapped directly to task-specific labels. For instance, a textual prompt "*It was* _____." can be added to a sentiment classification input sentence "*The food was delicious*". Probabilities assigned by the pre-trained language models on the word "*good*" and "*bad*" can then be compared to determine whether the sentiment is *positive* or *negative*.

In this work, we perform systematic robustness evaluation of zero-shot and few-shot prompted models for NLU tasks using varying training example sizes. We found that zero-shot models are more robust to common spurious correlation, as measured by their performance on the corresponding diagnostic datasets. Interestingly, prompt-based fine-tuning gradually degrades this robustness as more labeled examples are used to train the pre-trained language models. We then address this by proposing a regularization term that penalizes the fine-tuning from updating the weights too far off the original pre-trained values. Our evaluation suggests that the proposed regularization helps models gain in-distribution performance improvement while maintaining their pre-trained robustness. Overall, this work highlights the importance of preserving useful knowledge extracted during pre-training to obtain robust models in low-resource learning settings.

### Data Augmentation for Robust Downstream Applications

Finally, in Chapter 8, we look at the implication of model robustness on the downstream application tasks. More specifically, we focus on the task of detecting the factually inconsistent output of summarization models. Out-of-the-box Natural Language Inference models are adopted for this task by formulating the input document as the premise text and the summarization output as the hypothesis, where entailed pairs are equivalent to factually correct summaries (Falke et al., 2019). However, this application of NLI models has seen limited success with poor performance scores that are close to random (Kryscinski et al., 2020).

This poor performance largely stems from the mismatch between the NLI training data and the downstream task test data. Namely, most NLI models are trained on single-sentence premise text, while at test time, the model is used to make predictions on the document-length input text. Models may learn spurious patterns that are idiosyncratic to single-sentence NLI datasets, which do not generalize to document-level test cases in the downstream tasks. Additionally, the resulting NLI models also may not capture the kind of entailment phenomena which naturally arise in longer text input in summarization.

We argue that addressing this challenge requires effort to reduce the characteristics discrepancy between the training and test data. More specifically, the NLI training dataset should include document-level examples which are more oriented to the downstream task. However, collecting such labeled training examples manually can be costly and time-consuming. In this work, we address this challenge by proposing a novel data generation framework that produces diverse and naturalistic NLI examples on document-level granularity. We leverage the advances in text-generating language models to build a text generator that takes as an input a pair of the source document and its corresponding summary. It then generates a perturbed summary output that is no longer entailed by the source document, which constitutes the negative examples of the resulting dataset. We use the automatically constructed document-level dataset to augment the existing sentence-level NLI dataset. The improved performance in our evaluation on several factual inconsistencies benchmark shows that the resulting NLI models are more robust when applied to the intended downstream tasks.

## 1.3 Terminology

We briefly clarify the terminology that we use throughout this thesis document:

- In the literature, spurious correlation is also often referred to as dataset bias (Clark et al., 2019; He et al., 2019), annotation artifacts (Gururangan et al., 2018; Poliak et al., 2018b), or shallow features. The choice of terminology in this document is meant to be informal and we use these terms interchangeably.

- The resulting behavior from relying on spurious correlation is commonly referred to as inference heuristics (McCoy et al., 2019). Several works also refer to it as shortcut learning (Du et al., 2021; Lai et al., 2021).

- We refer to the standard evaluation set as in-distribution data since it is drawn from the same distribution as the training data. The diagnostic datasets used to evaluate the improvement in robustness will be referred to as *out-of-distribution* (OOD) data or *challenge sets.*

## 1.4 Organization

This thesis is organized as follows:

- In Chapter 2, we start with the discussion about the existing analysis and evaluation methods of the NLU models. We discuss the definition of the tasks used for evaluating the models and how their performance is measured. We finally discuss the potential downstream applications of the NLU models and highlight the importance of robustness improvement towards this end.
- In Chapter 3, we then discuss the existing approaches for NLU tasks that are based on neural network and deep learning. The discussion includes the recent advances in Transformers architectures and pre-trained language models that are the keys to the current state-of-the-art performance across NLU benchmarks.
- In Chapter 4, we summarize the body of work that studies the existing limitation of NLU datasets and models with respect to robustness against spurious correlations. We first discuss the methods to investigate the artifacts in the existing NLU datasets and the diagnoses to demonstrate the bias-reliant behaviors in the NLU models. We then highlight the existing robustness-improving methods that address the problem from the dataset and modeling point of view.
- In Part 2 of the thesis, we include the publications that comprise the main contributions of this thesis. We start with Chapter 5 where we discuss our proposed method to improve the robustness against the known biases in the dataset. Chapter 6 addresses the debiasing scenario where the prior knowledge about the biases is minimal. In Chapter 7, we study the low resource settings where we show that the robustness against spurious correlation can be improved by maintaining the knowledge acquired during pre-training. Lastly, in Chapter 8, we discuss our proposed framework to automatically augment the existing NLU dataset to improve the robustness of the models which is specific toward downstream applications.
- Finally, we conclude in Chapter 9 where we discuss the summary of the contribution of this thesis and highlight the possible research directions.

# Chapter 2

# Evaluating Natural Language Understanding

## 2.1 Task Definition

In this thesis, we address the robustness problem in tasks that are designed to evaluate and analyze the capabilities of general NLU systems. The emerging standard practice in this direction is based on evaluation protocols using a suite of supervised tasks that are uniformly formatted, such as question answering (McCann et al., 2018) and classification (Conneau and Kiela, 2018). GLUE benchmark (Wang et al., 2018) serves as a prominent example of this approach: it provides a collection of diverse text classification tasks of varying domains, dataset sizes, and difficulties. The benchmark measures the overall performance of NLU models across tasks as an estimate of models' general linguistic knowledge and task-specific capabilities. The majority of these NLU tasks are formulated as input pair classification, which determines the output label based on the semantics of each individual input text and the relationship to the other. In our studies, we focus specifically on these paired input tasks, which we argue allow the evaluation of wider linguistic phenomena occurring from cross-text contextualization and alignment.

Formally, the input pair classification tasks are formulated as the following: given an input text pair $(x_1, x_2)$, the objective is to determine a label $y$ that represents the relation between meanings of $x_1$ and $x_2$. Existing tasks address varying types of semantic relationships that are encapsulated by a different set of discrete labels. Consider the following example of a sentence pair input:

1. $x_1$: The skyscraper is under construction
2. $x_2$: The building is under construction

In textual entailment tasks such as NLI, the relation between $x_1$ and $x_2$ is assigned to the positive **entailment** label since $x_1$ entails $x_2$. Whereas Paraphrase Identification task (Xu et al., 2014; Vo et al., 2015) considers a more precise relation to determine whether the two sentences are semantically equivalent. Specifically, the text pair $x_1$ and $x_2$ are **paraphrase** when they entail each other (symmetrical entailment). This example illustrates the need for NLU models to acquire task-specific capabilities besides their general knowledge. To gauge the extent of these capabilities, NLU models have been evaluated on a wide range of paired input tasks encompassing varying text domains and characteristics. The followings are the exemplary paired NLU tasks and their descriptions:

### 2.1.1 Recognizing Textual Entailment

Recognizing Textual Entailment (e.g., Dagan et al., 2006; Bentivogli et al., 2010), known also as Natural Language Inference or NLI (e.g., Bowman et al., 2015; Williams et al., 2018) is a broader task that is concerned with the directional entailment relationship between two fragments of text (e.g., paragraphs or sentences), namely the entailing "*premise*" text $P$ and the entailed "*hypothesis*" text $H$. Dagan et al. (2006) introduced the somewhat imprecise definition of entailment described below:

> Text expression $P$ entails text expression $H$ if, a typical human reader can infer the meaning of $H$ from the meaning of $P$, based on the common understanding of language and world knowledge.

The datasets for the task are therefore constructed by collecting human annotators' judgments on the entailment relationship between the given text pairs. In the annotation process, annotators are typically asked to obtain the premise ($P$) and hypothesis texts ($H$) in varying ways including by performing information retrieval from the web, asking a Question Answering (QA) systems, translating sentences with Machine Translation (MT) systems (Giampiccolo et al., 2007). Alternatively, annotators can also be prompted to generate free-form hypothesis texts with intended entailment labels (Bowman et al., 2015; Williams et al., 2018). The resulting P-H pairs from the earlier RTE datasets are then formulated as binary classification problems where P-H texts are the inputs which are labeled as either *entailment* or *not-entailment*. Some representative examples from the earlier RTE datasets are the following:

1. ▷ $P$**:** Oil prices fall back as Yukos oil threat lifted.
   ▷ $H$**:** Oil prices rise. *(not-entailment)*

2. ▷ $P$**:** Money raised from the sale will go into a trust for Hepburn's family.
   ▷ $H$**:** Proceeds go to Hepburn's family. *(entailment)*

More recently, successor datasets such as SICK (Marelli et al., 2014), SNLI (Bowman et al., 2015), and MNLI (Williams et al., 2018) break down the *not-entailment* label into two separate labels *contradiction* and *neutral* and introduced a 3-way classification formulation for the RTE/NLI task. de Marneffe et al. (2008) define the *contradiction* relationship as the following:

> The text expression $H$ contradicts the text expression $P$ if, a typical human reader infers that the statements described in $P$ and $H$ are unlikely to be true at the same time.

Similar to the definition of the entailment relationship, the judgment for the contradiction label is also based on human intuition and common sense. For cases where the truth value of $H$ is unknown given that $P$ is true, the *neutral* label is used. Consider the following example from the SNLI dataset (Bowman et al., 2015):

- ▷ $P$: A man in a black shirt overlooking bike maintenance.
- ▷ $H_c$: A man destroys a bike.
- ▷ $H_n$: A man learns bike maintenance.

In the above, annotators generated the contradicting sentence $H_c$ as they inferred that $H_c$ is extremely unlikely to be true given the event described in $P$. On the other hand, sentence $H_n$ is generated for the neutral label because it is still possible that $H_n$ is true although it may not be certain.

### 2.1.2 Semantic Textual Similarity

Semantic Textual Similarity (STS) (e.g., Agirre et al., 2012, 2014; Cer et al., 2017) is a task that assigns a degree of semantic relatedness between a pair of text. The task differs from textual entailment in several ways. First, the target semantic relationship between the text is symmetric, meaning that labels assigned to the text pairs $(T_1, T_2)$ and $(T_2, T_1)$ are the same. Unlike STS, the textual entailment relationship is directional, e.g., "playing sport" does not entail "playing football" even though they both have the same semantic relatedness to each other. Second, the task introduced a notion of graded semantic similarity where a label is closer to one label than the other. Consider the following sentences:

- ▷ $T_1$: The person is driving a minivan.
- ▷ $T_2$: The person is sitting in a sedan.
- ▷ $T_3$: The person is sitting in a living room.

In the textual entailment task, the meaning of both sentences $T_2$ and $T_3$ are contradicting sentence $T_1$. However, the notion of graded semantic similarity in the STS task takes into account that "minivan" is semantically more related to "sedan" than "living room". Therefore, the label assigned to $(T_1, T_2)$ is closer to the higher end of the scale than the label for $(T_1, T_3)$. In the corresponding benchmark dataset for the task, STS-B (Cer et al., 2017), which includes a selection of English STS shared tasks from 2012 to 2017, the labels are defined on a Likert scale ranging from 0 to 5. Label 0 indicates no meaning overlap, while label 5 indicates that the meanings of the two sentences are equivalent. The intermediate values from 1 to 4 express partial overlap in meaning where the sentences share different degrees of details and information.

Similar to the RTE and NLI datasets, the annotations for STS tasks are also designed to reflect pragmatic and the common understanding of the language and the world. The resulting tasks are commonly modeled by formulating them as regression problems where models take the sentence pairs as input and map them to scalar values between 0 and 5.

### 2.1.3 Paraphrase Identification

Paraphrase identification task (e.g., Dolan et al., 2004; Dolan and Brockett, 2005; Xu et al., 2014) aims to identify text pairs that are semantically equivalent. The task is formulated as a simple binary classification, where the inputs are the sentence pairs and the labels are either "*paraphrase*" or "*not-paraphrase*". Formally, a pair of sentences is considered as "*paraphrase*" when the two sentences hold a "bidirectional entailment" relationship, i.e., the sentences entail each other. However, Dolan and Brockett (2005) suggest that, in practice, such a strict definition would mostly lead to pairs that are identical at the string level. They illustrate using the following example

- ▷ *Sentence 1:* The euro rose above US$1.18, the highest since its January 1999 launch.
- ▷ *Sentence 2:* The euro rose above $1.18 the highest level since its launch in January 1999.

that the resulting pairs tend to present discrepancies mostly by synonymy and local syntactic changes. To obtain a richer dataset with more types of complex paraphrases, they, therefore, adopt a looser definition of paraphrase, where the two sentences may differ in some details to a certain degree. Human annotators are asked to judge whether a sampled pair of sentences, based on common sense, is similar enough in meaning to be considered a paraphrase.

The relaxed paraphrase definition used in this task is akin to using the conflated labels in the Semantic Textual Similarity (STS) task (Cer et al., 2017). Specifically, the paraphrase criterion conflates the definition of labels **3**, **4**, and **5** of the STS tasks where the two sentences are roughly equivalent but some information or details may still differ. Interesting and complex paraphrases with minor discrepancies in details can be illustrated by the following example:

- ▷ *Sentence 1:* They were at Raffles Hospital over the weekend for further evaluation.
- ▷ *Sentence 2:* They underwent more tests over the weekend, and are now warded at Raffles Hospital.

To make sure the datasets for this task consist of naturally occurring and non-handcrafted sentence pairs, researchers extracted the pairs from various sources using several heuristics to narrow down the search space. Dolan and Brockett (2005) collected news story pairs from the web and use heuristics based on lexical properties and the sentence position in the document. More recently, online community question-answering websites such as Quora[1] have become major sources of various NLU datasets (Nakov et al., 2016, 2017; Abujabal et al., 2019) including for paraphrase detection. Quora Question Pairs (QQP)[2] dataset consists of question sentence pairs with expert annotations on whether the questions are duplicates. The syntactic variation in lexically similar questions can pose a challenge for NLU models to determine the difference in meaning, e.g., "Is there a direct flight from New York to London?" vs. "Is there a direct flight from London to New York?".

### 2.1.4 Fact Verification

Fact Verification (e.g., Vlachos and Riedel, 2014; Ferreira and Vlachos, 2016; Thorne et al., 2018) presents a practical task of automatically recognizing the factual correctness of a textual claim given source or evidence texts. This task adopted definitions of factual relationship labels which are similar to the labels in the textual entailment (RTE/NLI) tasks, in that a claim is factual only if all details and information in the claim text are *entailed* by the given evidence text. Datasets for the claim verification task such as (Thorne et al., 2018) formulated the task as a 3-way classification problem where the evidence and claim text pairs are labeled as either "*support*", "*refute*", or "*not-enough-info*". These labels correspond to the RTE/NLI labels "*entailment*", "*contradiction*", and "*neutral*", respectively.

The key difference between the fact verification task and the textual entailment task is that in RTE/NLI datasets, typically both the input texts (premise and hypothesis) consist of a single sentence. In fact verification task, however, the claim is verified against longer passages that are obtained from additional retrieval steps from a large collection of documents. For NLU task formulation, this retrieval step can be omitted, and the gold evidence passages are already provided. The resulting multi-sentence source or context texts present challenging inference phenomena for NLU models. Consider the following example from the Fever dataset:

- ▷ **Evidence 1:** The 1992 Los Angeles riots, also known as the Rodney King riots were a series of riots, lootings, arsons, and civil disturbances that occurred in Los Angeles County, California in April and May 1992.
- ▷ **Evidence 2:** Los Angeles County, officially the County of Los Angeles, is the most populous county in the USA.
- ▷ **Claim:** The Rodney King riots took place in the most populous county in the USA.

The above evidence passages that are relevant to the claim text are retrieved from multiple Wikipedia pages. The relevant information from each passage then needs to be consolidated for the claim text to be properly verified. The fact verification in such an example requires multi-hop (multi-step) inference (Welbl et al., 2018): inferring that *The Rodney King riots* happened in *Los Angeles*, based on evidence 1,

[1] https://www.quora.com/

[2] https://www.kaggle.com/competitions/quora-question-pairs/data

and figuring out that *Los Angeles* is *the most populous county in the USA*, based on evidence 2. Evaluating NLU models on this type of example can provide an estimate of their ability to handle more inference phenomena that are present in tasks with longer input texts.

## 2.2 Performance Measures

**Metrics** The performance of the models applied for the NLU tasks is measured by varying metrics depending on the characteristics of the datasets. For classification tasks like RTE/NLI where the class distribution is balanced, the accuracy metric is typically used. It is calculated by the following:

$$\texttt{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP (true positive) and TN (true negative) represent the number of correct predictions and the summation of TP, TN, FP (false positive), and FN (false negative) represent the total number of predictions. In tasks where the class distribution is imbalanced, the accuracy metric is not sufficient to estimate the performance. Two better metrics commonly used for class-imbalanced tasks are precision and recall. Precision is interpreted as the proportion of positive predictions that are actually correct. It is defined as follows:

$$\texttt{Precision} = \frac{TP}{TP + FP}$$

On the other hand, recall measures the proportion of actual positive cases that were predicted correctly by the model. The definition is as follows:

$$\texttt{Precision} = \frac{TP}{TP + FN}$$

The performance of the models in the class-imbalanced cases should be evaluated in terms of both precision and recall. The improvement of either of these two metrics, however, is often at odds with the other. It is often reasonable to optimize models such that both precision and recall are maximized. F1 score is used to summarize the metrics into a single score. It is mathematically defined as follows:

$$\texttt{F}_1 = 2 \cdot \frac{\texttt{precision} \cdot \texttt{recall}}{\texttt{precision} + \texttt{recall}}$$

This harmonic mean formulation between precision and recall ensures that the decrease of either metric will penalize the F1 score.

**Baselines** There are several baselines that are typically used to measure the progress in modeling an NLU task. The *random baseline*, which randomly assigns a target label to a test case, can be used as a performance lower bound for the NLU models. Alternatively, *most frequent class* baseline, which always predicts the most frequent label in the training set, is useful to interpret models' performance metrics. For example, on a binary classification task, where 70% of the training examples are labeled as the negatives, the *most frequent class* baseline would achieve around 70% of accuracy. Based on this, we can infer that the model that achieves 72% accuracy is not making substantial progress for the task. Lastly, the human performance baseline is often used in many NLU tasks to gauge the upper-bound performance of the NLU models (Wang et al., 2018). Establishing a human baseline performance on already annotated NLU datasets requires the further collection of human judgment on the existing test cases. For instance, Nangia and Bowman (2019) presents an estimation of human performance on GLUE benchmark tasks by asking crowdworkers to perform the task after training them using a few examples from each task. This human baseline was able to outperform several state-of-the-art NLU models on this benchmark. However, the human baseline performance presents a tight upper bound, meaning there was only a limited headroom remaining for further progress in this set of tasks.

## 2.3 Task Application

A myriad of inference phenomena, which occur in multiple downstream applications, can be cast in terms of paired text task formulation, such as textual entailment (Poliak et al., 2018a). This means that the paired input tasks present a generic formulation for common evaluation and comparison between various applied NLU models. In the following, we describe how several NLU applications can be expressed in terms of paired text task format.

**Summarization** The summarization task, in which systems are developed to generate human-readable summaries from longer text input. While existing summarization systems can produce highly fluent and coherent summaries, a large body of work shows that the generated summaries are often factually inconsistent with respect to the source document (Kryscinski et al., 2019). This gives rise to a subtask of recognizing factual consistency which aims to identify whether the newly generated sentence contains information already expressed in the source document. This sub-task can be formulated as an entailment pair where the source document is the premise text and the generated summary is the hypothesis text. Consider the following document ($P$) and generated summary ($H$) pair:

- ▷ $P$: The classic video game "Space Invaders" was developed in Japan back in the late 1970's – and now their real-life counterparts are the topic of an earnest political discussion in Japan's corridors of power.
- ▷ $H$: Video game "Space Invaders" was developed in Japan back in 1970.

In the above, the summarization system incorrectly consolidates the phrase "*back in the late 1970's*" into "*back in 1970*". Using RTE/NLI or Fact Verification task formulation, the pair should be labeled as *contradiction* or *refute*. The generated summary is considered factually consistent only when the source document and the summary pair hold the *entailment* or *refute* labels. Despite this aligned task formulation, applying out-of-the-box RTE/NLI or Fact Verification models directly to the task still result in unsatisfactory performance (Kryscinski et al., 2020; Falke et al., 2019). Several follow-up works, e.g., the work by Laban et al. (2022) and our work discussed in this thesis, address this limitation of NLU models to improve their downstream application performance.

**Relation Extraction** The task of relation extraction (RE) aims to recognize a fixed set of real-world relationships between pairs of entities from natural language texts. The outputs of a relation extraction system are (`entity1`, `relation`, `entity2`) tuples that are defined in the schema of a knowledge base. These fixed sets of relations and entities can be expressed in varying surface forms in the text. For instance, the relation `born_in` can appear in the text in the following sentence:

> "Jo went to visit Alex in his native York."

where the target relation tuple is (`Alex`, `born_in`, `York`). This extraction can be recast into an entailment format by formulating the above text as the premise sentence and the candidate tuple as the following hypothesis sentence:

> "Alex was born in York."

The tuple is determined to occur in the text if the applied RTE/NLI system predicts that the source text entails the generated hypothesis text.

**Question Answering** Reading comprehension question answering task (RCQA) aims to answer a question `Q` to a passage `P` by generating either a free-form text (Lai et al., 2017; Tapaswi et al., 2015) or by extracting a text span (substring) from `P` (Rajpurkar et al., 2016; Trischler et al., 2017). The task can be formulated to paired text input format by transforming the question `Q` and the answer output `A` into a declarative answer statement `D`. Consider the following example from the SQuAD dataset:

- ▷ `P`: "Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion ***Carolina Panthers*** 24–10 to earn their third Super Bowl title. (. . . )"
- ▷ `Q`: "Which NFL team represented the NFC at Super Bowl 50?"
- ▷ `A`: "Carolina Panthers"

By using various approaches such as rule-based syntactic transformation or neural sequence modeling Demszky et al. (2018); Mishra et al. (2021), the text `Q` and `A` can be converted to a declarative sentence `D`:

> "Carolina Panthers *is the* NFL team *that* represented the NFC at Super Bowl 50"

Once the (P, D) pair is obtained, a textual entailment can be directly applied. One promising application of the RTE/NLI model for RCQA tasks is to verify the output of QA systems Chen et al. (2021). More specifically, an answer A is determined to be correct when the converted declarative sentence D (hypothesis) is entailed by the given passage P (premise) which contain all the necessary information. The probability assigned by the NLI model to the entailment label can then be used to re-rank the candidate answers or to improve the calibration for "unanswerable" (Rajpurkar et al., 2018) or "selective" (Kamath et al., 2020) QA settings.

## 2.4 Thesis Contribution

In the above, we elaborate on the wide range of NLU tasks that are formulated as paired input text classification. While the work in this thesis focuses on these tasks, the utility of this task formulation means that the contribution of this thesis extends to wider NLU problems. In Utama et al. (2020a,b, 2021), we evaluate the proposed methods on tasks including Natural Language Inference (NLI), Paraphrase Identification, and Fact Verification. Our evaluations use a variety of large-scale datasets such as MNLI (Williams et al., 2018), SNLI (Bowman et al., 2015), and Fever (Thorne et al., 2018) which cover a broad range of domains and language phenomena. As previously discussed in the above section, the significance of our contribution is also highlighted by the utility of the resulting models for these paired input tasks for downstream applications. In Utama et al. (2022), we study the application of NLI models for the task of verifying factuality in summarization. We show that the robustness improvement by our proposed method allows for a more reliable application of out-of-the-box models in downstream tasks.

# Chapter 3

# Modeling Natural Language Understanding

## 3.1 Methods

The current dominant approaches for NLU tasks are based on deep learning algorithms, which aim to encode input text into meaningful vector representations and map them to the target task labels. In this section, we briefly discuss the deep learning models' main building blocks, which include the artificial neural network, sequence encoder, and transfer learning from the pre-trained language models based on the transformers architecture.

### 3.1.1 Neural Networks

Neural networks are computational models used in machine learning that work by learning the *representations* of the input that are expressed in terms of other simpler *representations*. This learning approach is argued to enable the model to build complex concepts out of simpler concepts (Goodfellow et al., 2016). For instance, the meaning representation of a sentence can be expressed as a composition of the words and phrases representations.

**Feedforward neural network** which is also known as Multilayer Perceptron (MLP), is a quintessential example of deep neural network models that are still widely used across domains as part of more complex neural models. MLP is in essence a function that is formed by composing a set of simpler functions. It maps the input values to the output values by sequentially applying the simpler functions to the output of the previous ones. Figure 3.1 illustrates a simple 2-layer feedforward neural network for a 2-way classification task. Given an input vector $x \in \mathbb{R}^n$, the neural network $\mathbf{F}$ consists of learned weights matrices $\mathbf{W}_0 \in \mathbb{R}^{n \times m}$ and $\mathbf{W}_1 \in \mathbb{R}^{m \times 2}$. $\mathbf{F}$ is a function defined as follows:

$$\mathbf{F}(x) = \sigma_1(\sigma_0(x\mathbf{W}_0)\mathbf{W}_1)$$

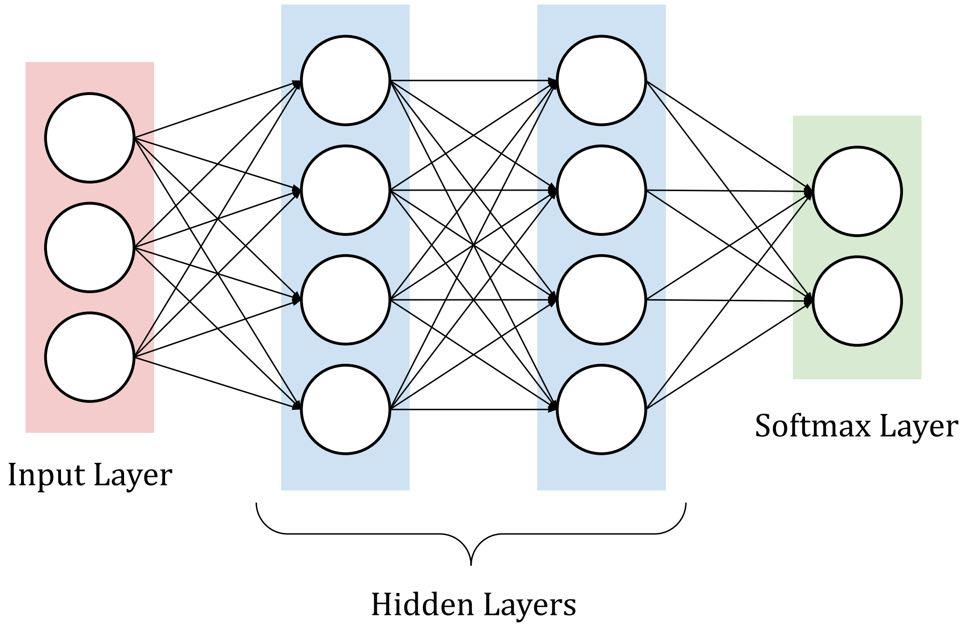


Figure 3.1: An illustration of a simple 2-layer feedforward network for a 2-way classification task. The output layer uses a softmax function which normalizes the output values into probabilities that sum to 1.

where $\sigma_0$ is a non-linear activation function, e.g., $ReLU(x) = max(0, x)$, that follows the matrix multiplication on each intermediate layer. On the final layer, a softmax function is commonly used to produce normalized probability values which sum to 1 and are assigned to each target class. The output of a simple softmax function $\sigma_1$ on each class $i$ can be defined as $\sigma_1(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{2} e^{x_j}}$.

**Training** or learning of the neural network is performed by finding the weights' values that minimize a certain objective or loss function $\mathbf{J}(\mathbf{W})$ via some variant of gradient descent optimization. Typically, the loss function is defined as a performance measure that is evaluated over the training data (as well as the additional regularization terms), e.g., log-likelihood loss for classification. The gradient of the loss with respect to each weight can be efficiently calculated using the Back-propagation algorithm (Rumelhart et al., 1986). The optimization algorithm then uses the gradients computed by Back-propagation to update the weights with certain learning rates.

### 3.1.2 Sequence Encoder

The basic feedforward neural network is rarely used by itself as it is insufficient to represent the sequential structure of text input in NLU tasks. In the following, we introduce the most common neural model architecture classes designed to capture and represent sequential data: recurrent networks and transformers.

**Recurrent neural networks** process a sequence of values $x = x^{(1)}, x^{(2)}, \dots, x^{(n)}$, e.g., word vector representations in a sentence, by using network connections that form a directed cycle. More specifically, a recurrent neural network (RNN) shares the same weights $\mathbf{W}$ across time step $t$ to compute the hidden state representation $h^{(t)}$, given the current input $x^{(t)}$ and the previous hidden state $h^{(t-1)}$. Namely, the current hidden state $h^{(t)}$ is computed as the following:

$$h^{(t)} = \mathbb{F}(h^{(t-1)}; x^{(t)}; \mathbf{W})$$

This chain-like structure allows RNN to incorporate the information from the previous values in the input sequence into the representation of the current time step. The ability to "memorize" past information is crucial for various language tasks such as language modeling, where the context from the previous words is required to correctly predict the next word (Bengio et al., 2003). Consider the following text:

*I grew up in Germany. I speak fluent* _____.

By looking at the longer context, which includes information about the country, the model not only can predict that the missing word is a name of a country but also narrow down which language it is most likely to be. While in theory, vanilla RNNs should be capable of handling such "long-term dependencies", in practice, RNNs often fail to learn them. The two main reasons are the vanishing gradient and exploding gradient problems which are described in Bengio et al. (1994) and explored further in Pascanu et al. (2013). Researchers, therefore, proposed variants of RNN based on Long Short Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997) to address the known training issues. Models derived from LSTM are successfully applied to a large variety of problems including sequence-to-sequence modeling tasks such as machine translation (Sutskever et al., 2014). LSTM networks are also adopted for paired input NLU tasks including NLI using various approaches. For instance, in Conneau et al. (2017), a sentence representation of each input sentence is obtained by either taking the last hidden state $h^{(t)}$ or by taking mean/max pooling over all of the hidden states. Once the vector representations for each sentence $v_1$ and $v_2$ are obtained, the inference is performed by a 3-class MLP classifier $\mathbb{G}$. Namely, $\mathbb{G}$ takes as input the concatenation of these two vectors and their element-wise comparisons, e.g., $\mathbb{G}(v_1, v_2, v_1 * v_2, |v_1 - v_2|)$.

**Transformers** architecture (Vaswani et al., 2017) is designed to address the fundamental constraint of sequential computation in models based on RNN. Specifically, in RNN, computing hidden states for each time step $t$ depends on the previous hidden state $h^{(t-1)}$, which makes it unsuitable to parallelize the computation over separate time steps. This introduces a scalability issue when processing longer sequence lengths. Transformer networks dispense with this limiting recurrence structure by solely using attention mechanisms (Bahdanau et al., 2015).

While Transformer architecture is originally introduced as an encoder-decoder model for sequence-to-sequence tasks, the following description focuses only on the encoder side, which is more relevant to its application to the paired input NLU

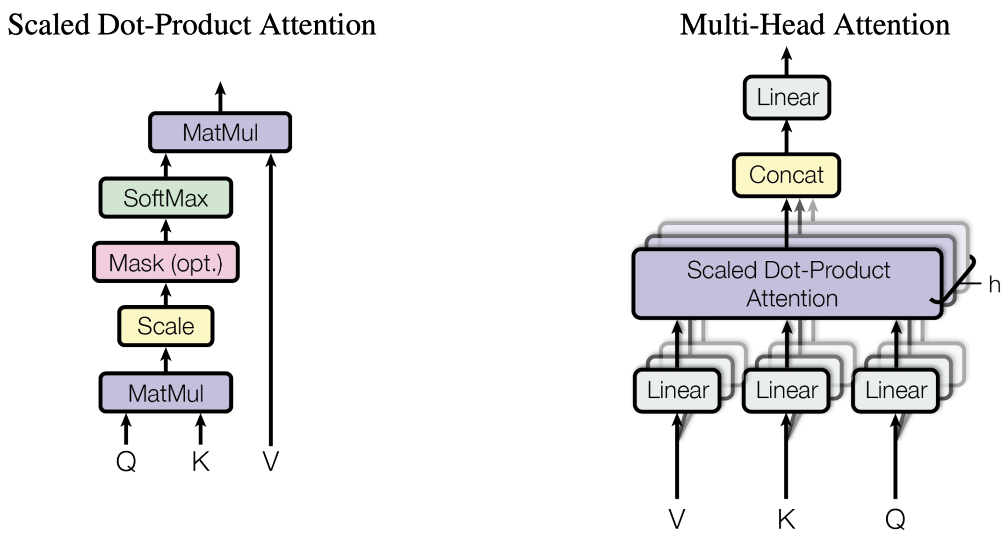


Figure 3.2: The illustration of each multi-head self-attention layer in the Transformer architecture (right). Each layer computes the scaled dot product attention (left). The figures are taken from Vaswani et al. (2017).

tasks (Devlin et al., 2019). The encoder consists of a stack of $N$ transformer layers, each encodes the input sequence of symbol representations $(x_1, \ldots, x_n)$ and maps them to a sequence of intermediate representations $h = (h_1, \ldots, h_n)$. We show the illustration of each layer in Figure 3.2. Each layer $l$ has two sub-layers: the multi-head self-attention mechanism and a simple position-wise feedforward neural network (FFN). The multi-head structure of the self-attention layer allows the model to attend to different representation subspaces of the symbol in the input sequence. Each attention head first takes the input vector representation $x_1 \in \mathbb{R}^d$ at each position and calculates the scaled dot product attention:

$$z_i = \sum_{j=1}^{n} \alpha_{ij} \mathbb{W}_V x_j$$

where $z_i$ represents other symbols in the sequence , i.e., $x_j$, that $x_i$ attends using the soft-alignment weights $\alpha_{ij}$ which is computed as the following softmax function:

$$\alpha_{ij} = \frac{\exp e^{ij}}{\sum_k^n \exp e^{ik}}$$

and sum to 1 over $j$, i.e., $\sum_k^n \alpha_{ij}$. The attention formulation above uses the alignment score $e^{ij}$ between the projection of $x_i$ and $x_j$:

$$e^{ij} = \frac{\left(\mathbb{W}_Q \cdot x_i\right)^T \left(\mathbb{W}_K \cdot x_j\right)}{\sqrt{d_z}}$$

where $d_z$ is the dimension size of the output vector. This alignment score is akin to a dot-product similarity measure with a scaling factor of $\frac{1}{\sqrt{d_z}}$. The weights $\mathbb{W}_K, \mathbb{W}_Q, \mathbb{W}_K \in \mathbb{R}^{d_z \times d_x}$ used for the projections of input representations are parameters that are learned during the training. The output of the multi-head attention layer on each symbol, denoted as $\bar{z}_i$, is then obtained by taking the linear transformation of the concatenation of each attention head's output:

$$\bar{z}_i = \mathbb{W}_O \cdot \textsc{Concat}(z_i^{(1)}, \ldots, z_i^{(p)})$$

where $p$ is the number of heads and $\mathbb{W}_O$ is another learnable parameters. Finally, layer normalization and a fully connected feedforward network (FFN) are applied along with the residual connections (He et al., 2015) to compute the output representation $h_i$ of the transformer layer:

$$\begin{aligned}\tilde{z}_i &= \textsc{LayerNorm}(\bar{z}_i + x_i)\\ h_i &= \textsc{LayerNorm}(\text{FFN}(\tilde{z}_i) + \tilde{z}_i)\end{aligned}$$

The output representation $h_i^{(l)}$ of each tranformer layer $l$ is later used as the input for the next layer $(l+1)$. The output of the final layer is usually used as the representation that encodes rich information about the input sequence. For instance, subsequent work such as BERT (Devlin et al., 2019) uses the representation of a special input symbol produced by the Transformer model as the input to an additional task-specific MLP layer.

### 3.1.3 Language Model Pre-training

#### Contextualized Word Embeddings

Word embeddings are continuous vector representations for each smallest unit of the input text to the neural models introduced above. In practice, these units are obtained using certain tokenization algorithms which may produce units smaller than words such wordpiece (Schuster and Nakajima, 2012; Wu et al., 2016) or BPE (Sennrich et al., 2016).[1] The word embeddings, similar to other neural network weights, can be randomly initialized and learned using task-specific training objectives. However, training from scratch requires more data and *unseen* words during the test will still be represented by random vectors that do not capture the necessary information for the downstream tasks.

Due to these limitations, pre-trained word embeddings (Collobert et al., 2011; Mikolov et al., 2013; Pennington et al., 2014) are instead used in many successful neural models in NLP. These embeddings are trained on a large corpus of text using word co-occurrence information to capture meaningful and general-purpose representations that can be further fine-tuned for specific tasks. Using pre-trained word embeddings as an input to neural architectures such as RNN is shown to help the training and ultimately improve the performance on various NLU tasks, including NLI (Chen et al., 2017) and question answering (Liu et al., 2018).

[1] These units are also commonly referred to as subwords.

Although pre-trained word embeddings offer many advantages, the methods for learning these embeddings only allow for a single, *context-independent* representation of each word. The task-specific models, therefore, still have to learn to contextualize the word representations from the limited number of labeled data, instead of focusing on learning the alignment between the words and the mapping to the target labels. This motivates the development of the *pre-trained neural text encoder* models that assign each token with vector representation which is a function of the entire input sentence. These *contextualized word embeddings* models allow the same words to have different representations depending on their context in a sentence, e.g., the polysemy of the word "*bank*" in "*river bank*" and in "*investment bank*". Furthermore, the embeddings are also shown to capture some aspects of syntax, e.g., part-of-speech tags (Peters et al., 2018). Similar to static word embeddings, adapting to downstream tasks can be performed by taking the output representations as the input to the task-specific models. Two common choices of adaptation include using the pre-trained model as a feature extractor (pre-trained weights are frozen) or directly fine-tuning the pre-trained model for the target task Peters et al. (2019).

### Language Modeling

Language modeling (LM) has emerged as the dominant approach that allows the pre-training of neural text encoders on large unlabeled text data. In recent years, various pre-trained language models, including ELMo (Peters et al., 2018), ULM-FiT (Howard and Ruder, 2018), BERT (Devlin et al., 2019), and GPT, have been introduced and have significantly improved the state-of-the-art performance on a range of natural language understanding tasks. The task of language modeling itself is traditionally formulated to model the likelihood of a sequence of words in a language. Namely, given a sequence of $N$ words, it aims to estimate the probability $P(t_1, t_2, \ldots, t_N)$ which can be written as the product of the probabilities of each individual word, given the previous words in the sentence:

$$P(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} P(t_k \mid t_1, t_2, \ldots, t_{k-1})$$

The above formulation is also commonly referred to as the "*autoregressive*" language model because it uses the previous words or tokens in the sequence to predict the probability of the next word or token. RNN-based models such as ELMo (Peters et al., 2018) usually model both directions of the sequence by jointly maximizing the log-likelihood of the forward and backward directions. Combining the forward and backward LM allows the model to consider both past and future contexts to learn a better representation of a token.

More recently, the Transformers architecture has allowed an alternative language modeling formulation where the probability of a token is estimated given other tokens in the input sequence that may appear before or after it. This offers an advantage over the RNN-based pre-trained LM that can only model each direction separately. In the seminal work that introduces the BERT model (Devlin et al., 2019), the authors put this formulation into practice by using a novel task referred to

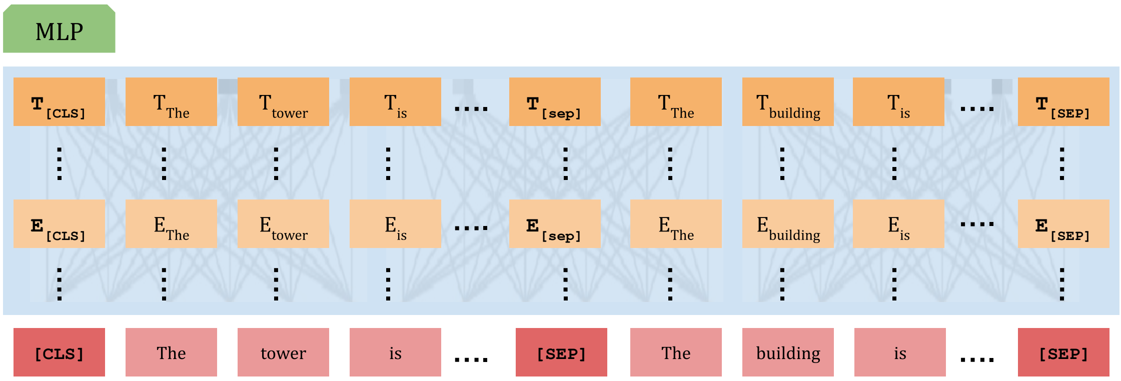


Figure 3.3: An overview of input formulation of text pair NLU classification using Transformers-based language model. The intermediate token representations are denoted as $E$ and the final hidden vectors are denoted as $T$. The final hidden vector for special token [CLS] is fed into the multi-layer perceptron (MLP).

as the Masked Language Model (MLM). The procedure involves randomly selecting some of the words or tokens in the input sequence and replacing them with special [MASK] tokens. Specifically, the input sequence $t = (t_1, t_2, \ldots, t_N)$ is randomly masked into its corrupted form $\tilde{t} = (\tilde{t_1}, \tilde{t_2}, \ldots, \tilde{t_N})$, where $\tilde{t_k}$ can be either the original token or the masking token.[2] The Transformer model is then trained to predict all the tokens that are randomly selected to be "masked", i.e., it estimates $P(t_k \mid \tilde{t_1}, \ldots, \tilde{t_N})$. Finally, the Transformer architecture allows the masked language model pre-training to scale to higher capacity models (bigger number of parameters) and larger text corpus such as BookCorpus (Zhu et al., 2015) and English Wikipedia which contain more than 3B words.

### Transfer Learning Paradigms

The advances in large-scale pre-training of language models are closely followed by novel transfer learning paradigms that have gained prominence across NLU tasks. These paradigms work by assuming that the language model pre-training captures rich linguistic and real-world knowledge that is encoded in the contextualized word or sentence representation of the input text. This useful knowledge can then be transferred to the downstream target tasks via supervised fine-tuning. Specifically, given the availability of labeled task examples, the model weights $\theta$, which is initialized by the pre-training step, is fine-tuned to optimize the likelihood objective $P_\theta(y|x)$, where $x$ is the input text and $y$ is the target label.

Existing approaches for effective transfer learning often involve formatting target task inputs into a single contiguous sequence, similar to how inputs are formatted for language modeling tasks. In paired input tasks such as NLI, the tokens of the sentence pair $(x_1, x_2)$ are concatenated using a special token [SEP] as the separator.

[2] The paper describes that 15% of the tokens are randomly selected for the predictions. Out of the 15%, only 80% are replaced by the special [MASK] tokens, while the remaining are replaced by either random or the original tokens.

Another special token [CLS] is also appended to the front of every input. The final hidden state that corresponds to this [CLS] token is expected to aggregate the representation of the two input sequences and their alignment. Finally, an output layer, typically a multi-layer perceptron, takes this final hidden vector as the input to compute the task-specific label prediction. The output of this layer can either be softmax probability distribution for a classification task or continuous value for a regression task. Figure 3.3 illustrates the input formulation of paired text classification in a fine-tuned Transformers model.

There are varying strategies to fine-tune pre-trained language models for the target tasks. The common practice is to plug in task-specific output layers on top of the pre-trained Transformers model. The weights of the output layers are initialized randomly. During training on the end task, the gradient from the loss function is propagated from the output layers to the pre-trained Transformers layers and their weights are updated. The resulting model, which includes the output layers and the Transformer layers, is retrofitted for the specific end task.

More recently, prompting has emerged as a promising alternative paradigm in adopting pre-trained language models for downstream tasks. The new approach reformulates downstream tasks as fill-in-the-blank problems, in which specific words correspond to the target labels (Schick and Schütze, 2021). Language models, which are pre-trained to predict missing words, can thus transfer the knowledge acquired in pre-training more directly to the target tasks. Given a textual prompt for the target task, pre-trained language models make predictions by mapping the most probable word to the associated label. For instance, in the Paraphrase Identification task, we can formulate the sentence pair in 2.1 as:

> "*The skyscraper is under construction* **?** [MASK] **,** *the building is under construction.*"

We can then use the language model prediction on the missing [MASK] token. Probabilities that the model assigns to the token "**Yes**" and "**No**" can be compared to make a classification decision between *paraphrase* and *no-paraphrase* label. The prompting paradigm does not require newly initialized layer extension to the pre-trained models and thus allows them to perform on zero-shot settings. Recent studies also show that prompting formulation can further improve data efficiency of fine-tuning on low resource settings (Gao et al., 2021).

## 3.2 Thesis Contribution

In this thesis, we focus mainly on improving the robustness of the state-of-the-art modeling approach for NLU tasks, e.g., BERT (Devlin et al., 2019) and Roberta (Liu et al., 2019b).[3] Namely, we evaluate our proposed methods on Transformers-based pre-trained language models which are fine-tuned with task-specific supervision.

[3] At the time of the publication of our work, BERT model (Devlin et al., 2019) and its derivations are the top performing models for various NLU benchmarks including GLUE Wang et al. (2018).

Even though more recently introduced models have outperformed our evaluated models, Transformers architectures and language model fine-tuning remain the essential building blocks of these newer models, e.g., T5 (Raffel et al., 2020). In what follows, we detail how the significance of the contribution in each chapter to the existing NLU modeling approaches that we discuss in previous sections:

- In Utama et al. (2020a), we propose a regularized objective function that disincentivized models from exploiting non-robust features during training. While we perform the evaluation on Transformers models, the proposed method is architecture-agnostic and can be applied to different architecture and training paradigms. Concurrent work of He et al. (2019) applies a similar robustness improvement method to non-Transformers models such as Decomposable Attention (Parikh et al., 2016) and ESIM (Chen et al., 2017). They show that their method also improves the robustness of these models. This suggests that our proposed method is highly likely to also benefit other types of models.

- We then investigate in Utama et al. (2020b) the learning dynamics of pre-trained language models during fine-tuning and establish the connection with the robustness against spurious correlations. Even though our analysis in this work focuses on the BERT model, we note that a similar tendency of neural models to exploit simple patterns early in the training is also reported in other model architectures in varying domains (Arpit et al., 2017; Liu et al., 2020). We, therefore, believe that the insight from our work will still be highly relevant to future modeling approaches which are predominantly based on deep learning.

- Next, in Utama et al. (2021), we extend our study on robustness to the recently introduced prompting paradigm which allows pre-trained language models to perform well on low resource NLU settings. As more language models with increased general-purpose NLU capabilities emerge, the advances in prompting approaches are increasingly more important (Liu et al., 2023). Consequently, the insight and the proposed method from our work remain relevant for improving the robustness of prompt-based NLU models.

- Finally, in Utama et al. (2022), we propose a data generation framework that improves the generalization of NLU models for the downstream task of summarization. The framework and the resulting datasets are model or architecture-specific and therefore can be extended to more recent approaches. Better-performing pre-trained LM can benefit from our work by using our generated dataset to augment the targeted NLI datasets.

# Chapter 4

# Robustness in Natural Language Understanding

In this chapter, we summarize and discuss a growing body of literature that highlights the shortcomings of NLU datasets and the brittleness of the resulting models. We first look at the methods proposed to quantify the lack of robustness in NLU systems. We then discuss the existing effort in both dataset collection and modeling to improve robustness.

## 4.1 Robustness Analysis Methods

We distinguish between the methods to investigate the issues in the training and evaluation data, and the methods to measure the impact of the flaws in the dataset to the models' performance:

### 4.1.1 Datasets Investigation

Realistic datasets for NLU tasks are prone to contain spurious correlations which are characterized as surface features that are predictive to the target label but are irrelevant to the underlying task. As such, they fail to transfer to the out-of-distribution data for the same task. The occurrence of these spurious correlations can be traced back to a large extent to the artifacts of the data collection and annotation processes. During the collection step, the selected examples to be annotated can be biased with respect to the features and the target labels, e.g., the frequency of a unique sentence occurrence in the Quora Question Pairs dataset is reported to be highly correlated with the "*duplicate*" label (Zhang et al., 2019). Furthermore, once the data are selected for annotation, human annotators are likely to adopt simple strategies to maximize the output. As an example, *contradicting* sentences for the NLI task can simply be obtained by negating the given sentence by adding negation words such as "*don't*" or "*never*". This shortcut, if disproportionately employed by the annotators, results in a spurious correlation between the negation words and the target label "*contradiction*" which models will likely capture.

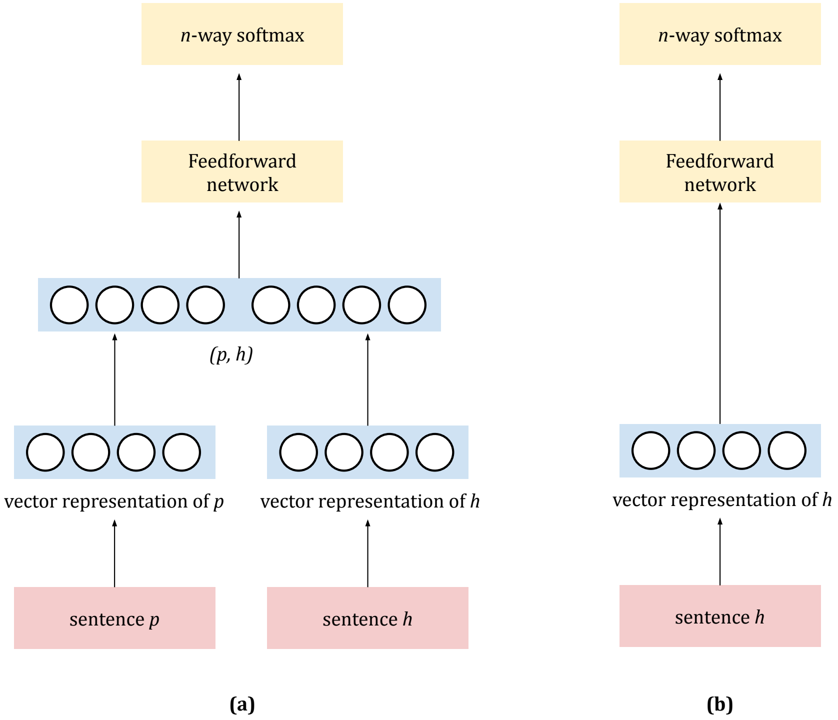


Figure 4.1: **(a)** shows a typical bi-encoder architecture approach for NLI that encodes both the premise $p$ and hypothesis $h$ before classifying them. **(b)** illustrates the partial input baseline method that excludes the premise and only encodes the hypothesis sentence to make the predictions.

Existing investigation methods are proposed to quantify these spurious correlations and to identify examples that support them. In NLU tasks where an input consists of a text pair, *Partial Input Baselines* method is used to reveal whether all parts of the input are necessary for making the correct predictions in the task. Gururangan et al. (2018); Poliak et al. (2018b); Tsuchiya (2018) propose to train classifiers for NLI tasks using only the hypothesis sentence to predict the entailment relationship. As illustrated in Figure 4.1, given the premise sentence $p$ and the hypothesis sentence $h$, they train the model to estimate the target label using only $h$, i.e., $P(y \mid h)$, instead of $p(y \mid p, h)$. Ideally, the partial input $h$ should not be informative and the label $y$ should not be determinable without the premise sentence $p$. However, they find that the partial input models perform significantly above the random baselines. This indicates the presence of predictive features in the hypothesis sentences that models can utilize to make correct predictions. Similar artifacts that the partial input baseline method reveals also occur in many other paired input NLU tasks. For instance, in reading comprehension question answering tasks, Kaushik and Lipton (2018) show that models achieve non-trivial performance by looking only at either the context passage text or the question sentence.

The predictions of the partial input models can be used to identify whether an example is consistent with the spurious correlation (Gururangan et al., 2018). Namely, examples are considered as *easy* when the model, which relies solely on the

| **Task** | **Artifact** | **Example** |
|---|---|---|
| NLI | Lexical cues in the hypothesis sentence | *Premise:*<br>Two dogs are running through a field.<br>*Hypothesis:*<br>The animals are **sleeping**.<br>*Label: contradiction* |
| NLI | Lexical overlap between the premise and hypothesis sentences | *Premise:*<br>**A child** was **pulled by a woman on a sled**.<br>*Hypothesis:*<br>**A woman pulled a child on a sled**.<br>*Label: entailment* |
| Paraphrase Identification | Lexical overlap between the input sentence pair | $S_1$:<br>**Flights from Florida to NYC**<br>$S_2$:<br>**Flights to NYC from Florida**<br>*Label: duplicate* |
| Fact Verification | Lexical cues in the claim sentence | *Evidence*:<br>Johnson played point guard for the Lakers for 13 seasons. [...]<br>*Claim*:<br>Johnson **did not** play for the Lakers.<br>*Label: refutes* |

Table 4.1: Examples of known dataset artifacts in several NLU tasks. Input words in boldface are cues that models can potentially use to make correct predictions.

spurious correlation, assigns them to the correct labels. The remaining examples are then considered to be the *hard* subset of the dataset. These harder examples, however, are not necessarily free of spurious features. Exploitable shortcuts can occur when both of the input texts are considered by the models. Feng et al. (2019) demonstrate, in both synthetic and real NLI datasets, the presence of shortcut features that are only visible when both input texts are combined. Firstly, they show how a combination of synthetic "code" words that are appended to both the premise and hypothesis can give away the target label. Next, they show that hypothesis-only model performance improves even more by simply adding the last noun word from the premise sentence to the input. The improvement from the addition of this supposedly non-predictive feature is an indication of spurious correlation that partial input baselines fail to detect. McCoy et al. (2019) further identify a more concrete spurious correlation beyond partial input features. They demonstrate that the lexical overlap between premise and hypothesis sentences in popular NLI datasets is strongly correlated with the "entailment" label, which they expect to be easily exploited by the models.[1] Table 4.1 shows examples of several known dataset artifacts in various NLU tasks.

[1] Premise sentence "*The doctor near the actor danced.*" does not entail the hypothesis "*The actor danced.*" even though they consist of overlapping words.

| **Dataset** / **Task** | **Heuristic** | **Counter-example** |
|---|---|---|
| **HANS**<br>McCoy et al. (2019)<br>/ **NLI** | Assumes "entailment" if the premise and hypothesis highly overlap. | *Premise:*<br>**The artist** was **paid** by **the lawyer**.<br>*Hypothesis:*<br>**The artist paid the lawyer**.<br>*Label: not-entailment* |
| **PAWS**<br>Zhang et al. (2019)<br>/ **Paraphrase Identification** | Assumes "duplicate" if the sentence pairs highly overlap. | $S_1$:<br>Can a **bad** person become **good**?<br>$S_2$:<br>Can a **good** person become **bad**?<br>*Label: non-duplicate* |
| **Symmetric**<br>Schuster et al. (2019)<br>/ **Fact Verification** | Assumes "refutes" label if the claim sentence contains a negation. | *Evidence*:<br>Johnson played for **the Giants and no other team**. [...]<br>*Claim*:<br>Johnson **did not** play for the Lakers.<br>*Label: supports* |

Table 4.2: The examples of several heuristics that describe the shallow features models use along with the expected labels. The counter-examples demonstrate cases where the heuristics lead to wrong predictions.

### 4.1.2 Model diagnosis

Most NLU datasets are commonly split randomly to obtain the train, validation, and test subsets. The spurious correlations that models pick up in the training set therefore also present in the test data used to evaluate the performance. The results of models' evaluation on this *in-distribution* test data may not be therefore a reliable estimate of their true NLU capabilities. To illustrate the drawback of this standard evaluation, Nie et al. (2019) train NLI models on examples in which the input word orders are randomly shuffled while keeping the labels the same. They then evaluate the resulting models on the original evaluation data. They observe that this procedure does not substantially degrade the models' performance on the evaluation sets. These results provide evidence that the in-distribution test data may be insufficient to distinguish between models that sufficiently capture compositional semantics of the input and models that are mostly reliant on spurious correlation. Gauging models' robust performance, therefore, requires evaluating models on test sets that are carefully designed to minimize spurious correlations.

Previous work designed and collected such test sets, which are commonly referred to as *out-of-distribution* (OOD) or *challenge sets*, based on the insight about the characteristics of the spurious features. For instance, McCoy et al. (2019) introduce the HANS evaluation set to diagnose whether NLI models have adopted syntactic heuristics, i.e., models relying on word overlap cue between input sentence pair to predict "*entailment*" label. All examples in HANS consist of sentence pairs that exhibit the lexical overlap cue. These examples are generated using templates and rules such that they can *support* (labeled as "*entailment*") or *against* (labeled as "*not-*

*entailment*") the syntactic heuristics. Table 4.2 shows counter-examples to several known heuristics in NLU. Poor performance on these examples that are inconsistent with the heuristics and high scores on the other set is interpreted as an indication of models' reliance on the spurious correlation.

Besides using templates, challenge sets are also collected for varying tasks using other automatic or manual approaches. These methods include adversarial input perturbation (Glockner et al., 2018; Naik et al., 2018; Jia and Liang, 2017) or manual human annotation (Schuster et al., 2019; Kaushik et al., 2020). Even though these test sets are designed to target different types of phenomena, they collectively show performance degradation of NLU models relative to the standard evaluation results. Previous work later look at whether models can be "*inoculated*" against phenomena-specific weaknesses by fine-tuning them on the small portion of the *challenge sets* (Liu et al., 2019a). They observe that fine-tuning improves the scores on the remaining examples of the challenge set with only a negligible drop in the original data performance. However, Rozen et al. (2019) demonstrate that this performance improvement does not necessarily indicate systematic generalization to the phenomena of interest, but rather that models overfit on the specifics of the *challenge set*. These results illustrate that the utility of *challenge sets* is mostly limited for evaluation and that more naturalistic OOD datasets are required for augmenting the training data. Furthermore, collecting new training examples each time new artifacts are identified is costly. Adjustments to the model architecture and training procedure are equally crucial to avoid learning spurious correlations in the datasets.

## 4.2 Robustness-improving Methods

We broadly classify the existing approaches aimed to mitigate the learning of spurious correlation into two categories: training data refinement and model-centric training adjustment. The improvement in robustness from these approaches is typically estimated by models' out-of-distribution performance increase on the challenge sets. It is also crucial that models achieve the OOD improvement while still maintaining the predictive performance in the original in-distribution evaluation set.

### 4.2.1 Datasets Quality Improvement

Existing approaches that aim to improve the quality of the datasets with respect to spurious correlation can be broadly categorized as **(a)** approaches to build entirely new datasets that do not contain the undesired artifacts; and **(b)** approaches that refine the *existing datasets* by identifying and removing instances that contribute to the presence of the artifacts. The two sets of approaches are not mutually exclusive and can be applied simultaneously to complement each other's advantages. For instance, a newly created dataset for the task can still be refined to remove the remaining or newly identified artifacts to improve the quality further. In what follows, we discuss several specific methods for dataset construction or refinement:

**Enhanced Annotation Protocol** By identifying and characterizing the weak points of a dataset that cause robustness issues, researchers are able to design novel annotation protocols to alleviate the known shortcomings. As an example, the work by Sharma et al. (2018) identified that models for the task of Story Cloze Test (Mostafazadeh et al., 2016) are mostly reliant on the stylistic features of the partial story input rather than comprehending the narrative context. They then design a refined annotation guideline to reduce the statistical difference between the positive and negative samples on the surface level. The guideline imposes restrictions on the number of tokens to write, token n-grams to use, or the changes of topic or sentiment between the sentences of opposing labels. Similarly for NLI, Han et al. (2020); Hu et al. (2020) employ several strategies to elicit diverse hypotheses generation from the annotators that contain less bias. The following are several constraints and encouragements in the new annotation guideline along with the resulting examples:

1. Constraint: write a contradicting hypothesis sentence that does not contain negation words.

   ▹ **P:** "Going there at the end of October, be back at the end of November."

   ▹ **H:** "Will stay there for two months before coming back."

   ▹ **Label:** contradiction

2. Encouragement: write an entailing hypothesis sentence whose tokens overlap with the premise by at most 70%.

   ▹ **P:** "Yes, look, what he talked about is very interesting."

   ▹ **H:** "What he talked about has caught my attention."

   ▹ **Label:** entailment

While this work has demonstrated the importance of the specificity of annotation instruction, it does not study systematic ways to verify whether the instructions can be followed or need to be iteratively updated throughout the annotation process. Parrish et al. (2021) investigate several strategies to put the expertise and knowledge of linguists in the loop during data collection. As more samples are collected, these experts are asked to dynamically assess the annotated data to identify any artifacts or other weaknesses. As issues arise, the experts can notify the non-expert annotators and advise them to employ alternative annotation strategies accordingly. They show that NLI models that are fine-tuned on their collected data perform better on the *challenge* datasets. This further suggests that improving the annotation protocol has significant implications for the robustness of the resulting models.

**Counterfactual Perturbation** The quality of the datasets can also be improved by augmenting them with counterfactual examples, i.e., perturbed examples that preserve the original features except for a key attribute that can flip the label. The idea is to have minimally differing example texts with contrasting labels in the training data so that models are able to learn the difference that makes a difference.

Kaushik et al. (2020); Gardner et al. (2020) propose strategies to obtain counterfactual examples by asking annotators to minimally existing labeled examples so that they correspond to the counterfactual label while preserving the original characteristics such as coherence and stylistic features. For the NLI task, the manual perturbation can be applied to the hypothesis sentence as in the following example:

- ▷ **P:** "Several farmers bent over working on the fields while lady with a baby and four other children accompany them."
- ▷ **Orig. H:** "The lady has **three** children." **(contradiction)**
- ▷ **New H:** "The lady has **many** children." **(entailment)**

The numerical modification in the example above (boldface) is sufficient to flip the label of the existing pair from *contradiction* to *entailment*. Having these two examples in the training dataset can provide an incentive for the model to learn a more robust representation of quantifier words that allow better numerical reasoning. Kaushik et al. (2020); Khashabi et al. (2020) demonstrate that training models on counterfactually augmented datasets improves robustness and generalization as shown by the out-of-distribution performance.

**Model–in-the-Loop** Researchers also investigated adversarial methods which involve other models that adversarially reduce the artifacts during the data collection process. The adversarial models are employed through different mechanisms including filtering out examples that exhibit the artifacts (Sakaguchi et al., 2020; Le Bras et al., 2020; Zellers et al., 2018) or to prompt annotators to generate more challenging examples (Nie et al., 2020). While these adversarial models are shown to reduce the presence of several known dataset artifacts (e.g., hypothesis-only biases in NLI), many of them are not explicitly designed to target specific types of artifacts. This is in contrast with the top-down approaches mentioned above where the intuitions and insights of the researchers about the characteristics of the biases are required to apply the intervention during the annotation. For instance, in AFLite algorithm (Sakaguchi et al., 2020), an ensemble of weak learners (logistic regression models) are trained on different subsets of the annotated dataset and tested on their corresponding test sets. The predictions of these models on each instance are then aggregated and used to determine whether the instance should be filtered or not. More specifically, if the ratio of correct predictions over the total number of predictions on a given instance is higher than the threshold $\tau$, then the example is likely to exhibit biases and will be filtered. This process is performed iteratively until the number of filtered instances is below $k$ or until the number of remaining dataset instances is below $m$.[2] Similarly, Nie et al. (2020) employs an iterative approach in which the adversarial models are expected to have stronger predictive performance at each iteration. This strategy aims to elicit annotators to produce higher quality and more challenging examples that expose the weaknesses of the increasingly

[2] The variables $\tau$, $k$, and $m$ are hyperparameters that the authors set during the construction of their dataset.

strong adversarial models. Specifically, annotators are asked to write task instances that satisfy the target label but are predicted incorrectly by the current best model. The collected data from each round are then used to re-train the model for the next iteration. They show that the proposed framework incentivizes the annotators to be more creative in devising examples that contain a wider range of inference types (Williams et al., 2022).

**Rule-based Data Augmentation** Finally, researchers also study the use of rule-based text perturbation to generate examples that reduce the spurious correlation when augmented into the existing datasets. The perturbation rules are usually designed based on insights about the targeted artifacts in the datasets. For instance, Naik et al. (2018) propose a perturbation rule that addresses the spurious correlation between negation words such as "not" with the contradiction label in the NLI task. They construct examples with negation words that are not necessarily labeled as a contradiction, such as the following:

- ▷ **P:** "Possibly no other country has had such a turbulent history."
- ▷ **H:** "The country's history has been turbulent ***and false is not true***."
- ▷ **Label:** Entailment

While the appended tautology "***and false is not true***" contains the known cue word "not" (Gururangan et al., 2018), the statement is independently true in all world and its conjunction with the original hypothesis sentence will preserve the label. Training models on such examples can be useful to reduce the effects of the existing artifacts (Liu et al., 2019a). However, Rozen et al. (2019) highlighted the limitation of this augmentation approach, where the training and evaluation are performed on the synthetic examples that are drawn from the same distribution. Min et al. (2020); Rozen et al. (2019) recommend that the construction of these augmentation examples should be diversified on various factors, such as syntactic complexity or lexical variations. Evaluating models on different splits based on the level of complexity can therefore ensure a more reliable estimation of the models' generalization.

### 4.2.2 Model Training Improvement

Model-centric improvement approaches seek to reduce the adverse impact of spurious correlations that potentially still exist in the training dataset. These approaches introduce various mechanisms through which models are incentivized to preferably use more robust features that generalize well for the underlying task. In what follows, we discuss several categories of model improvement with respect to robustness:

**Pre-training** Researchers have been studying the connection between models' pre-training with the out-of-distribution (OOD) robustness. Hendrycks et al. (2020)

systematically compare pre-trained Transformers such as RoBERTa against non-pretrained models such as LSTM on OOD generalization setup. They found that the relative decreases in OOD performance (compared to the in-distribution) of pre-trained Transformers are significantly lower. Tu et al. (2020) further show that the RoBERTa model, which has a similar architecture to its predecessor BERT model but is trained on ten times as much text data, performs more reliably on the OOD examples. They also show that when other factors are controlled (e.g., training data, pre-training objective, model architectures) the size of the model contributes significantly to the overall robustness. A study by Lovering et al. (2021) provides an explanation of the effectiveness of models' pre-training to improve robustness. They introduce a notion of *extractibility* of rich features of linguistic information in the pre-trained models which can be measured using information-theoric probing methods (Hewitt and Manning, 2019; Voita and Titov, 2020). They then show that the extent to which useful linguistic features are preferably used (instead of the non-robust spurious correlations) is determined by the degree of *extractibility* and the amount of statistical *evidence* available in the fine-tuning dataset. More evidence, which is defined as the co-occurrence rate between the robust features and the target label, may be required for pre-trained models with weaker *extractibility* and *vice versa*. Pre-training on larger and more diverse text data with larger capacity models can increase the degree of robust features *extractibility* which results in higher OOD performance despite the same fine-tuning datasets.

**Leveraging External Resources** Various approaches have been explored to leverage external knowledge to regularize the fine-tuning and minimize the effects of spurious correlations. Tu et al. (2020) applies multi-task learning (Caruana, 1997) to jointly learn the target task using the main dataset and several *auxiliary datasets* of other related tasks. The idea is to allow the useful knowledge from other tasks to transfer to the main task and to counter the existing spurious correlations. Specific external tasks such as semantic role labeling (SRL) are widely used as they provide the explicit signal for the model to learn and utilize syntactic and semantic information that is useful for the target task. The auxiliary supervision from the semantic role labeling instances can either be used to decompose and re-align the input text (Chen and Durrett, 2021; Wu et al., 2019), augmented directly to the target task input (Moosavi et al., 2020), or learned jointly with the main task (Cengiz and Yuret, 2020). Lastly, researchers also investigate the use of task *explanation* to discourage models from exploiting spurious correlations. Camburu et al. (2018) introduce e-SNLI corpus which extends the existing SNLI dataset (Bowman et al., 2015) with sentential justification texts which are collected through human annotation. Consider the example below from their paper:

- ▷ **P:** "P: An adult dressed in black **holds a stick**."
- ▷ **H:** "An adult is walking away, **empty-handed**."
- ▷ **Label:** Contradiction
- ▷ **Explanation:** "Holds a stick implies using hands so it is not empty-handed."

Each SNLI instance is annotated by highlighting phrases that human considers to be salient for inferring the target label. They show that models, which are trained to jointly predict the NLI label and generate the explanation tokens, generalize better to OOD datasets. Using this resource, Stacey et al. (2022) propose an approach to explicitly supervise models' self-attention weights to add more attention to important tokens according to human explanation. They show that the resulting NLI models perform better on both the in-distribution OOD evaluation while also become more interpretable.

**Adversarial Training** In the context of addressing the spurious correlations in NLU, adversarial training is applied to minimize the information about the non-robust features in the learned representation of the input text. Earlier work by Belinkov et al. (2019a), proposes the use of two models that share the same text encoder and are trained jointly. Specifically, the first model is the main classifier for the target task while the other model is the adversarial classifier which aims to predict the presence of the artifact attributes in the input text. The adversarial classifier is a limited capacity model such as a hypothesis-only model, which is designed based on prior knowledge about the characteristics of the artifacts. Using the gradient reversal layer method (Ganin and Lempitsky, 2015), they train the encoder and the main task classifier to optimize the target task objective while degrading the ability of the adversarial classifier to predict the artifacts. While conceptually promising, the strength of the adversaries is crucial to ensure that the artifacts do not remain hidden in the representation.[3] Stacey et al. (2020) address this by ensembling a set of adversarial classifiers which increase the overall strength of the adversary.

**Debiasing Loss Function** Removing biased features in the text representation may degrade the overall performance since these features often conflate broadly useful semantic information and surface-level cues, e.g., negation words. Instead of explicitly removing certain information from the representation, more recent methods aim to improve robustness by emphasizing the learning on "harder" examples where the simple features are insufficient to make correct predictions. This incentivizes models to learn more robust representations and inference strategies that generalize to out-of-distribution examples. There are multiple mechanisms through which models can focus on challenging examples. They typically involve the modification of the loss functions to take into account the presence of the artifacts in each training instance. The measure of artifacts in an instance can be measured by a hand-crafted weighting scheme based on word occurrence (Schuster et al., 2019) or by using the output of biased models that are trained to solely rely on spurious correlations. The simple method to incorporate the measure of bias into the training is by reweighting the individual loss term on each training instance. Specifically,

[3] Belinkov et al. (2019b) show that the adversarial classifiers can indeed re-gain high performance after being re-trained separately on the frozen text encoder which means that the artifacts are not fully removed from the representations.

given a training example $x_i$ labeled with $y_i$ with bias weight of $b_i \in [0, 1]$, with 0 means that the example is bias-free, the loss from that particular example is defined as:

$$\mathcal{L}_i = -(1 - b_i) \cdot y_i \cdot \log p$$

where $p$ is the probability assigned by the model to the label $y_i$ using the output of the softmax layer. The weight term in the loss function means that the contribution of an example to the overall loss is gradually decreased as the bias measure increases. An alternative mechanism to reweighting involves an ensembling of the main model with the biased model. As proposed by the existing work (Clark et al., 2019; He et al., 2019; Karimi Mahabadi et al., 2020), the prediction of the two models can be combined using product-of-expert method (Hinton, 2002). Specifically, the ensembled predicted probability $p_i$ is obtained by taking the sum of the biased model's prediction $p_b$ and the main model's prediction $p_d$ in the logarithmic space:

$$p_i = \texttt{softmax}(\log p_d + \log p_b)$$

The ensembling of the models allows the main model to learn to fit the residual of the biased model prediction (He et al., 2019). Intuitively, this residual represents a signal on the information in the input that cannot be captured using only the biased features or the spurious correlations. Examples that are predicted well by the biased model with low loss will provide a weak signal for the main model to learn from. On the contrary, high loss by the biased model on an example incentivizes the main model to learn an inference strategy that is less reliant on the biased features.

## 4.3 Thesis Contribution

In what follows we discuss the relevance of the contributions of this thesis to the robustness-improving methods that we discussed above.

- We build upon the work that investigates the datasets and the models to identify and characterize the robustness issues across NLU tasks (Section 4.1). Specifically, in Utama et al. (2020a), we propose a robustness-improving method that mitigates models' reliance on spurious correlation based on the knowledge about the bias in the datasets. This knowledge is used to design a bias-only model which quantifies the presence of biased features in each training instance. The proposed confidence regularization method that we introduce addresses the drawback of the existing *debiasing loss methods* which degrade the in-distribution performance as a trade-off with the OOD performance improvement. Our novel regularization strategy prevents the model to exploit the biased features by discouraging overconfident predictions on easier examples that can be solved by relying on spurious correlation. This allows the model to still learn from these examples which in turn improves the OOD performance without hurting the existing in-distribution performance.

- While effective, the above-proposed method still relies on the insight from the robustness analysis work. The insight and knowledge about the biases may be limited for newly collected NLU tasks or datasets. In Utama et al. (2020b), we, therefore, proposed a framework to automatically quantify the presence of biases in each training instance with minimum prior knowledge about the dataset. Related to the model-in-the-loop approaches we discussed above, the proposed framework utilizes an additional model that is trained to be a bias-reliant model. We train this model based on our finding that the bias-reliant behavior of a model is most prominent during the early stage of the training. We then find that intentionally allowing the model to overfit to a very small portion of the training data can amplify this behavior. The predictions of this model can therefore be used by the existing debiasing methods as a proxy measure of biases on each example.

- As previously discussed, pre-training of language models has been shown to be a crucial determiner of robustness against spurious correlation. In Utama et al. (2021), we systematically study the importance of pre-training representation using a prompt-based approach where the OOD performance of the models can be directly evaluated even without task-specific fine-tuning and the introduction of newly added model weights. This allows us to find that pre-trained language models perform surprisingly well on zero-shot settings and that the fine-tuning, even only with a small number of target task examples, can be destructive to this OOD performance. This motivates our proposed regularized fine-tuning objective that allows the prompt-based model to optimize the target task on a few shot settings without significantly degrading the existing OOD performance.

- Lastly, in Utama et al. (2022), we propose a data augmentation strategy to improve robustness which is oriented towards the better downstream application of NLU models. Specifically, we introduce a data generation framework that produces high-quality negative examples for document-level NLI tasks. The resulting NLI models are capable of performing NLI predictions that are reliable for the downstream task of detecting factuality in document summarization. This work shows that the quality of the augmented data, in terms of naturalness and diversity, plays a crucial role in improving the robustness toward the intended applications.

## 4.4 Related Topics

**Fairness** Studies on robustness in NLU are closely related to a body of work on fairness in machine learning. Some work in fairness aims to remove sensitive features (e.g., race or gender information) from the learned representation of the model. For instance, Zhao et al. (2018); Prost et al. (2019) propose methods to learn word embeddings that do not have gender information. This is not applicable to NLU since biased features often conflate useful linguistics information with shallow cues (e.g., negation words in the NLI tasks). Existing work on the robustness of NLU, therefore, focus on improving how models learn more robust inference strategies from the training examples that still contain biases.

**Domain Generalization and Adaptation** Robustness in NLU is related to a broader problem of *domain generalization* and *domain adaptation*. Existing work in *domain generalization* are similar in that they also seek to mitigate the tendency of models in learning domain-specific spurious features (Ganin et al., 2016). On the contrary, domain *adaption* aims to increase the generalization of models to a specific set of target domains (Wang et al., 2019), whereas the goal of both robust learning in NLU and domain generalization are to generalize over unforeseen distribution shifts (Hendrycks et al., 2020).

# Part II

# Publications

# Chapter 5

# Improving Robustness against Known Spurious Features

# Mind the Trade-off: Debiasing NLU Models without Degrading the In-distribution Performance

**Prasetya Ajie Utama**[†‡] **, Nafise Sadat Moosavi**[‡]**, Iryna Gurevych**[‡]

[†]Research Training Group AIPHES
[‡]Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
https://www.ukp.tu-darmstadt.de

## Abstract

Models for natural language understanding (NLU) tasks often rely on the idiosyncratic biases of the dataset, which make them brittle against test cases outside the training distribution. Recently, several proposed debiasing methods are shown to be very effective in improving out-of-distribution performance. However, their improvements come at the expense of performance drop when models are evaluated on the in-distribution data, which contain examples with higher diversity. This seemingly inevitable trade-off may not tell us much about the changes in the reasoning and understanding capabilities of the resulting models on broader types of examples beyond the small subset represented in the out-of-distribution data. In this paper, we address this trade-off by introducing a novel debiasing method, called *confidence regularization*, which discourage models from exploiting biases while enabling them to receive enough incentive to learn from all the training examples. We evaluate our method on three NLU tasks and show that, in contrast to its predecessors, it improves the performance on out-of-distribution datasets (e.g., 7pp gain on HANS dataset) while maintaining the original in-distribution accuracy.[1]

## 1 Introduction

Despite the impressive performance on many natural language understanding (NLU) benchmarks (Wang et al., 2018), recent pre-trained language models (LM) such as BERT (Devlin et al., 2019) are shown to rely heavily on idiosyncratic biases of datasets (McCoy et al., 2019b; Schuster et al., 2019; Zhang et al., 2019). These biases are commonly characterized as *surface features* of input examples that are strongly associated with the target labels, e.g., occurrences of negation words in natural language inference (NLI) datasets which are biased towards the *contradiction* label (Gururangan et al., 2018; Poliak et al., 2018). As a ramification of relying on biases, models break on the *out-of-distribution* data, in which such associative patterns between the surface features and the target labels are not present. This brittleness has, in turn, limited their practical applicability in some extrinsic use cases (Falke et al., 2019).

This problem has sparked interest among researchers in building models that are robust against *dataset biases*. Proposed methods in this direction build on previous works, which have largely explored the format of several prominent label-revealing biases on certain datasets (Belinkov et al., 2019). Two current prevailing methods, *product-of-expert* (He et al., 2019; Mahabadi and Henderson, 2019) and *learned-mixin* (Clark et al., 2019a) introduce several strategies to overcome the *known* biases by correcting the conditional distribution of the target labels given the presence of biased features. They achieve this by reducing the importance of examples that can be predicted correctly by using only biased features. As a result, models are forced to learn from harder examples in which utilizing solely superficial features is not sufficient to make correct predictions.

While these two state-of-the-art debiasing methods provide a remarkable improvement on the targeted out-of-distribution test sets, they do so at the cost of degrading the model's performance on the *in-distribution* setting, i.e., evaluation on the original test data which contains more diverse inference phenomena. It raises a question on whether these debiasing methods truly help in capturing a better notion of language understanding or simply biasing models to other directions. Ideally, if such an improvement is achieved for the right reasons (i.e., better reasoning capabilities by learning a more general feature representation), a debiased model

[1] The code is available at https://github.com/UKPLab/acl2020-confidence-regularization

|  | product-of-expert | learned-mixin | **conf-reg (our)** |
|---|---|---|---|
| in-distribution | ▼ | ▼ | ▲ |
| out-of-distribution | ▲ | ▲ | ▲ |
| calibration | △ | ▼ | ▲ |
| requires biased model | ✔ | ✔ | ✔ |
| requires hyperparameter | ✖ | ✔ | ✖ |

Table 1: Comparison of our method against the state-of-the-art debiasing methods. Learned-mixin (Clark et al., 2019a) is a parameterized variant of Product-of-expert (He et al., 2019; Mahabadi and Henderson, 2019). Our novel confidence regularization method improves the out-of-distribution performance while optimally maintain the in-distribution accuracy.

should still be able to maintain its accuracy on previously unambiguous instances (i.e., instances that are predicted correctly by the baseline model), even when they contain biases.

In this work, we address this shortcoming by introducing a novel debiasing method that improves models' performance on the out-of-distribution examples while preserves the in-distribution accuracy. The method, called *confidence regularization*, draws a connection between the robustness against dataset biases and the overconfidence prediction problem in neural network models (Feng et al., 2018; Papernot et al., 2016). We show that by preventing models from being overconfident on biased examples, they are less likely to exploit the simple cues from these examples. The motivation of our proposed training objective is to *explicitly* encourage models to make predictions with lower *confidence* (i.e., assigning a lower probability to the predicted label) on examples that contain biased features.

Table 1 shows the comparison of our method with the existing state-of-the-art debiasing methods: *product-of-expert* and *learned-mixin*. We show that our method is highly effective in improving out-of-distribution performance while preserving the in-distribution accuracy. For example, our method achieves 7 points gain on an out-of-distribution NLI evaluation set, while slightly improves the in-distribution accuracy. Besides, we show that our method is able to improve models' calibration (Guo et al., 2017) so that the confidences of their predictions are more aligned with their accuracies. Overall, our contributions are the following:

- We present a novel *confidence regularization* method to prevent models from utilizing biased features in the dataset. We evaluate the advantage of our method over the state-of-the-art debiasing methods on three tasks, including natural language inference, fact verification, and paraphrase identification. Experimental results show that our method provides competitive out-of-distribution improvement while retaining the original in-distribution performance.
- We provide insights on how the debiasing methods behave across different datasets with varying degrees of biases and show that our method is more optimal when enough bias-free examples are available in the dataset.

## 2 Related Work

**Biases in Datasets** Researchers have recently studied more closely the success of large fine-tuned LMs in many NLU tasks and found that models are simply better in leveraging biased patterns instead of capturing a better notion of language understanding for the intended task (Bender and Koller, 2020). Models' performance often drops to a random baseline when evaluated on out-of-distribution datasets which are carefully designed to be void of the biases found in the training data. Using such targeted evaluation, McCoy et al. (2019b) observe that models trained on MNLI dataset (Williams et al., 2018) leverage syntactic patterns involving word overlap to blindly predict entailment. Similarly, Schuster et al. (2019) show that the predictions of fact verification models trained for the FEVER task (Thorne et al., 2018) are largely driven by the presence of indicative words in the input claim sentences.

Following similar observations across other tasks and domains, e.g., visual question-answering (Agrawal et al., 2016), paraphrase identification (Zhang et al., 2019), and argument reasoning comprehension (Niven and Kao, 2019), researchers proposed improved data collection techniques to reduce the artifacts that result in dataset biases. While these approaches are promising, only applying them without additional efforts in the modeling part may still deliver an unsatisfactory outcome. For instance, collecting new examples by asking human annotators to conform to specific rules may be costly and thus limit the scale and diversity of the resulting data (Kaushik et al., 2020). Recently proposed adversarial filtering methods (Zellers et al., 2019; Sakaguchi et al., 2019) are more cost effective but are not guaranteed to be artifacts-free. It is, therefore, crucial to develop learning methods that can overcome biases as a complement to the data collection efforts.

**Debiasing Models** There exist several methods that aim to improve models' robustness and generalization by leveraging the insights from previous work about the datasets' artifacts. In the NLI task, Belinkov et al. (2019) make use of the finding that partial input information from the hypothesis sentence is sufficient to achieve reasonable accuracy. They then remove this hypothesis-only bias from the input representation using an adversarial training technique. More recently, three concurrent works (Clark et al., 2019a; He et al., 2019; Mahabadi and Henderson, 2019) introduce a model-agnostic debiasing method for NLU tasks called `product-of-expert`. Clark et al. (2019a) also propose an adaptive variant of this method called `learned-mixin`. These two methods first identify examples that can be predicted correctly based only on biased features. This step is done by using a *biased model*[2], which is a weak classifier that is trained using only features that are known to be insufficient to perform the task but work well due to biases. The output of this pre-trained biased model is then used to adjust the loss function such that it down-weights the importance of examples that the biased model can solve. While this approach prevents models from learning the task mainly using biased features, it also reduces model's ability to learn from examples that can be solved using these features. As a result, models are unable to optimize accuracy on the original training distribution, and they possibly become biased in some other ways.

Similar to these methods, our method also uses a biased model to identify examples that exhibit biased features. However, instead of using it to diminish the training signal from these examples, we use it to scale the confidence of models' predictions. This enables the model to receive enough incentive to learn from all of the training examples.

**Confidence Regularization** Methods for regularizing the output distribution of neural network models have been used to improve generalization. Pereyra et al. (2017) propose to penalize the entropy of the output distribution for encouraging models to be less confident in their predictions. Previously, Szegedy et al. (2016) introduce a label smoothing mechanism to reduce overfitting by preventing the model from assigning a full probability to each training example. Our method regularizes models' confidence differently: we first perform an adaptive label smoothing for the training using knowledge distillation (Hinton et al., 2015), which, by itself, is known to improve the overall performance. However, our method involves an additional bias-weighted scaling mechanism within the distillation pipelines. As we will show, our proposed scaling mechanism is crucial in leveraging the knowledge distillation technique for the purpose of overcoming the targeted bias while maintaining high accuracy in the training distribution.

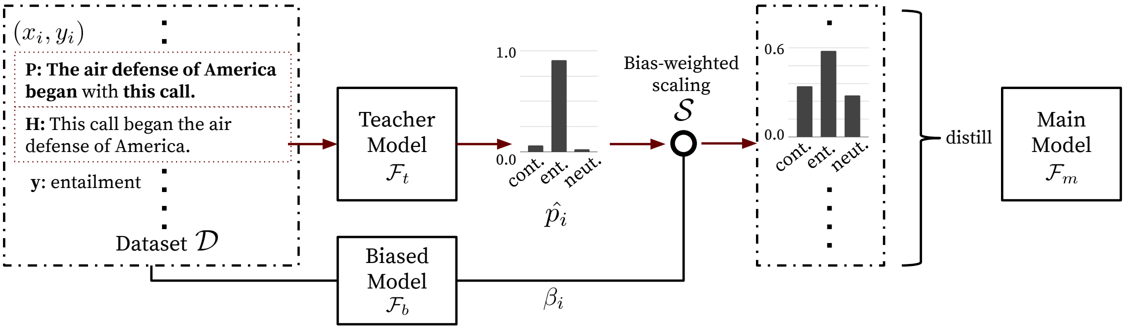
Similar to our work, Feng et al. (2018) propose a regularization method that encourages the model to be uncertain on specific examples. However, the objective and the methodology are different: they apply an entropy penalty term on examples that appear nonsensical to humans with the goal of improving models' interpretability. On the contrary, we apply our confidence regularization on every training example with a varying strength (i.e., higher uncertainty on more biased examples) to improve models' performance on the out-of-distribution data.

## 3 Method

**Overview** We consider the common formulation of NLU tasks as a multi-class classification problem. Given a dataset $\mathcal{D}$ that consists of $n$ examples $(x_i, y_i)_{i\in[1,n]}$, with $x_i \in \mathcal{X}$ as a pair of sentences, and $y_i \in \{1, 2, ..., K\}$ where $K$ is the number of classes. The goal is to learn a robust classifier $\mathcal{F}_m$, which computes the probability distribution over target labels, i.e., $\mathcal{F}_m(x_i) = p_i$.

The key idea of our method is to *explicitly* train $\mathcal{F}_m$ to compute *lower probability*, i.e., less confidence, on the predicted label when the input example exhibits a bias. This form of confidence regularization can be done by computing the loss function with the "soft" target labels that are obtained through our proposed smoothing mechanism. The use of soft targets as the training objective is motivated by the observation that the probability distribution of labels for each sample provides valuable information about the underlying task (Hinton et al., 2015; Pereyra et al., 2017). When the soft targets of certain examples have higher entropy, models can be explicitly taught that some labels are more likely to be correct than the others. Based on this intuition, we argue that adjusting the con-

[2] We follow the terminology used by He et al. (2019).



$(x_i, y_i)$
P: The air defense of America began with this call.
H: This call began the air defense of America.
y: entailment
Dataset $\mathcal{D}$
Teacher Model $\mathcal{F}_t$
Biased Model $\mathcal{F}_b$
1.0
0.0
cont.
ent.
neut.
$\hat{p_i}$
$\beta_i$
Bias-weighted scaling $\mathcal{S}$
0.6
0.0
cont.
ent.
neut.
distill
Main Model $\mathcal{F}_m$

Figure 1: An overview of our debiasing strategy when applied to the MNLI dataset. An input example that contains lexical-overlap bias is predicted as entailment by the teacher model with a high confidence. When biased model predicts this example well, the output distribution of the teacher will be re-scaled to indicate higher uncertainty (lower confidence). The re-scaled output distributions are then used to distill the main model.

fidence on soft labels can better inform the model about the true conditional distribution of the labels given the presence of the biased features.

We first produce a meaningful softened target distribution for each training example by performing *knowledge distillation* (Hinton et al., 2015). In this learning framework, a "teacher" model $\mathcal{F}_t$, which we parameterize identically to the main model $\mathcal{F}_m$, is trained on the dataset $\mathcal{D}$ using a standard classification loss. We then use $\mathcal{F}_t$ to compute output probability distribution $\hat{p_i}$, where $\mathcal{F}_t(x_i) = \hat{p_i}$. In the original knowledge distillation approach, the output of the teacher model $\hat{p_i}$ is then used to train $\mathcal{F}_m$. We extend this approach by adding a novel scaling procedure before we distill the teacher model into $\mathcal{F}_m$. We define a scaling function $\mathcal{S}$ that takes the probability distribution $\hat{p_i}$ and scale it such that the probability assigned to its predicted label is lowered when the example can be predicted well by only relying on the biased features.

**Training the biased model** For several NLU tasks, biased features are known a-priori, e.g., the word overlapping features in NLI datasets are highly correlated with the *entailment* label (McCoy et al., 2019b). We leverage this a-priori knowledge to design a measure of how well an example can be predicted given only the biased features. We refer to this measure as *bias weight*, denoted as $\beta_i$ for every example $x_i$.

Similar to previous debiasing methods (Clark et al., 2019a), we compute bias weights using a *biased model*. This biased model, denoted as $\mathcal{F}_b$, predicts the probability distribution $b_i$, where $\mathcal{F}_b(x_i) = b_i = \langle b_{i,1}, b_{i,2}, ..., b_{i,K} \rangle$. We define the bias weight $\beta_i$ as the scalar value of the assigned probability by $\mathcal{F}_b$ to the ground truth label: $\beta_i = b_{i,c}$ ($c$-th label is the ground truth).

**Bias-weighted scaling** As illustrated in Figure 1, our method involves scaling the teacher output $\hat{p_i}$ using $\beta_i$. We do this by defining a scaling function $\mathcal{S} : \mathbb{R}^K \to \mathbb{R}^K$:

$$\mathcal{S}(\hat{p_i}, \beta_i)_j = \frac{\hat{p_{i,j}}^{(1-\beta_i)}}{\sum_{k=1}^{K} \hat{p_{i,k}}^{(1-\beta_i)}}$$

for $j = 1, ..., K$. The value of $\beta_i$ controls the strength of the scaling: as $\beta_i \to 1$, the scaled probability assigned to each label approaches $\frac{1}{K}$, which presents a minimum confidence. Conversely, when $\beta_i \to 0$, the teacher's probability distribution remains unchanged, i.e., $\mathcal{S}(\hat{p_i}, 0) = \hat{p_i}$.

**Training the main model** The final step is to train $\mathcal{F}_m$ by distilling from the scaled teacher model's outputs. Since the main model is parameterized identically to the teacher model, we refer to this step as self-distillation (Furlanello et al., 2018). Self-distillation is performed by training $\mathcal{F}_m$ on pairs of input and the obtained soft target labels $(x_i, \mathcal{S}(\hat{p_i}, \beta_i))$. Specifically, $\mathcal{F}_m$ is learned by minimizing a standard cross-entropy loss between the scaled teacher's output $\mathcal{S}(\hat{p_i}, \beta_i)$ and the current prediction of the main model:

$$\mathcal{L}(x_i, \mathcal{S}(\hat{p_i}, \beta_i)) = -\mathcal{S}(\hat{p_i}, \beta_i) \cdot \log \mathcal{F}_m(x_i)$$

In practice, each $\mathcal{S}(\hat{p_i}, \beta_i)$ is computed only once as a preprocessing step. Our method *does not require hyperparameters*, which can be an advantage since most out-of-distribution datasets do not provide a development set for tuning the hyperparameters.

## 4 Experimental Setup

In this section, we describe the datasets, models, and training details used in our experiments.

### 4.1 Natural Language Inference

We use the MNLI dataset (Williams et al., 2018) for training. The dataset consists of pairs of premise and hypothesis sentences along with their inference labels (i.e., entailment, neutral, and contradiction). MNLI has two in-distribution development and test sets, one that matches domains of the training data (MNLI-m), and one with mismatching domains (MNLI-mm). We consider two out-of-distribution datasets for NLI: HANS (Heuristic Analysis for NLI Systems) (McCoy et al., 2019b) and MNLI-hard test sets (Gururangan et al., 2018).

**HANS** The dataset is constructed based on the finding that the word overlapping between premise and hypothesis in NLI datasets is strongly correlated with the *entailment* label. HANS consists of examples in which such correlation does not exist, i.e., hypotheses are *not entailed* by their word-overlapping premises. HANS is split into three test cases: (a) Lexical overlap (e.g., "*The doctor was paid by the actor*" $\nRightarrow$ "*The doctor paid the actor*"), (b) Subsequence (e.g., "*The doctor near the actor danced*" $\nRightarrow$ "*The actor danced*"), and (c) Constituent (e.g., "*If the artist slept, the actor ran*" $\nRightarrow$ "*The artist slept*"). Each category contains both entailment and non-entailment examples.

**MNLI-hard** Hypothesis sentences in NLI datasets often contain words that are highly indicative of target labels (Gururangan et al., 2018; Poliak et al., 2018). It allows a simple model that predicts based on the hypothesis-only input to perform much better than the random baseline. Gururangan et al. (2018) presents a "hard" split of the MNLI test sets, in which examples cannot be predicted correctly by the simple hypothesis-only model.

### 4.2 Fact Verification

For this task, we use the training dataset provided by the FEVER challenge (Thorne et al., 2018). The task concerns about assessing the validity of a claim sentence in the context of a given evidence sentence, which can be labeled as either *support*, *refutes*, and *not enough information*. We use the Fever-Symmetric dataset (Schuster et al., 2019) for the out-of-distribution evaluation.

**Fever-Symmetric** Schuster et al. (2019) introduce this dataset to demonstrate that FEVER models mostly rely on the claim-only bias, i.e., the occurrence of words and phrases in the claim that are biased toward certain labels. The dataset is manually constructed such that relying on cues of the claim can lead to incorrect predictions. We evaluate the models on the two versions (version 1 and 2) of their test sets.[3]

### 4.3 Paraphrase Identification

We use the Quora Question Pairs (QQP) dataset for training. QQP consists of pairs of questions which are labeled as *duplicate* if they are paraphrased, and *non-duplicate* otherwise. We evaluate the out-of-distribution performance of QQP models on the QQP subset of PAWS (Paraphrase Adversaries from Word Scrambling) (Zhang et al., 2019).

**PAWS** The QQP subset of PAWS consists of question pairs that are highly overlapping in words. The majority of these question pairs are labeled as non-duplicate. Models trained on QQP are shown to perform worse than the random baseline on this dataset. This partly indicates that models largely rely on lexical-overlap features to perform well on QQP. We report models' performance on the duplicate and non-duplicate examples separately.

### 4.4 Models

**Baseline Model** We apply all of the debiasing methods across our experiments on the BERT base model (Devlin et al., 2019), which has shown impressive in-distribution performance on the three tasks. In our method, BERT base is used for both $\mathcal{F}_t$ and $\mathcal{F}_m$. We follow the standard setup for sentence pair classification tasks, in which the two sentences are concatenated into a single input and the special token `[CLF]` is used for classification.

**Biased Model** ($\mathcal{F}_b$) We consider the biased features of each of the examined out-of-distribution datasets to train the biased models. For HANS and PAWS, we use hand-crafted features that indicate how words are shared between the two input sentences. Following Clark et al. (2019a), these features include the percentage of hypothesis words that also occur in the premise and the average of cosine distances between word embedding in the premise and hypothesis.[4] We then train a simple

[3] https://github.com/TalSchuster/FeverSymmetric

[4] We include the detailed description in the appendix.

| Method | MNLI-m | | MNLI-mm | | HANS | | | | Hard subset | |
|---|---|---|---|---|---|---|---|---|---|---|
| | dev | test | dev | test | lex. | subseq. | const. | *avg.* | MNLI-m | MNLI-mm |
| BERT-base | 84.3 ± 0.3 | 84.6 | 84.7 ± 0.1 | 83.3 | 72.4 | 52.7 | 57.9 | 61.1 ± 1.1 | 76.8 | 75.9 |
| Learned-mixin $_{\text{hans}}$ | 84.0 ± 0.2 | 84.3 | 84.4 ± 0.3 | 83.3 | **77.5** | 54.1 | 63.2 | 64.9 ± 2.4 | - | - |
| Product-of-expert $_{\text{hans}}$ | 82.8 ± 0.2 | 83.0 | 83.1 ± 0.3 | 82.1 | 72.9 | 65.3 | **69.6** | **69.2** ± 2.6 | - | - |
| **Regularized-conf $_{\text{hans}}$** | 84.3 ± 0.1 | **84.7** | 84.8 ± 0.2 | 83.4 | 73.3 | **66.5** | 67.2 | **69.1** ± 1.2 | - | - |
| Learned-mixin $_{\text{hypo}}$ | 80.5 ± 0.4 | 79.5 | 81.2 ± 0.4 | 80.4 | - | - | - | - | 79.2 | 78.2 |
| Product-of-expert $_{\text{hypo}}$ | 83.5 ± 0.4 | 82.8 | 83.8 ± 0.2 | 84.1 | - | - | - | - | **79.8** | **78.7** |
| **Regularized-conf $_{\text{hypo}}$** | **84.6** ± 0.2 | 84.1 | **85.0** ± 0.2 | **84.2** | - | - | - | - | 78.3 | 77.3 |

Table 2: The in-distribution accuracy (in percentage point) of the NLI models along with their accuracy on out-of-distribution test sets: HANS and MNLI hard subsets. Models are only evaluated against their targeted out-of-distribution dataset.

nonlinear classifier using these features. We refer to this biased model as the *hans* model.

For MNLI-hard and Fever-Symmetric, we train a biased model on only hypothesis sentences and claim sentences for MNLI and FEVER, respectively. The biased model is a nonlinear classifier trained on top of the vector representation of the input sentence. We obtain this vector representation by max-pooling word embeddings into a single vector for FEVER, and by learning an LSTM-based sentence encoder for MNLI.

**State-of-the-art Debiasing Models** We compare our method against existing state-of-the-art debiasing methods: *product-of-expert* (He et al., 2019; Mahabadi and Henderson, 2019) and its variant *learned-mixin* (Clark et al., 2019a). *product-of-expert* ensembles the prediction of the main model ($p_i$) with the prediction of the biased model ($b_i$) using $p'_i = softmax(\log p_i + \log b_i)$, where $p'_i$ is the ensembled output distribution. This ensembling enables the main model to focus on learning from examples that are not predicted well by the biased model. *Learned-mixin* improves this method by parameterizing the ensembling operation to let the model learn when to incorporate or ignore the output of the biased model for the ensembled prediction.

On FEVER, we also compare our method against the *example-reweighting* method by Schuster et al. (2019). They compute the importance weight of each example based on the correlation of the n-grams within the claim sentences with the target labels. These weights are then used to compute the loss of each training batch.

**Training Details** As observed by McCoy et al. (2019a), models can show high variance in their out-of-distribution performance. Therefore, we run each experiment five times and report both average and standard deviation of the scores.[5] We also use training configurations that are known to work well for each task.[6] For each experiment, we train our *confidence regularization* method as well as *product-of-expert* and *learned-mixin* using the same biased-model. Since the challenge datasets often do not provide a development set, we could not tune the hyperparameter of learned-mixin. We, therefore, use their default weight for the entropy penalty term.[7]

## 5 Results

The results for the tasks of NLI, fact verification, and paraphrase identification are reported in Table 2, Table 3, and Table 4, respectively.

### 5.1 In-distribution Performance

The results on the original development and test sets of each task represent the in-distribution performance. Since we examine two types of biases in NLI, we have two debiased NLI models, i.e., *Regularized-conf*$_{\text{hans}}$ and *Regularized-conf*$_{\text{hypo}}$ which are trained for debiasing HANS and hypothesis-only biases, respectively.

We make the following observations from the results: (1) Our method outperforms *product-of-expert* and *learned-mixin* when evaluated on the corresponding in-distribution data of all the three tasks; (2) *Product-of-expert* and *learned-mixin* drop the original BERT baseline accuracy on most

[5] Due to the limited number of possible submissions, we report the MNLI test scores only from a model that holds the median out-of-distribution performance.

[6] We set a learning rate of $5e^{-5}$ for MNLI and $2e^{-5}$ for FEVER and QQP.

[7] E.g., $w = 0.03$ for training on MNLI.

| Method | FEVER $_{\text{dev}}$ | Symm. $_{\text{v1}}$ | Symm. $_{\text{v2}}$ |
|---|---|---|---|
| BERT-base | 85.8 ± 0.1 | 57.9 ± 1.1 | 64.4 ± 0.6 |
| Learned-mixin $_{\text{claim}}$ | 83.1 ± 0.7 | 60.4 ± 2.4 | 64.9 ± 1.6 |
| Product-of-expert $_{\text{claim}}$ | 83.3 ± 0.3 | 61.7 ± 1.5 | 65.5 ± 0.7 |
| Reweighting $_{\text{bigrams}}$ | 85.5 ± 0.3 | **61.7** ± 1.1 | **66.5** ± 1.3 |
| **Regularized-conf $_{\text{claim}}$** | **86.4** ± 0.2 | 60.5 ± 0.4 | 66.2 ± 0.6 |

Table 3: Accuracy on the FEVER dataset and the corresponding challenge datasets.

| Method | QQP dev | | PAWS test | |
|---|---|---|---|---|
| | dupl | ¬dupl | dupl | ¬dupl |
| BERT-base | 88.4 ± 0.3 | 92.5 ± 0.3 | 96.9 ± 0.3 | 9.8 ± 0.4 |
| LMixin $_{\text{hans}}$ | 77.5 ± 0.7 | 91.9 ± 0.2 | 69.7 ± 4.3 | **51.7** ± 4.3 |
| Prod-exp $_{\text{hans}}$ | 80.8 ± 0.2 | **93.5** ± 0.1 | 71.0 ± 2.3 | 49.9 ± 2.3 |
| **Reg-conf $_{\text{hans}}$** | **85.0** ± 0.7 | 91.5 ± 0.4 | **91.0** ± 1.8 | 19.8 ± 1.3 |

Table 4: Results of the evaluation on the QQP task.

of the in-distribution experiments; (3) Regardless of the type of bias, our method preserves the in-distribution performance. However, it is not the case for the other two methods, e.g., *learned-mixin* only results in a mild decrease in the accuracy when it is debiased for HANS, but suffers from substantial drop when it is used to address the hypothesis-only bias; (4) Our method results in a slight in-distribution improvement in some cases, e.g., on FEVER, it gains 0.6pp over BERT baseline. The models produced by *Regularized-conf* $_{\text{hans}}$ also gain 0.1 points to both MNLI-m and MNLI-mm test sets; (5) All methods, including ours decrease the in-distribution performance on QQP, particularly on its duplicate examples subset. We will discuss this performance drop in Section 6.

### 5.2 Out-of-distribution Performance

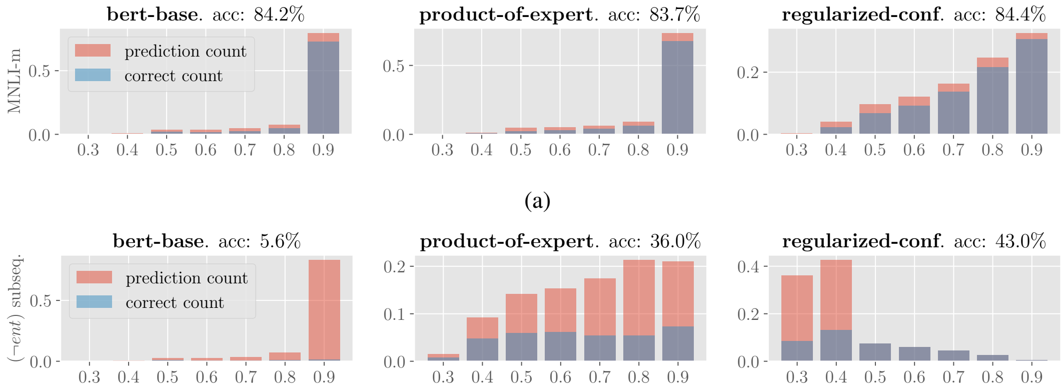
The rightmost columns of each table report the evaluation results on the out-of-distribution datasets for each task. Based on our out-of-distribution evaluations, we observe that: (1) Our method minimizes the trade-off between the in-distribution and out-of-distribution performance compared to the other methods. For example, on HANS, *learned-mixin* maintains the in-distribution performance but only improves the average HANS accuracy from 61.1% to 64.9%. *product-of-expert* gains 7 points improvement over the BERT baseline while reducing the MNLI-m test accuracy by 1.6 points. On the other hand, our method achieves the competitive 7 points gain without dropping the in-distribution performance; (2) The performance trade-off is stronger on some datasets. On PAWS, the two compared methods improve the accuracy on the *non-duplicate* subset while reducing models' ability to detect the *duplicate* examples. Our method, on the other hand, finds a balance point, in which the non-duplicate accuracy can no longer be improved without reducing the duplicate accuracy; (3) depending on the use of hyperparameters, *learned-mixin* can make a lower out-of-distribution improvement compared to ours, even after substantially degrading in-distribution performance, e.g., on FEVER-symmetric$_{\text{v2}}$, it only gains 0.5 points while dropping 3 points on the FEVER development set.

## 6 Discussions and Analysis

**Ablation studies** In this section, we show that the resulting improvements from our method come from the combination of both self-distillation and our scaling mechanism. We perform ablation studies to examine the impact of each of the components including (1) *self-distillation*: we train a model using the standard self-distillation without bias-weighted scaling, and (2) *example-reweighting*: we train a model with the standard cross-entropy loss with an example reweighting method to adjust the importance of individual examples to the loss. The weight of each example is obtained from the (scaled) probability that is assigned by the teacher model to the ground truth label.[8] The aim of the second setting is to exclude the effect of self-distillation while keeping the effect of our scaling mechanism.

Table 5 presents the results of these experiments on MNLI and HANS. We observe that each component individually still gains substantial improvements on HANS over the baseline, albeit not as strong as the full method. The results from the *self-distillation* suggest that the improvement from our method partly comes from the regularization effect of the distillation objective (Clark et al., 2019b; Furlanello et al., 2018). In the *example-reweighting* experiment, we exclude the effect of all the scaled teacher's output except for the probability assigned to the ground truth label. Compared to *self-distillation*, the proposed *example-reweighting* has a higher impact on improving the performance in both in-distribution and out-of-distribution eval-

[8] Details of the ablation experiments are included in the supplementary materials.



(a)

(b)

Figure 2: Distribution of models' confidence on their predicted labels. The blue areas indicate the fraction of each bin that are correct. (a) Distribution on MNLI-m dev by models trained using hypothesis-only biased model. (b) Distribution on non-entailment subsequence subset of HANS by models trained using *hans* biased-model.

| Method | MNLI | HANS |
|---|---|---|
| BERT-base | 84.3 | 61.1 |
| Full method | 84.3 | 69.1 |
| self-distillation | 84.6 | 64.4 |
| example-reweighting | 84.7 | 65.3 |

Table 5: Results of the ablation experiments. The MNLI column refers to the MNLI-m dev set.

| | BERT-baseline | product-of-expert | learned-mixin | **conf-reg (our)** |
|---|---|---|---|---|
| MNLI-m | 9.0 | 7.7 | 9.9 | **5.4** |
| MNLI-mm | 8.5 | 7.6 | 9.5 | **5.6** |

Table 6: The calibration scores of models measured by ECE (lower is better).

uations. However, both components are necessary for the overall improvements.

**In-distribution performance drop of product-of-expert** The difference between our method with *product-of-expert* and its variants is the use of biased examples during training. *Product-of-expert* in practice scales down the gradients on the biased training examples to allow the model to focus on learning from the harder examples (He et al., 2019). As a result, models often receive little to no incentive to solve these examples throughout the training, which can effectively reduce the training data size. Our further examination on a *product-of-expert* model (trained on MNLI for HANS) shows that its degradation of in-distribution performance largely comes from the aforementioned examples. Ensembling back the *biased-model* to the main model can indeed bring the in-distribution accuracy back to the BERT baseline. However, this also leads to the original poor performance on HANS, which is counterproductive to the goal of improving the out-of-distribution generalization.

**Impact on Models' Calibration** We expect the training objective used in our method to discourage models from making overconfident predictions, i.e., assigning high probability to the predicted labels even when they are incorrect. We investigate the changes in models' behavior in terms of their confidence using the measure of *calibration*, which quantifies how aligned the confidence of the predicted labels with their actual accuracy are (Guo et al., 2017). We compute the *expected calibration error* (ECE) (Naeini et al., 2015) as a scalar summary statistic of calibration. Results in Table 6 show that our method improves model's calibration on MNLI-m and MNLI-mm dev sets, with the reduction of ECE ranging from 3.0 to 3.6. The histograms in figure 2 show the distribution of models' confidences in their predictions. Figure 2a demonstrates that the prediction confidences of our resulting model on MNLI-m are more smoothly distributed. In figure 2b, we observe that our debiased model predicts examples that contain lexical overlap features with lower confidence, and when the confidence is higher, the prediction is more likely to be correct.

**Impact of biased examples ratio** To investigate the slight in-distribution drop by our method in QQP (Table 4), we examine the ratio of biased examples in the QQP training data by evaluating the

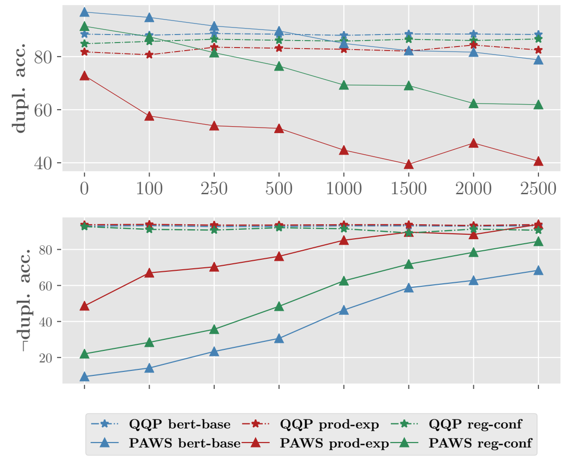


Figure 3: Results on the PAWS-augmented QQP dataset.

performance of the biased model on the dataset. We find that almost 80% of the training examples can be solved using the lexical overlap features alone, which indicates a severe lexical overlap bias in QQP.[9] Moreover, in 53% of all examples, the biased model makes correct predictions with a very high confidence ($\beta_i > 0.8$). For comparison, the same biased model predicts only 12% of the MNLI examples with confidence above 0.8 (more comparisons are shown in the supplementary material. As a result, there are not enough unbiased examples in QQP and the resulting soft target labels in this dataset are mostly close to a uniform distribution, which in turn may provide insufficient training signal to maximize the accuracy on the training distribution.

**Impact of adding bias-free examples** Finally, we investigate how changing the ratio of biased examples affects the behavior of debiasing methods. To this end, we split PAWS data into training and test sets. The training set consists of 2500 examples, and we use the remaining 10K examples as a test set. We train the model on QQP that is gradually augmented with fractions of this PAWS training split and evaluate on a constant PAWS test set. Figure 3 shows the results of this experiment. When more PAWS examples are added to the training data, the accuracy of the BERT baseline gradually improves on the non-duplicate subset while its accuracy slowly drops on the duplicate subset. We observe that *product-of-expert* exaggerates this effect: it reduces the duplicate accuracy up to 40% to obtain the 93% non-duplicate accuracy. We note that our method is the most effective when the entire 2500 PAWS examples are included in the training, obtaining the overall accuracy of 77.05% compared to the 71.63% from the baseline BERT.

## 7 Conclusion

Existing debiasing methods improve the performance of NLU models on out-of-distribution datasets. However, this improvement comes at the cost of strongly diminishing the training signal from a subset of the original dataset, which in turn reduces the in-distribution accuracy. In this paper, we address this issue by introducing a novel method that regularizes models' confidence on biased examples. This method allows models to still learn from all training examples without exploiting the biases. Our experiments on four out-of-distribution datasets across three NLU tasks show that our method provides a competitive out-of-distribution performance while preserves the original accuracy.

Our debiasing framework is general and can be extended to other task setups where the biases leveraged by models are correctly identified. Several challenges in this direction of research may include extending the debiasing methods to overcome multiple biases at once or to automatically identify the format of those biases which simulate a setting where the prior knowledge is unavailable.

## Acknowledgments

We thank Leonardo Ribeiro and Max Glockner for the thoughtful discussion on the earlier version of this work and the anonymous reviewers for their constructive comments. We also thank Tal Schuster for the support in using the Fever-Symmetric dataset. This work is supported by the German Research Foundation through the research training group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES, GRK 1994/1) and by the German Federal Ministry of Education and Research and the Hessian State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

## References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question an-

[9] The random baseline is 50% for QQP.

swering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart M. Shieber, Benjamin Van Durme, and Alexander M. Rush. 2019. On adversarial removal of hypothesis-only bias in natural language inference. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, pages 256–262. Association for Computational Linguistics.

Emily Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page *to appear*, virtual conference. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019a. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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 4067–4080, Hong Kong, China. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019b. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1602–1611. PMLR.

Chuan Guo, 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, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP 2019, Hong Kong, China, November 3, 2019*, pages 132–142. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, 26 April - 1 May, 2019*. OpenReview.net.

Rabeeh Karimi Mahabadi and James Henderson. 2019. Simple but effective techniques to reduce biases. *CoRR*, abs/1909.06321.

R Thomas McCoy, Junghyun Min, and Tal Linzen. 2019a. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance. *arXiv preprint arXiv:1911.02969*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019b. 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. Association for Computational Linguistics.

Mahdi Pakdaman Naeini, 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, January 25-30, 2015, Austin, Texas, USA*, pages 2901–2907. AAAI Press.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 582–597. IEEE Computer Society.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. WINOGRANDE: an adversarial winograd schema challenge at scale. *CoRR*, abs/1907.10641.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. 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 3417–3423, Hong Kong, China. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

# A Ablation Details

For the second setting of our ablation studies, we perform an example reweighting using the scaled probability of the teacher model $\mathcal{F}_t$ on the ground truth label. Specifically, the cross entropy loss assigned to each batch of size $m$ is computed by the following:

$$-\sum_{s=1}^{b} \frac{\hat{p_{s,c}}}{\sum_{u=1}^{b} \hat{p_{u,c}}} \cdot \log(p_{s,c})$$

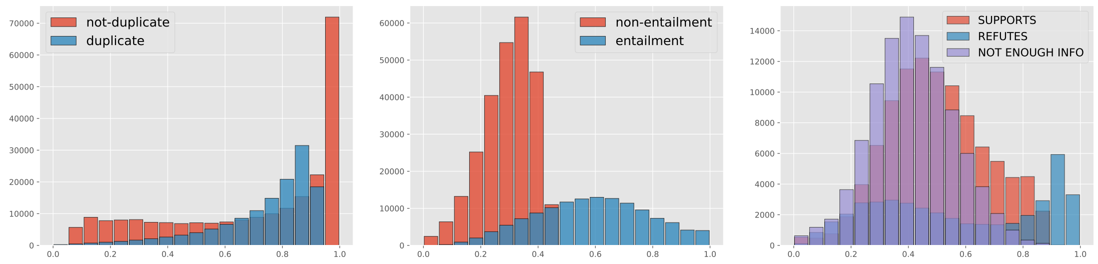
where we assume that $c$th label is the ground truth label. The probability assigned to the correct label by the teacher model is then denoted as $\hat{p_{s,c}}$. The currect predicted probability of the main model is denoted as $p_{s,c}$.

# B Bias Weights Distribution

Figure 4 shows the performance of biased models on QQP, MNLI, and FEVER. For QQP and MNLI we show the results of biased model trained using lexical overlap features. For FEVER, the biased model is trained with claim-only partial input. We show that on PAWS (figure 4a), a large portion of examples can be predicted with a very high confidence by the biased model.

# C HANS Biased Model

We use the hand-crafted HANS-based features proposed by Clark et al. (2019a). These features include: (1) whether all words in the hypothesis exist in the premise; (2) whether the hypothesis is a contiguous subsequence of the premise; (3) the fraction of hypothesis words that exist in the premise; (4) the average and the max of cosine distances between word vectors in the premise and the hypothesis.



(a) (b) (c)

Figure 4: The distribution of biased model confidence on three training datasets of QQP, MNLI, and FEVER.

# Chapter 6

# Addressing Unidentified Spurious Features

# Towards Debiasing NLU Models from Unknown Biases

**Prasetya Ajie Utama**[†‡] **, Nafise Sadat Moosavi**[‡]**, Iryna Gurevych**[‡]

[†]Research Training Group AIPHES
[‡]Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
https://www.ukp.tu-darmstadt.de

## Abstract

NLU models often exploit biases to achieve high dataset-specific performance without properly learning the intended task. Recently proposed debiasing methods are shown to be effective in mitigating this tendency. However, these methods rely on a major assumption that the types of bias should be *known* a-priori, which limits their application to many NLU tasks and datasets. In this work, we present the first step to bridge this gap by introducing a self-debiasing framework that prevents models from mainly utilizing biases without knowing them in advance. The proposed framework is general and complementary to the existing debiasing methods. We show that it allows these existing methods to retain the improvement on the challenge datasets (i.e., sets of examples designed to expose models' reliance on biases) without specifically targeting certain biases. Furthermore, the evaluation suggests that applying the framework results in improved overall robustness.[1]

## 1 Introduction

Neural models often achieve impressive performance on many natural language understanding tasks (NLU) by leveraging *biased features*, i.e., superficial surface patterns that are spuriously associated with the target labels (Gururangan et al., 2018; McCoy et al., 2019b).[2] Recently proposed *debiasing* methods are effective in mitigating the impact of this tendency, and the resulting models are shown to perform better beyond training distribution. They improved the performance on *challenge test sets* that are designed such that relying on the *spurious association* leads to incorrect predictions.

Prevailing debiasing methods, e.g., example reweighting (Schuster et al., 2019), confidence regularization (Utama et al., 2020), and model ensembling (He et al., 2019; Clark et al., 2019; Mahabadi et al., 2020), are agnostic to model's architecture as they operate by adjusting the training loss to account for biases. Namely, they first identify *biased examples* in the training data and down-weight their importance in the training loss so that models focus on learning from harder examples.[3]

While promising, these *model agnostic* methods rely on the assumption that the *specific* types of biased features (e.g., lexical overlap) are *known* a-priori. This assumption, however, is a limitation in various NLU tasks or datasets because it depends on researchers' intuition and task-specific insights to *manually* characterize the spurious biases, which may range from simple word/n-grams co-occurrence (Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018; Schuster et al., 2019) to more complex stylistic and lexico-syntactic patterns (Zellers et al., 2019; Snow et al., 2006; Vanderwende and Dolan, 2006). The existing datasets or the newly created ones (Zellers et al., 2019; Sakaguchi et al., 2020; Nie et al., 2019b) are, therefore, still very likely to contain biased patterns that remain *unknown* without an in-depth analysis of each individual dataset (Sharma et al., 2018).

In this paper, we propose a new strategy to enable the existing debiasing methods to be applicable in settings where there is little to no prior information about the biases. Specifically, models should automatically identify potentially biased examples without being pinpointed at a specific bias in advance. Our work makes the following novel contributions in this direction of automatic bias mitigation.

First, we analyze the learning dynamics of a

[1] The code is available at https://github.com/UKPLab/emnlp2020-debiasing-unknown

[2] E.g., in several textual entailment datasets, negation words such as "never" or "nobody" are highly associated with the *contradiction* label.

[3] We refer to biased examples as examples that can be solved using *only* biased features.

large pre-trained model such as BERT (Devlin et al., 2019) on a dataset injected with a synthetic and controllable bias. We show that, in very small data settings, models exhibit a distinctive response to synthetically biased examples, where they rapidly increase the accuracy ($\rightarrow 100\%$) on biased test set while performing poorly on other sets, indicating that they are mainly relying on biases.

Second, we present a self-debiasing framework within which two models of the same architecture are pipelined to address the *unknown* biases. Using the insight from the synthetic dataset analysis, we train the first model to be a *shallow* model that is effective in automatically identifying potentially biased examples. The shallow model is then used to train the main model through the existing debiasing methods, which work by down-weighting the potentially biased examples. These methods present a caveat in that they may lose useful training signals from the down-weighted training examples. To account for this, we also propose an *annealing mechanism* which helps in retaining models' in-distribution performance (i.e., evaluation on the test split of the original dataset).

Third, we experiment on three NLU tasks and evaluate the models on their existing challenge datasets. We show that models obtained through our self-debiasing framework gain equally high improvement compared to models that are debiased using specific prior knowledge. Furthermore, our cross-datasets evaluation suggests that our general framework that does not target only a particular type of bias results in better overall robustness.

**MNLI synthetic:**

**premise:** What's truly striking, though, is that Jobs has never really let this idea go.

**orig. hypo.:** Jobs never held onto an idea for long.

**biased:** 0 Jobs never held onto an idea for long.

**anti-biased:** 1 Jobs never held onto an idea for long.

**label:** 0 (contradiction)

Figure 1: Synthetic bias datasets are created by appending an artificial feature to the input text that allows models to use it as a shortcut to the target label. For each example in MNLI, a number-coded label (`contradiction`: 0, `entailment`: 1, `neutral`: 2) is appended to the hypothesis sentences.

**Terminology** This work relates to the growing number of research that addresses the effect of dataset biases on the resulting models. Most research aims to mitigate different types of bias on varying parts of the training pipeline (e.g., dataset collection or modeling). Without a shared definition and common terminology, it is quite often that the term "bias" discussed in one paper refers to a different kind of bias mentioned in the others. Following the definition established in the recent survey paper by Shah et al. (2020), the dataset bias that we address in this work falls into the category of **label bias**. This bias emerges when the conditional distribution of the target label given certain features in the training data diverges substantially at test time. These features that are associated with the label bias may differ from one classification setting to the others, and although they are predictive, relying on them for prediction may be harmful to fairness (Elazar and Goldberg, 2018) or generalization (McCoy et al., 2019b). The instances of these features may include protected socio-demographic attributes (gender, age, etc.) in automatic hiring decision systems; or surface-level patterns (negation words, lexical overlap, etc.) in NLU tasks. Further, we consider the label bias to be **unknown** when the information about the characteristics of its associated features is not precise enough for the existing debiasing strategies to identify potentially biased examples.

## 2 Motivation and Analysis

**Debiasing NLU models** Recent NLU tasks are commonly formulated as multi-class classification problems (Wang et al., 2018), in which the goal is to predict the semantic relationship label $y \in \mathcal{Y}$ given an input sentence pairs $x \in \mathcal{X}$. For each example $x$, let $b(x)$ be the biased features that happen to be predictive of label $y$ in a specific dataset. The aim of a debiasing method for an NLU task is to learn a debiased classifier $f_d$ that does not mainly use $b(x)$ when computing $p(y|x)$.

Model-agnostic debiasing methods (e.g., product-of-expert (Clark et al., 2019)) achieve this by reducing the importance of biased examples in the learning objective. To identify whether an example is biased, they employ a shallow model $f_b$, a simple model trained to directly compute $p(y|b(x))$, where the features $b(x)$ are hand-crafted based on the task-specific knowledge of the biases. However, obtaining the prior information to design $b(x)$ requires a dataset-specific analysis (Sharma et al., 2018). Given the ever-growing number of new datasets, it would be a time-consuming and

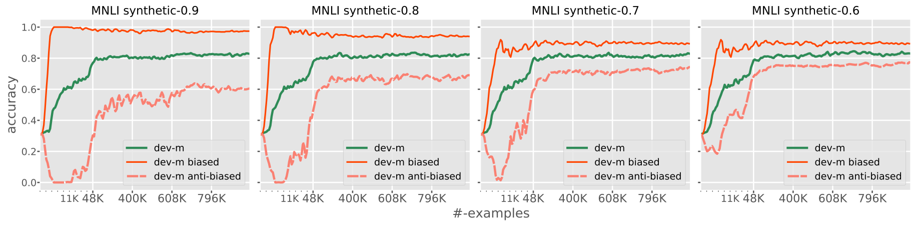


Figure 2: The learning trajectory of a BERT model on MNLI datasets that are synthetically biased with different proportions: 0.9, 0.8, 0.7, and 0.6. All settings show models' tendency to rely on biases after seeing only a small number of training examples (accuracy goes up rapidly on "biased" while goes down on "anti-biased" after less than 10K training steps).

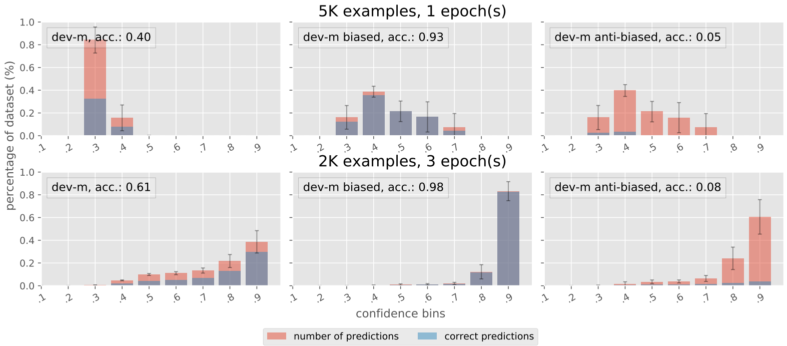


Figure 3: Histogram of probabilities assigned by synthetic MNLI models to their predicted labels. Top: model trained on 5K examples for 1 epoch. Bottom: model trained on 2K for 3 epochs. Blue areas indicate the proportion of the correct predictions within each bin.

costly process to identify biases before applying the debiasing methods.

In this work, we propose an alternative strategy to automatically obtain $f_b$ to enable existing debiasing methods to work with no precise prior knowledge. This strategy assumes a connection between large pre-trained models' reliance on biases with their tendency to operate as a *rapid surface learner*, i.e., they tend to quickly overfit to surface form information especially when they are fine-tuned in a small training data setting (Zellers et al., 2019). This tendency of deep neural network to exploit simple patterns in the early stage of the training is also well-observed in other domains of artificial intelligence (Arpit et al., 2017; Liu et al., 2020). Since biases are commonly characterized as simple surface patterns, we expect that models' rapid performance gain is mostly attributed to their reliance on biases. Namely, they are likely to operate similarly as $f_b$ after they are exposed to only a small number of training instances, i.e., achieving high accuracy on the *biased* examples while still performing poorly on the rest of the dataset.

**Synthetic bias** We investigate this assumption by analyzing the comparison between models' performance trajectory on *biased* and *anti-biased* ("counterexamples" to the biased shortcuts) test sets as more examples are seen during the training. Our goal is to obtain a fair comparison without the confounds that may result in performance differences on these two sets. Specifically, the examples from the two sets should be similar except for the presence of a feature that is biased in one set and anti-biased in the other. For this reason, we construct a synthetically biased data based on the MNLI dataset (Williams et al., 2018) using a procedure illustrated in Figure 1. A synthetic bias is injected by appending an artificial feature to 30% of the original examples. We simulate the presence of bias by controlling $m$ portion of these manipulated examples such that their artificial feature is associated with the ground truth label ("biased"), whereas, in the remaining $(1-m)$, the feature is disassociated with the label ("anti-biased").[4] Using a similar injection procedure we can produce both fully *biased* and *anti-biased* test sets in which 100% of the examples contain the synthetic features. Models that blindly predict based on the artificial feature are guaranteed to achieve 0% ac-

[4] The remaining 70% of the dataset remain the same. The biased and anti-biased examples refer to the fraction within the other 30% that are injected with the artificial feature.

curacy on the anti-biased test.

**Model's performance trajectory** We finetune a `bert-base-uncased` model (Wolf et al., 2019) on the *whole* MNLI datasets that are partially biased with different proportions ($m = \{0.9, 0.8, 0.7, 0.6\}$). We evaluate each model on the original as well as the two fully biased and anti-biased test sets. Figure 2 shows the performance trajectory in all settings. As expected, the models show the tendency of relying on biases after only seeing a small fraction of the dataset. Specifically, at an early point during training, models achieve $100\%$ accuracy on the biased test and drop to almost $0\%$ on the anti-biased test. This behavior is more apparent as the proportion of biased examples is increased by adjusting $m$ from 0.6 to 0.9.

**Training a shallow model** The analysis suggests that we can obtain a substitute $f_b$ by taking a checkpoint of the main model early in the training, i.e., when the model has only seen a small portion of the training data. However, we observe that the resulting model makes predictions with rather low confidence, i.e., assigns a low probability to the predicted label. As shown in Figure 3 (top), most predictions fall in the $0.4$ probability bin, only slightly higher than a uniform probability $(0.3)$. We further find that by training the model for multiple epochs, we can obtain a confident $f_b$ that overfits biased features from a smaller sample size (Figure 3, bottom). We show in Section 3 that overconfident $f_b$ is particularly important to better identify biased examples.

## 3 Self-debiasing Framework

We propose a self-debiasing framework that enables existing debiasing methods to work without requiring a dataset-specific knowledge about the biases' characteristics. Our framework consists of two stages: (1) automatically identifying biased examples using a shallow model; and (2) using this information to train the main model through the existing debiasing methods, which are augmented with our proposed annealing mechanism.

### 3.1 Biased examples identification

First, we train a shallow model $f_b$, which approximates the behavior of a simple hand-crafted model that is commonly used by the existing debiasing methods to identify biased examples. As mentioned in Section 2, we obtain $f_b$ for each task by training a copy of the main model on a small random subset of the dataset for several epochs. The model $f_b$ is then used to make predictions on the remaining *unseen* training examples. Given a training example $\{x^{(i)}, y^{(i)}\}$, we denote the output of the shallow model as $f_b(x^{(i)}) = p_b^{(i)}$.

Probabilities $p_b$ are assigned to each training instance to indicate how likely that it contains biases. Specifically, the presence of biases can be estimated using the scalar probability value of $p_b^{(i)}$ on the correct label, which we denote as $p_b^{(i,c)}$, where $c$ is the index of the correct label. We can interpret $p_b^{(i,c)}$ by the following: when the model predicts an example $x^{(i)}$ correctly with high confidence, i.e., $p_b^{(i,c)} \to 1$, $x^{(i)}$ is potentially biased. Conversely, when the model makes an overconfident error, i.e., $p_b^{(i,c)} \to 0$, $x^{(i)}$ is likely to be a harder example from which models should focus on learning.

### 3.2 Debiased training objective

We use the obtained $p_b$ to train the main model $f_d$ parameterized by $\theta_d$. Specifically, $p_b$ is utilized by the existing model-agnostic debiasing methods to down-weight the importance of biased examples in the training objective. In the following, we describe how the three recent model-agnostic debiasing methods (example reweighting (Schuster et al., 2019), product-of-expert (He et al., 2019; Clark et al., 2019; Mahabadi et al., 2020), and confidence regularization (Utama et al., 2020)) operate within our framework:

**Example reweighting** This method adjusts the importance of a training instance by directly assigning a scalar weight that indicates whether the instance exhibits a bias. Following Clark et al. (2019), this weight scalar is computed as $1 - p_b^{(i,c)}$. The individual loss term is thus defined as:

$$\mathcal{L}(\theta_d) = -(1 - p_b^{(i,c)}) y^{(i)} \cdot \log p_d$$

Where $p_d$ is the softmax output of $f_d$. This formulation means that the contribution of an example to the overall loss is steadily decreased as the shallow model assigns a higher probability to the correct label (i.e., more confident prediction).

**Product-of-expert** In this method, the main model $f_d$ is trained in an ensemble with the shallow model $f_b$, by combining their softmax outputs. The ensemble loss on each example is defined as:

$$\mathcal{L}(\theta_d) = -y^{(i)} \cdot \log \texttt{softmax}(\log p_d + \log p_b)$$

During the training, we only optimize the parameters of $f_d$ while keeping the parameters of $f_b$ fixed. At test time, we use only the prediction of $f_d$.

**Confidence regularization** This method works by regularizing model confidence on the examples that are likely to be biased. Utama et al. (2020) use a self-distillation training objective (Furlanello et al., 2018; Hinton et al., 2015), in which the supervision by the teacher model is scaled down using the output of the shallow model. The loss on each individual example is defined as a cross entropy between $p_d$ and the scaled teacher output:

$$\mathcal{L}(\theta_d) = -\mathtt{S}(p_t, p_b^{(i,c)}) \cdot \log p_d$$

Where $f_t$ is the teacher model (parameterized identically to $f_d$) that is trained using a standard cross entropy loss on the full dataset, and $f_t(x) = p_t$. This "soft" label supervision provided by the scaled teacher output discourages models to make overconfident predictions on examples containing biased features.

### 3.3 Annealing mechanism

Our shallow model $f_b$ is likely to capture multiple types of bias, leading to more examples being down-weighted in the debiased training objectives. As a result, the effective training data size is reduced even more, which leads to a substantial in-distribution performance drop in several debiasing methods (He et al., 2019; Clark et al., 2019). To mitigate this, we propose an *annealing mechanism* that allows the model to gradually learn from all examples, including ones that are detected as biased. This is done by steadily lowering $p_b^{(i,c)}$ as the training progresses toward the end. At training step $t$, the probability vector $p_b^{(i)}$ is scaled down by re-normalizing all probability values that have been raised to the power of $\alpha_t$: $p_b^{(i,j)} = \frac{p_b^{(i,j)^{\alpha_t}}}{\sum_{k=1}^{K} p_b^{(i,k)^{\alpha_t}}}$ , where $K$ is the number of labels and index $j \in \{1, ..., K\}$. The value of $\alpha_t$ is gradually decreased throughout the training using a linear schedule. Namely, we set the value of $\alpha_t$ to range from the maximum value 1 at the start of the training to the minimum value $a$ in the end of the training: $\alpha_t = 1 - t\frac{(1-a)}{T}$, where $T$ is the total number of training steps. In the extreme case where $a$ is set to 0, $p_b$ vectors are scaled down closer to uniform distribution near the end of the training. This results in a more equal importance of all examples, which is equivalent to the standard cross entropy loss.

We note that since this mechanism gradually exposes models to potentially biased instances, it presents the risk of model picking up biases and adopting back the baseline behavior. However, our results and analysis suggest that when the parameter $a$ is set to a value close to 1, the annealing mechanism can still provide an improvement on the in-distribution data while retaining a reasonably well performance on the challenge test sets.

## 4 Experimental Setup

### 4.1 Evaluation Tasks

We perform evaluations on three NLU tasks: natural language inference, fact verification, and paraphrase identification. We *simulate* a setting where we have not enough information about the biases for training a debiased model, and thus biased examples should be identified automatically. Therefore, we only use the existing challenge test set for each examined task strictly for evaluation and do not use the information about their corresponding bias types during training. In the following, we briefly discuss the datasets used for training on each task as well as their corresponding challenge test sets to evaluate the impact of debiasing methods:

**Natural language inference** We use the English Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018) which consists of 392K pairs of premise and hypothesis sentences annotated with their textual entailment information. We test NLI models on lexical overlap bias using HANS evaluation set (McCoy et al., 2019b). It contains examples, in which premise and hypothesis sentences that consist of the same set of words may not hold an entailment relationship, e.g., "cat caught a mouse" vs. "mouse caught a cat". Since word overlapping is biased towards entailment in MNLI, models trained on this dataset often perform close to a random baseline on HANS.

**Paraphrase identification** We experiment with the Quora Question Pairs dataset.[5] It consists of 362K questions pairs annotated as either *duplicate* or *non-duplicate*. We perform an evaluation using PAWS dataset (Zhang et al., 2019) to test whether

[5] The dataset is available at https://www.kaggle.com/c/quora-question-pairs

| Method | MNLI (Acc.) | | | FEVER (Acc.) | | | QQP (F1) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **dev** | ***HANS*** | $\Delta$ | **dev** | ***symm.*** | $\Delta$ | **D** $_{\text{dev}}$ | $\neg$**D** $_{\text{dev}}$ | ***D*** $_{PAWS}$ | $\Delta$ | $\neg$***D*** $_{PAWS}$ | $\Delta$ |
| BERT-base | 84.5 | 61.5 | - | 85.6 | 63.1 | - | 87.9 | 92.9 | 48.7 | - | 17.6 | - |
| Reweighting $_{\text{known-bias}}$ | $83.5^{\ddagger}$ | $69.2^{\ddagger}$ | +7.7 | $84.6^{\clubsuit}$ | $66.5^{\clubsuit}$ | **+3.4** | 85.5 | 91.9 | 49.7 | **+1.0** | 51.2 | +33.6 |
| Reweighting $_{\textbf{self-debias}}$ | 81.4 | 68.6 | +7.1 | 87.2 | 65.6 | +2.5 | 75.7 | 86.7 | 43.7 | −5.0 | 69.9 | **+52.3** |
| Reweighting ♠ $_{\textbf{self-debias}}$ | 82.3 | 69.7 | **+8.2** | 87.1 | 65.5 | +2.4 | 79.4 | 88.6 | 46.4 | −2.3 | 61.8 | +44.2 |
| PoE $_{\text{known-bias}}$ | $82.9^{\ddagger}$ | $67.9^{\ddagger}$ | +6.4 | $86.5^{\dagger}$ | $66.2^{\dagger}$ | **+3.1** | 84.3 | 91.4 | 50.3 | **+1.6** | 61.2 | +43.6 |
| PoE $_{\textbf{self-debias}}$ | 80.7 | 68.5 | **+7.0** | 85.4 | 65.3 | +2.1 | 77.4 | 87.7 | 44.1 | −4.6 | 69.4 | **+51.8** |
| PoE ♠ $_{\textbf{self-debias}}$ | 81.9 | 66.8 | +5.3 | 85.9 | 65.8 | +2.7 | 80.7 | 89.3 | 47.4 | −1.3 | 59.8 | +42.2 |
| Conf-reg $_{\text{known-bias}}$ | $84.5^{\flat}$ | $69.1^{\flat}$ | **+7.6** | $86.4^{\flat}$ | $66.2^{\flat}$ | **+3.1** | 85.0 | 91.3 | 49.0 | +0.3 | 30.9 | +13.3 |
| Conf-reg $_{\textbf{self-debias}}$ | 83.9 | 67.7 | +6.2 | 87.9 | 66.1 | +3.0 | 83.9 | 90.6 | 49.2 | **+0.5** | 33.1 | **+15.5** |
| Conf-reg ♠ $_{\textbf{self-debias}}$ | 84.3 | 67.1 | +5.6 | 87.6 | 66.0 | +2.9 | 85.0 | 91.3 | 48.8 | +0.1 | 28.7 | +11.1 |

Table 1: Models' performance when evaluated on MNLI, Fever, QQP, and their corresponding challenge test sets. The `known-bias` results for MNLI and FEVER are taken from Utama et al. (2020)(♭), Clark et al. (2019)(‡), Mahabadi et al. (2020)(†), and Schuster et al. (2019)(♣). The results of the proposed framework are indicated by `self-debias`. (♠) indicates the training with our proposed *annealing mechanism*. Boldface numbers indicate the highest challenge test set improvement for each debiasing setup on a particular task.

the resulting models perform the task by relying on lexical overlap biases.

**Fact verification** We run debiasing experiments on the FEVER dataset (Thorne et al., 2018). It contains pairs of claim and evidence sentences labeled as either *support*, *refutes*, and *not-enough-information*. We evaluate on the FeverSymmetric test set (Schuster et al., 2019), which is collected to reduced claim-only biases (e.g., negative phrases such as "refused to" or "did not" are associated with the *refutes* label).

### 4.2 Main Model

We apply our self-debiasing framework on the BERT model (Devlin et al., 2019), which performs very well on the three considered tasks.[6] It also shows substantial improvements on the corresponding challenge datasets when trained through the existing debiasing methods (Clark et al., 2019; He et al., 2019). For each examined debiasing method, we show the comparison between the results when it is applied within our framework and when it is trained using prior knowledge to detect training examples with a specific bias. For the second scenario, MNLI and QQP models are debiased using a lexical overlap bias prior, whereas FEVER model is debiased using hand-crafted claim-only biased features. We use the results reported in their corresponding papers. Additionally, we train a baseline BERT model with a standard cross entropy loss.

### 4.3 Hyperparameters

The hyperparameters of our framework include the number of training samples and epochs to train the shallow model $f_b$ as well as parameter $a$ to schedule the annealing process. We only use the training data, and no information about the challenging sets, for tuning these parameters. Based on the insight from our synthetic bias analysis (Section 2), we choose the sample size and the number of epochs which result in $f_b$ that satisfies the following conditions: (1) its accuracy on the unseen training examples is around $60\%$ to $70\%$; (2) More than $90\%$ of their predictions fall into the high confidence bin ($> 0.9$). These variables vary for each task depending on their diversity and difficulty. For instance, it takes $2000$ examples and $3$ epochs of training for MNLI, and only $500$ examples and $4$ epochs for an easier task such as QQP.[7] For the annealing mechanism, we set $a = 0.8$ as the minimum value of $\alpha_t$ for all experiments across the three tasks. Although this may not be an optimal configuration for all tasks, it still allows us to observe how gradually increasing the importance of "biased" examples may affect the overall performance.

## 5 Results and Discussion

**Main results** We experiment with several training methods for each task: the baseline training, debiased training with prior knowledge, and the debiased training using our self-debiasing framework (with and without annealing mechanism). We present the results on the three tasks in Table 1.

[6] We use the pre-trained `bert-base-uncased` model available at `https://huggingface.co/transformers/pretrained_models.html`.

[7] We perform a search on all combinations of $1, 2, 3, 4$, and $5$ epochs and $500, 1000, 1500$, and $2000$ examples.

| Dataset | base. | confidence-regularization (Δ) | | |
|---|---|---|---|---|
| | | known $_{\text{HANS}}$ | self-deb. | self-deb. ♠ |
| SICK | 55.2 | +1.2 ⇒ | +3.0 ⟹ | +2.1 ⟹ |
| RTE | 63.6 | −0.5 ⇐ | +0.5 ⇒ | +0.6 ⇒ |
| Diag. | 58.6 | −0.6 ⇐ | +0.4 ⇒ | +0.5 ⇒ |
| Scitail | 65.4 | +1.4 ⟹ | +0.4 ⇒ | +1.0 ⟹ |

Table 2: Accuracy results of self-debias confidence regularization on cross-dataset evaluation.

Each model is evaluated both in terms of their in-distribution performance on the original development set and their out-of-distribution performance on the challenge test set. For each setting, we report the average results across 5 runs.

We observe that: (1) models trained through self-debiasing framework obtain equally high improvements on challenge sets of the three tasks compared to their corresponding debiased models trained with a prior knowledge (indicated as `known-bias`). In some cases, the existing debiasing methods can even be more effective when applied using the proposed framework, e.g., `self-debias` example reweighting obtains 52.3 F1 score improvement over the baseline on the non-duplicate subset of PAWS (compared to 33.6 by its `known-bias` counterpart). This indicates that the framework is equally effective in identifying biased examples without previously needed prior knowledge; (2) Most improvements on the challenge datasets come at the expense of the in-distribution performance (dev column) except for the confidence regularization models. For instance, the `self-debias` product-of-expert (PoE) model, without annealing, performs 2.2pp lower than the `known-bias` model on MNLI dev set. This indicates that self-debiasing may identify more potentially biased examples and thus effectively omit more training data; (3) Annealing mechanism (indicated by ♠) is effective in mitigating this issue in most cases, e.g., improving PoE by 0.5pp on FEVER dev and 1.2pp on MNLI dev while keeping relatively high challenge test accuracy. Self-debias reweighting augmented with annealing mechanism even achieves the highest HANS accuracy in addition to its improved in-distribution performance.

**Cross-datasets evaluation** Previous work demonstrated that targeting a specific bias to optimize performance in the corresponding challenge dataset may bias the model in other unwanted directions, which proves to be counterproductive in improving the overall robustness (Nie et al., 2019a; Teney et al., 2020). One way to evaluate the impact of debiasing methods on the overall robustness is to train models on one dataset and evaluate them against other datasets of the same task, which may have different types and amounts of biases (Belinkov et al., 2019a). A contemporary work by Wu et al. (2020) specifically finds that debiasing models based on only a single bias results in models that perform significantly worse upon cross-datasets evaluation for the reading comprehension task.

Motivated by this, we perform similar evaluations for models trained on MNLI through the three debiasing setups: `known-bias` to target the HANS-specific bias, `self-debiasing`, and `self-debiasing` augmented with the proposed annealing mechanism. We do not tune the hyperparameters for each target dataset and use the models that we previously reported in the main results. As the target datasets, we use 4 NLI datasets: Scitail (Khot et al., 2018), SICK (Marelli et al., 2014), GLUE diagnostic set (Wang et al., 2018), and 3-way version of RTE 1, 2, and 3 (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007).[8]

We present the results in Table 2. We observe that the debiasing with prior knowledge to target the specific lexical overlap bias (indicated by $\texttt{known}_{\texttt{HANS}}$) can help models to perform better on SICK and Scitail. However, its resulting models under-perform the baseline in RTE sets and GLUE diagnostic, degrading the accuracy by 0.5 and 0.6pp. In contrast, the self-debiased models, with and without annealing mechanism, outperform the baseline on all target datasets, both achieving additional 1.1pp on average. The gains by the two self-debiasing suggest that while they are effective in mitigating the effect of one particular bias (i.e., lexical overlap), they do not result in models learning other unwanted patterns that may hurt the performance on other datasets. These results also extend the findings of Wu et al. (2020) to the NLU settings in that addressing multiple biases at once, as done by our general debiasing method, leads to a better overall generalization.

**Analyzing the annealing mechanism** In previous experiments, we show that setting the minimum $\alpha_t$ to only slightly lower than 1 (i.e., $a = 0.8$)

[8] We compiled and reformated the dataset files which are available at `https://nlp.stanford.edu/projects/contradiction/`.

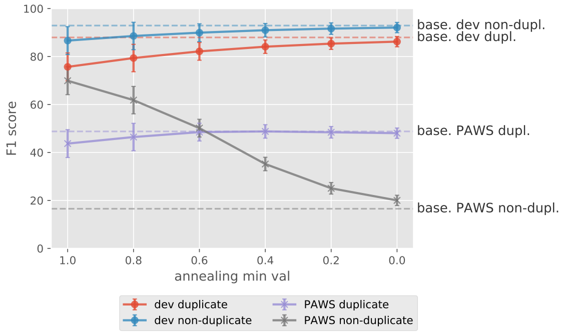


Figure 4: Analysis of the annealing mechanism using different values of minimum $\alpha_t$.

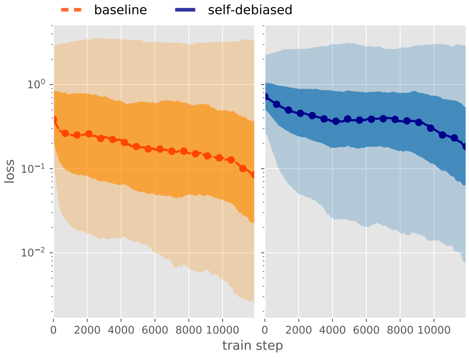


Figure 5: Training loss curves for the first 15K steps by the baseline and self-debias example reweighting training (shown in log scale). Solid lines indicate the median loss within each training batch. The dark and light shadow areas represent the range between 25th to 75th percentile and the range between 0th (minimum) and 100th percentile (maximum), respectively.

results in improvements on the in-distribution without substantial degradation on challenge datasets scores. We question whether this behavior persists once we set $a$ closer to 0. Specifically, do models fall back to the baseline performance when the loss gets more equivalent to the standard cross-entropy at the end of the training?

We run additional experiments using the self-debiased example reweighting on QQP ⇒ PAWS evaluations. We consider the following values to set the minimum $\alpha_t$: $1.0, 0.8, 0.6, 0.4, 0.2$, and $0.0$. For each experiment, we report the average scores across multiple runs. As we see in Figure 4, the challenge test scores decrease as we set minimum $a$ to lower values. Annealing can still offer a reasonable trade-off between in-distribution and challenge test performances up until $a = 0.6$, before falling back to baseline performance at $a = 0$. These results suggest that models are still likely to learn spurious shortcuts from biased examples that they are exposed to even at the end of the training. Consequently, the annealing mechanism should be used cautiously by setting the minimum $\alpha_t$ to moderate values, e.g., 0.6 or 0.8.

**Impact on learning dynamics** We previously show (Figure 2) that baseline models tend to learn easier examples more rapidly, allowing them to make correct predictions by relying on biases. As the self-debiasing framework manages to mitigate this fallible reliance, we expect some changes in models' learning dynamics. We are, therefore, interested in characterizing these changes by analyzing their training loss curve. In particular, we examine the individual losses on each training batch and measure their variability using percentiles (i.e., 0th, 25th, 50th, 75th, and 100th percentile). Figure 5 shows the comparison of the individual loss variability between the baseline and the self-debiased models when trained on MNLI. We observe that the median loss of the baseline model converges faster than the self-debiased counterpart (dotted solid lines). However, examples below its 25th percentile already have losses smaller than $10^{-1}$ when the majority of the losses are still high (darker shadow area). This indicates that unregularized training optimizes faster on certain examples, possibly due to the presence of biases. On the contrary, self-debiased training maintains relatively less variability of losses throughout the training. This result suggests that overconfident predictions (unusually low loss examples) can be an indication of the model utilizing biases. This is in line with the finding of Utama et al. (2020), which shows that regularizing confidence on biased examples leads to improved robustness against biases.

**Bias identification stability** Researchers have recently observed large variability in the generalization performance of fine-tuned BERT model (Mosbach et al., 2020; Zhang et al., 2020), especially in the out-of-distribution evaluation settings (McCoy et al., 2019a; Zhou et al., 2020). This may raise concerns on whether our shallow models, which are trained on the sub-sample of the training data, can consistently learn to rely mostly on biases. We, therefore, train 10 instances of shallow models on the MNLI dataset using different random seeds (for classifier's weight initialization and training sub-sampling). For evaluation, we perform two different partitionings of MNLI dev set based on the output of two simple hand-crafted models, which use lexical overlap and hypothesis-only features

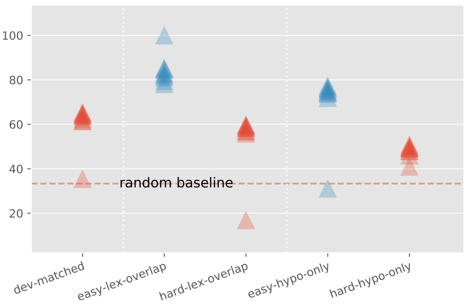


Figure 6: Evaluation of 10 shallow model instances on easy/hard partitioning of MNLI dev based on the presence of lexical overlap and hypothesis-only biases. The results suggest the stability of shallow models in capturing the two biases.

(Gururangan et al., 2018), respectively. The stability of bias utilization across the runs is evaluated by measuring their performance on *easy* and *hard* subsets of each partitioning, where examples that simple models predicted correctly belong to *easy* and the rest belong to *hard*.[9]

Figure 6 shows the results. We observe small variability in the overall dev set performance which ranges in 61-65% accuracy. Similarly, the models obtain consistently higher accuracy on the easy subsets over the hard ones: 79-85% vs. 56-59% on the lexical-overlap partitioning and 72-77% vs. 48-50% on the hypothesis-only partitioning. The results indicate that: 1) the bias-reliant behavior of shallow models is stable; and 2) shallow models capture multiple types of bias. However, we also observe one rare instance of the shallow model that fails to converge during training and is stuck at making random predictions (33% in MNLI). This may indicate that the biased examples are undersampled in that particular run. In that case, we can easily spot this undesired behavior, discard the model, and perform another sampling.

## 6 Related Work

The artifacts of large scale dataset collections result in dataset biases that allow models to perform well without learning the intended reasoning skills. In NLI, models can perform better than chance by only using the partial input (Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018) or by basing their predictions on whether the inputs are highly overlapped (McCoy et al., 2019b; Dasgupta et al., 2018). Similar phenomena exist in various tasks, including argumentation mining (Niven and Kao, 2019), reading comprehension (Kaushik and Lipton, 2018), or story cloze completion (Schwartz et al., 2017; Cai et al., 2017). To allow a better evaluation of models' reasoning capabilities, researchers constructed challenge test sets composed of "counterexamples" to the spurious shortcuts that models may adopt (Jia and Liang, 2017; Glockner et al., 2018; Zhang et al., 2019; Naik et al., 2018). Models evaluated on these sets often fall back to random baseline performance.

There has been a flurry of work in dynamic dataset construction to systematically reduce dataset biases through adversarial filtering (Zellers et al., 2018; Sakaguchi et al., 2020; Bras et al., 2020) or human in the loop (Nie et al., 2019b; Kaushik et al., 2020; Gardner et al., 2020). While promising, researchers also show that newly constructed datasets may not be fully free of hidden biased patterns (Sharma et al., 2018). It is thus crucial to complement the data collection efforts with learning algorithms that are more robust to biases, such as the recently proposed product-of-expert (Clark et al., 2019; He et al., 2019; Mahabadi et al., 2020), confidence regularization (Utama et al., 2020), or adversarial training (Belinkov et al., 2019b). Despite their effectiveness, these methods are limited by their assumption on the availability of information about the task-specific biases. Our framework aims to alleviate this limitation and enable them to address *unknown* biases. In the same vein as ours, Yaghoobzadeh et al. (2019) and Sanh et al. (2021) identify biased training instances automatically via "example forgetting" measure (Toneva et al., 2019) and limited capacity models, respectively. Lastly, Tu et al. (2020) incorporate auxiliary datasets through multi-task learning to improve model's robustness without strong *a priori* knowledge about the biases.

## 7 Conclusion

We present a general self-debiasing framework to address the impact of unknown dataset biases by omitting the need for thorough dataset-specific analysis to discover the types of biases for each new dataset. We adopt the existing debiasing methods into our framework and enable them to obtain equally high improvements on several *challenge test sets* without targeting a specific bias. The evaluation also suggests that our framework results

[9] Although this may seem to be against the spirit of not using prior knowledge about the biases, we believe that this step is necessary to show the stability of the shallow models and to validate if they indeed capture the intended biases.

in better overall robustness compared to the bias-specific counterparts. Based on our analysis, future work in the direction of automatic bias mitigation may include identifying potentially biased examples in an *online* fashion and discouraging models from exploiting them throughout the training.

## Acknowledgments

We thank Max Glockner, Mingzhu Wu, Johannes Daxenberger, and the anonymous reviewers for their constructive feedback. This work is funded by the German Research Foundation through the research training group AIPHES (GRK 1994/1) and by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

## References

Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 233–242. JMLR.org.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second pascal recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019a. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart M. Shieber, Benjamin Van Durme, and Alexander Rush. 2019b. On adversarial removal of hypothesis-only bias in natural language inference. In *Joint Conference on Lexical and Computational Semantics (StarSem)*.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *ICML*.

Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay attention to the ending:strong neural baselines for the ROC story cloze task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 616–622, Vancouver, Canada. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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 4067–4080, Hong Kong, China. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. *ArXiv*, abs/1802.04302.

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. Association for Computational Linguistics.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1602–1611. PMLR.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating NLP models via contrast sets. *arXiv preprint arXiv:2004.02709*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, 26 April - 1 May, 2019*. OpenReview.net.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. 2020. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*.

Rabeeh Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

R Thomas McCoy, Junghyun Min, and Tal Linzen. 2019a. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance. *arXiv preprint arXiv:1911.02969*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019b. 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. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019a. Analyzing compositionality-sensitivity of nli models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6867–6874.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019b. Adversarial nli: A new benchmark for natural language understanding. *ArXiv*, abs/1910.14599.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *9th International Conference on Learning Representations, ICLR 2021, 2021*. OpenReview.net.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. 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 3417–3423, Hong Kong, China. Association for Computational Linguistics.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online. Association for Computational Linguistics.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, Melbourne, Australia. Association for Computational Linguistics.

Rion Snow, Lucy Vanderwende, and Arul Menezes. 2006. Effectively using syntax for recognizing false entailment. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 33–40, New York City, USA. Association for Computational Linguistics.

Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel. 2020. On the value of out-of-distribution testing: An example of goodhart's law. *ArXiv*, abs/2005.09241.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association of Computational Linguistics*.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.

Lucy Vanderwende and William B. Dolan. 2006. What syntax can contribute in the entailment task. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 205–216, Berlin, Heidelberg. Springer Berlin Heidelberg.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Mingzhu Wu, Nafise Sadat Moosavi, Andreas Rücklé, and Iryna Gurevych. 2020. Improving QA generalization by concurrent modeling of multiple biases. In *Proceedings of the Findings of ACL: EMNLP 2020*, Online. Association for Computational Linguistics.

Yadollah Yaghoobzadeh, Remi Tachet, Timothy J Hazen, and Alessandro Sordoni. 2019. Robust natural language inference models with example forgetting. *arXiv preprint arXiv:1911.03861*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. *arXiv preprint arXiv:2004.13606*.

## A Natural Language Inference

**Main model** We finetune the BERT base model for all settings (baseline, known-bias, and self-debiasing) using default parameters: 3 epochs of training with learning rate $5^{-5}$. An exception is made for product-of-expert and confidence regularization, where we follow He et al. (2019) to run the training longer, i.e. 5 epochs.

**Shallow model** The shallow model for MNLI is trained on 2K of examples for 3 epochs using the default learning rate of $5^{-5}$.

## B Fact verification

**Main model** We follow Schuster et al. (2019) in finetuning the BERT base model on FEVER dataset using the following parameters: learning rate $2^{-5}$ and 3 epochs of training.

**Shallow model** The shallow model can be trained in lesser amount of data, 500 examples. We train the model for 5 epochs with the same learning rate, $2^{-5}$.

## C Paraphrase Identification

**Main model** We follow Utama et al. (2020) in setting the parameters for training a QQP model: learning rate $2^{-5}$ and 3 epochs of training.

**Shallow model** Similar to FEVER, we train the shallow model using only 500 examples. It converges in 4 epochs using the same learning rate, $2^{-5}$.

## D Synthetic MNLI Results

We report the final accuracy of models when trained on our synthetic bias datasets. We show that the anti-biased accuracy correlates negatively with the proportion of the biased examples. We present the results in Table 3.

| **Bias prop.** | **test sets** | | |
|---|---|---|---|
| | original | biased | anti-biased |
| 0.9 | 83.6 $\Leftarrow$ | 97.1 $\Longrightarrow$ | 61.7 $\Longleftarrow$ |
| 0.8 | 83.7 $\Leftarrow$ | 95.3 $\Longrightarrow$ | 70.4 $\Longleftarrow$ |
| 0.7 | 83.9 $\Leftarrow$ | 92.8 $\Rightarrow$ | 75.5 $\Leftarrow$ |
| 0.6 | 84.1 $=$ | 90.9 $\Rightarrow$ | 78.5 $\Leftarrow$ |

Table 3: Final accuracy of models trained on synthetic bias datasets.

## E Detailed HANS Results

HANS dataset (McCoy et al., 2019b) consist of three subsets, covering different inference phenomena which happen to have lexical overlap: (a) Lexical overlap e.g., "*The doctor was paid by the actor*" vs. "*The doctor paid the actor*"; (b) Subsequence, e.g., "*The doctor near the actor danced*" vs. "*The actor danced*"; and (c) Constituent e.g., "*If the artist slept, the actor ran*" vs. "*The artist slept*". Each subset contains examples of both entailment and non-entailment. The 3-way predictions on MNLI is mapped to HANS by taking max pool between neutral and contradiction labels. We present the results of our experiments in Table 4.

| **Method** | **HANS all sets (Acc.)** | | | | | |
|---|---|---|---|---|---|---|
| | **Lex** | **Lex.** | **Sub.** | **Sub.** | **Con.** | **¬Con.** |
| BERT-base | 96.0 | 51.8 | 99.5 | 7.4 | 99.4 | 14.5 |
| Rew. **self-debias** | 81.3 | 73.3 | 94.7 | 34.5 | 92.8 | 42.3 |
| Rew. ♠ **self-debias** | 84.7 | 77.1 | 96.0 | 30.5 | 95.3 | 37.4 |
| PoE **self-debias** | 77.0 | 73.6 | 92.1 | 42.2 | 89.3 | 49.8 |
| PoE ♠ **self-debias** | 78.5 | 67.7 | 91.3 | 28.6 | 95.4 | 45.1 |
| Conf-reg **self-debias** | 81.8 | 78.2 | 93.7 | 31.7 | 95.1 | 31.5 |
| Conf-reg ♠ **self-debias** | 87.4 | 74.5 | 96.3 | 27.4 | 95.1 | 26.6 |

Table 4: Models' performance on HANS challenge test set (McCoy et al., 2019b). Column `lex.`, `con.`, and `sub.` stand for lexical overlap, constituency, and subsequence, respectively. The (¬) symbol indicates the non-entailment subset.

# Chapter 7

# Robustness in Low Resource Learning Settings

# Avoiding Inference Heuristics in Few-shot Prompt-based Finetuning

**Prasetya Ajie Utama**[†‡] **, Nafise Sadat Moosavi**[‡], **Victor Sanh**[♣], **Iryna Gurevych**[‡]

[†]Research Training Group AIPHES
[‡]UKP Lab, Technische Universität Darmstadt
[♣]Hugging Face, Brooklyn, USA
[‡]https://www.ukp.tu-darmstadt.de
utama@ukp.tu-darmstadt.de

## Abstract

Recent *prompt-based* approaches allow pretrained language models to achieve strong performances on *few-shot finetuning* by reformulating downstream tasks as a language modeling problem. In this work, we demonstrate that, despite its advantages on low data regimes, finetuned prompt-based models for sentence pair classification tasks still suffer from a common pitfall of adopting inference heuristics based on lexical overlap, e.g., models incorrectly assuming a sentence pair is of the same meaning because they consist of the same set of words. Interestingly, we find that this particular inference heuristic is significantly less present in the zero-shot evaluation of the prompt-based model, indicating how finetuning can be *destructive* to useful knowledge learned during the pretraining. We then show that adding a regularization that preserves pretraining weights is effective in mitigating this destructive tendency of few-shot finetuning. Our evaluation on three datasets demonstrates promising improvements on the three corresponding challenge datasets used to diagnose the inference heuristics.[1]

## 1 Introduction

Prompt-based finetuning has emerged as a promising paradigm to adapt Pretrained Language Models (PLM) for downstream tasks with limited number of labeled examples (Schick and Schütze, 2021a; Radford et al., 2019). This approach reformulates downstream task instances as a language modeling input,[2] allowing PLMs to make non-trivial task-specific predictions even in zero-shot settings. This in turn, provides a good initialization point for data efficient finetuning (Gao et al., 2021), resulting in a strong advantage on low data regimes where the standard finetuning paradigm struggles. However, the success of this prompting approach has only been shown using common held-out evaluations, which often conceal certain undesirable behaviors of models (Niven and Kao, 2019).

One such behavior commonly reported in downstream models is characterized by their preference to use surface features over general linguistic information (Warstadt et al., 2020). In the Natural Language Inference (NLI) task, McCoy et al. (2019) documented that models preferentially use the lexical overlap feature between sentence pairs to blindly predict that one sentence *entails* the other. Despite models' high in-distribution performance, they often fail on counterexamples of this *inference heuristic*, e.g., they predict that "*the cat chased the mouse*" entails "*the mouse chased the cat*".

At the same time, there is a mounting evidence that pre-training on large text corpora extracts rich linguistic information (Hewitt and Manning, 2019; Tenney et al., 2019). However, based on recent studies, standard finetuned models often overlook this information in the presence of lexical overlap (Nie et al., 2019; Dasgupta et al., 2018). We therefore question whether direct adaptation of PLMs using prompts can better transfer the use of this information during finetuning. We investigate this question by systematically studying the heuristics in a prompt-based model finetuned across three datasets with varying data regimes. Our intriguing results reveal that: (i) zero-shot prompt-based models are more robust to using the lexical overlap heuristic during inference, indicated by their high performance on the corresponding challenge datasets; (ii) however, prompt-based finetuned models quickly adopt this heuristic as they learn from more labeled data, which is indicated by gradual degradation of the performance in challenge datasets.

We then show that regularizing prompt-based finetuning, by penalizing the learning from up-

[1] The code is available at https://github.com/UKPLab/emnlp2021-prompt-ft-heuristics

[2] E.g., appending a cloze prompt "It was [MASK]" to a sentiment prediction input sentence "Delicious food!", and obtaining the sentiment label by comparing the probabilities assigned to the words "great" and "terrible".

dating the weights too far from their original pre-trained values, is an effective approach to improve the in-distribution performance on target datasets, while mitigating the adoption of inference heuristics. Overall, our work suggests that while prompt-based finetuning has gained impressive results on standard benchmarks, it can has a negative impact regarding inference heuristics, which in turn suggests the importance of a more thorough evaluation setup to ensure meaningful progress.

## 2 Inference Heuristics in Prompt-based Finetuning

**Prompt-based PLM Finetuning** In this work, we focus on sentence pairs classification tasks, where the goal is to predict semantic relation $y$ of an input pair $x = (s_1, s_2)$. In a standard finetuning setting, $s_1$ and $s_2$ are concatenated along with a special token `[CLS]`, whose embedding is used as an input to a newly initialized *classifier head*.

The *prompt-based* approach, on the other hand, reformulates pair $x$ as a masked language model input using a pre-defined template and word-to-label mapping. For instance, Schick and Schütze (2021a) formulate a natural language inference instance $(s_1, s_2, y)$ as:

$$\texttt{[CLS]}\, s_1 \texttt{?}\, \texttt{[MASK]}\texttt{,}\, s_2 \texttt{[SEP]}$$

with the following mapping for the masked token: "yes"→ "entailment", "maybe"→"neutral", and "no" → "contradiction". The probabilities assigned by the PLM to the label words at the `[MASK]` token can then be directly used to make task-specific predictions, allowing PLM to perform in a zero-shot setting. Following Gao et al. (2021), we further finetune the prompt-based model on the available labeled examples for each task. Note that this procedure finetunes only the existing pre-trained weights, and does not introduce new parameters.

**Task and Datasets** We evaluate on three English language datasets included in the GLUE benchmark (Wang et al., 2018) for which there are challenge datasets to evaluate the lexical overlap heuristic: MNLI (Williams et al., 2018), SNLI (Bowman et al., 2015), and Quora Question Pairs (QQP). In MNLI and SNLI, the task is to determine whether premise sentence $s_1$ *entails*, *contradicts*, or is *neutral* to the hypothesis sentence $s_2$. In QQP, $s_1$ and $s_2$ are a pair of questions that are labeled as either *duplicate* or *non-duplicate*.

| **Original Input** | |
|---|---|
| **Premise**<br>**Hypothesis**<br>**Label** | The actors that danced saw the author.<br>The actors saw the author.<br>entailment **(support)** |
| **Premise**<br>**Hypothesis**<br>**Label** | The managers near the scientist resigned.<br>The scientist resigned.<br>non-entailment **(against)** |
| **Reformulated Input** | |
| **Premise**<br><br>**Label word** | The actors that danced saw the author? `[MASK]`, the actors saw the author.<br>*Yes* |
| **Premise**<br><br>**Label word** | The managers near the scientist resigned? `[MASK]`, the scientist resigned.<br>*No / Maybe* |

Table 1: **Top:** input examples of the NLI task that **support** or are **against** the lexical overlap heuristics. **Bottom:** reformulated NLI instances as masked language model inputs with the expected label words.

Researchers constructed corresponding *challenge* sets for the above datasets, which are designed to contain examples that are *against* the heuristics, i.e., the examples exhibit word overlap between the two input sentences but are labeled as non-entailment for NLI or non-duplicate for QQP. We evaluate each few-shot model against its corresponding challenge dataset. Namely, we evaluate models trained on MNLI against entailment and non-entailment subsets of the HANS dataset (McCoy et al., 2019), which are further categorized into lexical overlap (lex.), subsequence (subseq.), and constituent (const.) subsets; SNLI models against the long and short subsets of the Scramble Test challenge set (Dasgupta et al., 2018); and QQP models against the PAWS dataset (Zhang et al., 2019).[3] We illustrate challenge datasets examples and their reformulation as prompts in Table 1.

**Model and Finetuning** Our training and standard evaluation setup closely follow Gao et al. (2021), which measure finetuning performances across five different randomly sampled training data of size $\mathbf{K}$ to account for finetuning instability on small datasets (Dodge et al., 2020; Mosbach et al., 2021). We perform five data subsampling for each dataset and each data size $\mathbf{K}$, where $\mathbf{K} \in \{16, 32, 64, 128, 256, 512\}$. Note that $\mathbf{K}$ indicates the number of examples *per label*. We use the original development sets of each training dataset for testing the *in-distribution* performance. We per-

[3] See appendix A for details of HANS, PAWS, and Scramble Test test sets.

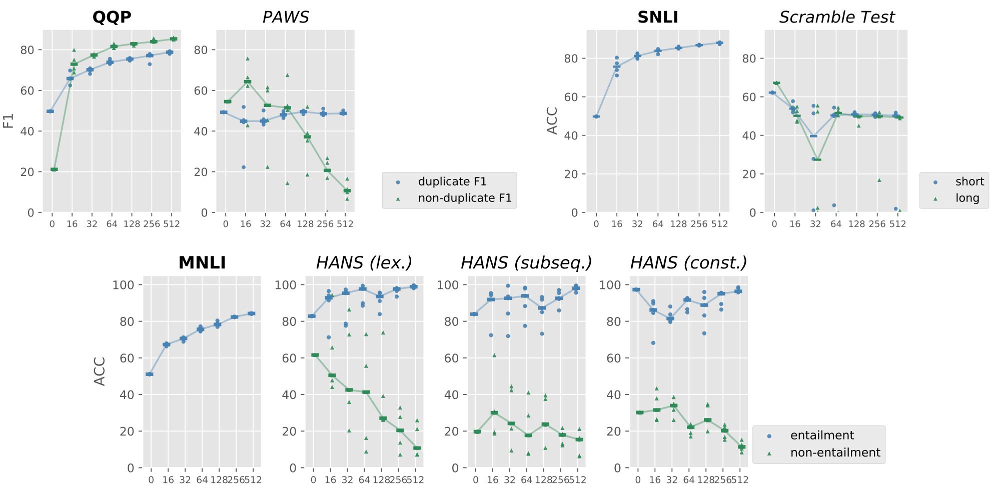


Figure 1: In-distribution (**bold**) vs. challenge datasets (*italic*) evaluation results of prompt-based finetuning across different data size $\mathbf{K}$ (x axis), where $\mathbf{K} = 0$ indicates zero-shot evaluation. In all challenge sets, the overall zero-shot performance (both blue and green plots) degrades as the model is finetuned using more data.

form all experiments using the RoBERTa-large model (Liu et al., 2019b).

**Inference heuristics across data regimes** We show the results of the prompt-based finetuning across different $\mathbf{K}$ in Figure 1. For the in-distribution evaluations (leftmost of each plot), the prompt-based models finetuned on MNLI, SNLI, and QQP improve rapidly with more training data before saturating at $\mathbf{K} = 512$. In contrast to the in-distribution results, we observe a different trajectory of performance on the three challenge datasets. On the Scramble and HANS sets, prompt-based models show non-trivial zero-shot performance ($\mathbf{K} = 0$) that is above its in-distribution counterpart. However, as more data is available, the models exhibit stronger indication of adopting heuristics. Namely, the performance on examples subset that *support* the heuristics increases, while the performance on cases that are *against* heuristics decreases. This pattern is most pronounced on the lexical overlap subset of HANS, where the median accuracy on non-entailment subset drops to below 10% while the entailment performance reaches 100%. The results suggest that few-shot finetuning can be destructive against the initial ability of prompt-based classifier to ignore surface features like lexical overlap. Finetuning appears to over-adjust model parameters to the small target data, which contain very few to no counter-examples to the heuristics (Min et al., 2020; Lovering et al., 2021).

## 3 Avoiding Inference Heuristics

Here we look to mitigate the adverse impact of finetuning by viewing the issue as an instance of catastrophic forgetting (French, 1999), which is characterized by the loss of performance on the original dataset after subsequent finetuning on new data. We then propose a regularized prompt-based finetuning based on the Elastic Weight Consolidation (EWC) method (Kirkpatrick et al., 2017), which penalizes updates on weights crucial for the original zero-shot performance. EWC identifies these weights using empirical Fisher matrix (Martens, 2020), which requires samples of the original dataset. To omit the need of accessing the pretraining data, we follow Chen et al. (2020) that assume stronger independence between the Fisher information and the corresponding weights. The penalty term is now akin to the L2 loss between updated weights $\theta_i$ and the original weights $\theta_i^*$, resulting in the following overall loss:

$$\mathbf{L}_{rFT} = \alpha \mathbf{L}_{FT} + (1-\alpha)\frac{\lambda}{2}\sum_i (\theta_i - \theta_i^*)^2$$

where $\mathbf{L}_{FT}$ is a standard cross entropy, $\lambda$ is a quadratic penalty coefficient, and $\alpha$ is a coefficient to linearly combine the two terms. We use the RecAdam implementation (Chen et al., 2020) for this loss, which also applies an annealing mechanism to gradually upweight the standard loss $\mathbf{L}_{FT}$ toward the end of training.[4]

[4] See Appendix A for implementation details.

| | **MNLI** (acc.) | | | **QQP** (F1) | | | **SNLI** (acc.) | | |
|---|---|---|---|---|---|---|---|---|---|
| | **In-dist.** | *HANS* | **avg.** | **In-dist.** | *PAWS* | **avg.** | **In-dist.** | *Scramble* | **avg.** |
| **Prompt-based** | | | | | | | | | |
| *zero-shot* #0 | 51.1 | 62.6 | 56.8 | 35.4 | 51.8 | 43.6 | 49.7 | 64.7 | 57.2 |
| FT #512 | 84.3 | 54.8 | 69.5 | 82.1 | 29.6 | 55.8 | 88.1 | 50.1 | 69.1 |
| **rFT #512** | 82.7 | 60.2 | **71.5** | 81.5 | 37.1 | **59.3** | 87.6 | 55.4 | **71.5** |
| FT-fix18 #512 | 76.5 | 61.6 | 69.1 | 78.6 | 35.6 | 57.1 | 84.5 | 45.3 | 64.9 |
| FT-fix12 #512 | 83.5 | 54.3 | 68.9 | 81.9 | 35.3 | 57.1 | 87.1 | 50.5 | 68.8 |
| FT-fix6 #512 | 84.2 | 52.9 | 68.5 | 82.1 | 32.7 | 57.4 | 87.9 | 50.1 | 68.9 |
| **Classifier head** | | | | | | | | | |
| FT #512 | 81.4 | 52.6 | 67.0 | 80.9 | 26.8 | 53.8 | 86.5 | 49.8 | 68.1 |

Table 2: Results of different strategies for finetuning prompt-based model (using #$k$ examples). Models are evaluated against the in-distribution set and corresponding challenge sets. The zero-shot row indicates prompting results before finetuning. The *avg* columns report the average score on in-distribution and challenge datasets.

**Baselines** We compare regularized finetuning with another method that also minimally update the pretraining weights. We consider simple weight fixing of the first $n$ layers of the pretrained model, where the $n$ layers are frozen (including the token embeddings) and only the weights of upper layers and LM head are updated throughout the finetuning. In the evaluation, we use $n \in \{6, 12, 18\}$. We refer to these baselines as FT-fix$n$.

**Results** We evaluate all the considered finetuning strategies by taking their median performance after finetuning on 512 examples (for each label) and compare them with the original zero-shot performance. We report the results on Table 2, which also include the results of standard classifier head finetuning (last row). We observe the following: (1) Freezing the layers has mixed challenge set results, e.g., FT-fix18 improves over vanilla prompt-based finetuning on HANS and PAWS, but degrades Scramble and all in-distribution performances; (2) The L2 regularization strategy, rFT, achieves consistent improvements on the challenge sets while only costs small drop on the corresponding in-distribution performance, e.g., +6pp, +8pp, and +5pp on HANS, PAWS, and Scramble, respectively; (3) Although vanilla *prompt-based* finetuning performs relatively poorly, it still has an advantage over standard *classifier head* finetuning by +2.5pp, +2.0pp, and +1.0pp on the average scores of each in-distribution and challenge dataset pair.

Additionally, Figure 2 shows rFT's improvement over vanilla prompt-based finetuning across data regimes on MNLI and HANS. We observe that the advantage of rFT is the strongest on the lexical overlap subset, which initially shows the highest zero-shot performance. The results also suggest that the benefit of rFT peaks at mid data regimes (e.g., $\mathbf{K} = 32$), before saturating when training data size is increased further. We also note that our results are consistent when we evaluate alternative prompt templates, or finetune for varying number of epochs.[5] The latter indicates that the adoption of inference heuristics is more likely attributed to the amount of training examples rather than the number of learning steps.

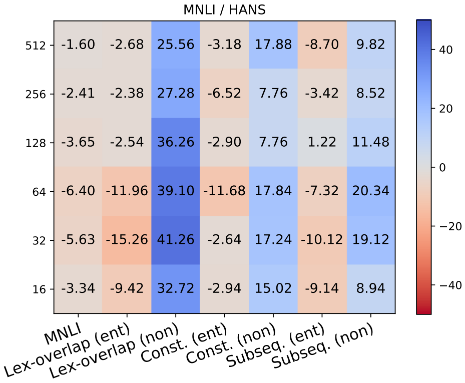


Figure 2: Relative difference between median accuracy of prompt-based finetuning across data regimes (y axis) with and without regularization on MNLI and HANS.

## 4 Related Work

**Inference Heuristics** Our work relates to a large body of literature on the problem of "bias" in the training datasets and the ramifications to the resulting models across various language understanding tasks (Niven and Kao, 2019; Poliak et al., 2018; Tsuchiya, 2018; Gururangan et al., 2020). Previ-

[5] See Appendix B for the detailed results.

ous work shows that the artifacts of data annotations result in spurious surface cues, which gives away the labels, allowing models to perform well without properly learning the intended task. For instance, models are shown to adopt heuristics based on the presence of certain indicative words or phrases in tasks such as reading comprehension (Kaushik and Lipton, 2018), story cloze completion (Schwartz et al., 2017; Cai et al., 2017), fact verification (Schuster et al., 2019), argumentation mining (Niven and Kao, 2019), and natural language inference (Gururangan et al., 2020). Heuristics in models are often investigated using constructed "challenge datasets" consisting of counter-examples to the spurious cues, which mostly result in incorrect predictions (Jia and Liang, 2017; Glockner et al., 2018; Naik et al., 2018; McCoy et al., 2019). Although the problem has been extensively studied, most works focus on models that are trained in standard settings where larger training datasets are available. Our work provides new insights in inference heuristics in models that are trained in zero- and few-shot settings.

**Heuristics Mitigation** Significant prior work attempt to mitigate the heuristics in models by improving the training dataset. Zellers et al. (2019); Sakaguchi et al. (2020) propose to reduce artifacts in the training data by using adversarial filtering methods; Nie et al. (2020); Kaushik et al. (2020) aim at a similar improvement via iterative data collection using human-in-the-loop; Min et al. (2020); Schuster et al. (2021); Liu et al. (2019a); Rozen et al. (2019) augment the training dataset with adversarial instances; and Moosavi et al. (2020a) augment each training instances with their semantic roles information. Complementary to this, recent work introduces various learning algorithms to avoid adopting heuristics including by re-weighting (He et al., 2019; Karimi Mahabadi et al., 2020; Clark et al., 2020) or regularizing the confidence (Utama et al., 2020a; Du et al., 2021) on the training instances which exhibit certain biases. The type of bias can be identified automatically (Yaghoobzadeh et al., 2021; Utama et al., 2020b; Sanh et al., 2021; Clark et al., 2020) or by hand-crafted models designed based on prior knowledge about the bias. Our finding suggests that prompted zero-shot models are less reliant on heuristics when tested against examples containing the cues, and preserving this learned behavior is crucial to obtain more robust finetuned models.

**Efficiency and Robustness** Prompting formulation enables language models to learn efficiently from a small number of training examples, which in turn reduces the computational cost for training (Le Scao and Rush, 2021). The efficiency benefit from prompting is very relevant to the larger efforts towards sustainable and green NLP models (Moosavi et al., 2020b; Schwartz et al., 2020a) which encompass a flurry of techniques including knowledge distillation (Hinton et al., 2015; Sanh et al., 2019), pruning (Han et al., 2015), quantization (Jacob et al., 2018), and early exiting (Schwartz et al., 2020b; Xin et al., 2020). Recently, Hooker et al. (2020) show that methods improving compute and memory efficiency using pruning and quantization may be at odds with robustness and fairness. They report that while performance on standard test sets is largely unchanged, the performance of efficient models on certain underrepresented subsets of the data is disproportionately reduced, suggesting the importance of a more comprehensive evaluation to estimate overall changes in performance.

## 5 Conclusion

Our experiments shed light on the negative impact of low resource finetuning to the models' overall performance that is previously obscured by standard evaluation setup. The results indicate that while finetuning helps prompt-based models to rapidly gain the *in-distribution* improvement as more labeled data are available, it also gradually increases models' reliance on *surface heuristics*, which we show to be less present in the zero-shot evaluation. We further demonstrate that applying regularization that preserves pretrained weights during finetuning mitigates the adoption of heuristics while also maintains high in-distribution performances.

## Acknowledgement

We thank Michael Bugert, Tim Baumgärtner, Jan Buchman, and the anonymous reviewers for their constructive feedback. This work is funded by the German Research Foundation through the research training group AIPHES (GRK 1994/1) and by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

## References

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. Association for Computational Linguistics.

Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay attention to the ending:strong neural baselines for the ROC story cloze task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 616–622, Vancouver, Canada. Association for Computational Linguistics.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7870–7881, Online. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. Association for Computational Linguistics.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society, CogSci 2018, Madison, WI, USA, July 25-28, 2018*. cognitivesciencesociety.org.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. *arXiv preprint arXiv:2103.06922*.

R. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3:128–135.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both weights and connections for efficient neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1135–1143, Cambridge, MA, USA. MIT Press.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*.

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. 2020. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, 26 April - 1 May, 2019*. OpenReview.net.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, G. Desjardins, Andrei A. Rusu, K. Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019a. Inoculation by fine-tuning: A method for analyzing challenge datasets. 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 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2021. Predicting inductive biases of pretrained models. In *International Conference on Learning Representations, ICLR 2021, Virtual Conference, 3 May - 8 May, 2021*. OpenReview.net.

James Martens. 2020. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76.

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. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Nafise Sadat Moosavi, Marcel de Boer, Prasetya Ajie Utama, and Iryna Gurevych. 2020a. Improving robustness by augmenting training sentences with predicate-argument structures. *arXiv preprint arXiv:2010.12510*.

Nafise Sadat Moosavi, Angela Fan, Vered Shwartz, Goran Glavaš, Shafiq Joty, Alex Wang, and Thomas Wolf, editors. 2020b. *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. Association for Computational Linguistics, Online.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019. Analyzing compositionality-sensitivity of nli models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6867–6874.

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. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Ohad Rozen, Vered Shwartz, Roee Aharoni, and Ido Dagan. 2019. Diversify your datasets: Analyzing generalization via controlled variance in adversarial datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 196–205, Hong Kong, China. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 8732–8740. AAAI Press.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. 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 3419–3425, Hong Kong, China. Association for Computational Linguistics.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020a. Green AI. *Communications of the ACM*, 63(12):54–63.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. 2020b. The right tool for the job: Matching model and instance complexities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6640–6651, Online. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020b. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018.

GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Ji Xin, Rodrigo Nogueira, Yaoliang Yu, and Jimmy Lin. 2020. Early exiting BERT for efficient document ranking. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 83–88, Online. Association for Computational Linguistics.

Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet des Combes, T. J. Hazen, and Alessandro Sordoni. 2021. Increasing robustness to spurious correlations using forgettable examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3319–3332, Online. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

# A Experimental Details

**Manual templates and mapping** We use the following prompt templates and word-to-label mapping for the three tasks we evaluate on:

| Template | Label Words |
|---|---|
| **MNLI (manual): entailment, neutral, contradiction** | |
| $s_1$?[MASK], $s_2$ | Yes, Maybe, No |
| **SNLI (manual): entailment, neutral, contradiction** | |
| $s_1$?[MASK], $s_2$ | Yes, Maybe, No |
| **QQP (manual): duplicate, non-duplicate** | |
| $s_1$[MASK], $s_2$ | Yes, No |
| **MNLI (auto): entailment, neutral, contradiction** | |
| $s_1$ . [MASK], you are right , $s_2$ | Fine, Plus, Otherwise |
| $s_1$ . [MASK], you're right , $s_2$ | There, Plus, Otherwise |
| $s_1$ . [MASK] ! $s_2$ | Meaning, Plus, Otherwise |

Table 3: Templates and label words used to finetune and evaluate on MNLI, SNLI, and QQP.

The last 3 rows are automatically generated templates and label words that are shown by Gao et al. (2021) to improve the few-shot finetuning further. Note that we use the corresponding task's template when evaluating on the challenge datasets.

**Challenge datasets** We provide examples from each challenge datasets considered in our evaluation to illustrate sentence pairs that support or are against the heuristics. Table 4 shows examples for HANS, PAWS, and Scramble Test. Following McCoy et al. (2019), we obtain the probability for the *non-entailment* label by summing the probabilities assigned by models trained on MNLI to the *neutral* and *contradiction* labels. We use the *same-type* subset of Scramble Test (Dasgupta et al., 2018) which contain examples of both entailment (*support*) and contradiction (*against*) relations.

**HANS details** HANS dataset is designed based on the insight that the word overlapping between premise and hypothesis in NLI datasets is spuriously correlated with the *entailment* label. HANS consists of examples in which relying to this correlation leads to incorrect label, i.e., hypotheses are *not entailed* by their word-overlapping premises. HANS is split into three test cases: (a) **Lexical overlap** (e.g., "*The doctor was paid by the actor*" → "*The doctor paid the actor*"), (b) **Subsequence** (e.g., "*The doctor near the actor danced*" → "*The actor danced*"), and (c) **Constituent** (e.g., "*If the artist slept, the actor ran*" → "*The artist slept*"). Each subset contains both entailment and non-entailment examples that always exhibit word overlap.

| | |
|---|---|
| **HANS** (McCoy et al., 2019) | |
| premise | The artists avoided the senators that thanked the tourists. |
| hypothesis | The artists avoided the senators. |
| label | entailment **(support)** |
| premise | The managers near the scientist resigned. |
| hypothesis | The scientist resigned. |
| label | non-entailment **(against)** |
| **PAWS** (Zhang et al., 2019) | |
| S1 | What are the driving rules in Georgia versus Mississippi? |
| S2 | What are the driving rules in Mississippi versus Georgia? |
| label | duplicate **(support)** |
| S1 | Who pays for Hillary's campaigning for Obama? |
| S2 | Who pays for Obama's campaigning for Hillary? |
| label | non-duplicate **(against)** |
| **Scramble Test** (Dasgupta et al., 2018) | |
| premise | The woman is more cheerful than the man. |
| hypothesis | The woman is more cheerful than the man. |
| label | entailment **(support)** |
| premise | The woman is more cheerful than the man. |
| hypothesis | The man is more cheerful than the woman. |
| label | contradiction **(against)** |

Table 4: Sampled examples from each of the challenge datasets we used for evaluation.

**Hyperparameters** Following Schick and Schütze (2021b,a), we use a fixed set of hyperparameters for all finetuning: learning rate of $1e^{-5}$, batch size of 8, and maximum length size of 256.

**Regularization implementation** We use the RecAdam implementation by Chen et al. (2020) with the following hyperparameters. We set the quadratic penalty $\lambda$ to 5000, and the linear combination factor $\alpha$ is set dynamically throughout the training according to a sigmoid function schedule, where $\alpha$ at step $t$ is defined as:

$$\alpha = s(t) = \frac{1}{1 + \exp(-k \cdot (t - t_0))}$$

where parameter $k$ regulates the rate of the sigmoid, and $t_0$ sets the point where $s(t)$ goes above 0.5. We set $k$ to 0.01 and $t_0$ to 0.6 of the total training steps.

# B Additional Results

**Standard `CLS` finetuning** Previously, Gao et al. (2021) reported that the performance of standard

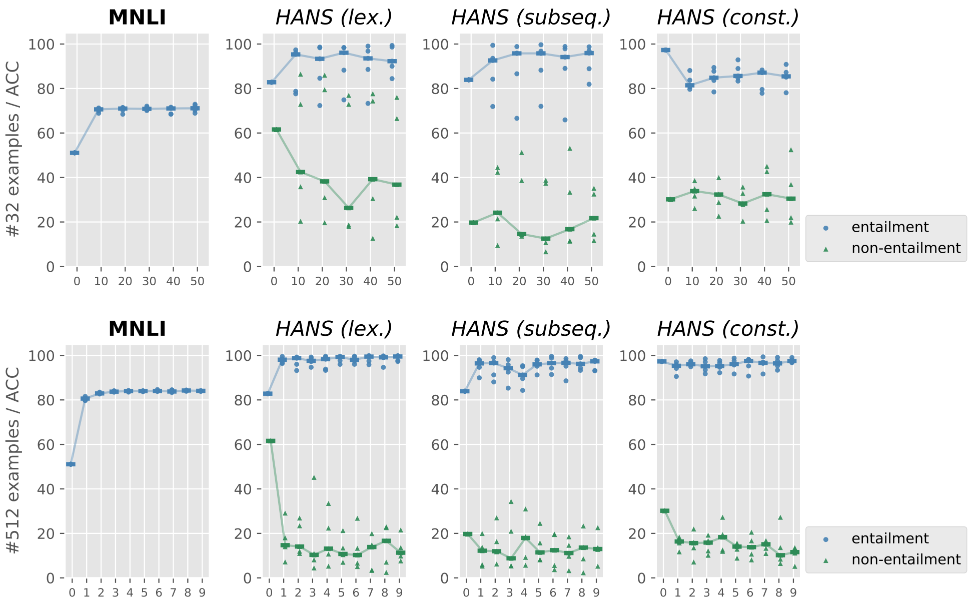


Figure 3: Results of prompt-based finetuning with varying number of epochs and fixed amount of training examples. Top: finetuning on 32 examples per label for epochs ranging from 10 to 50. Bottom: finetuning on 512 examples per label for 1 to 9 epochs. Both results show an immediate drop of non-entailment HANS performances which later stagnate even after more training steps.

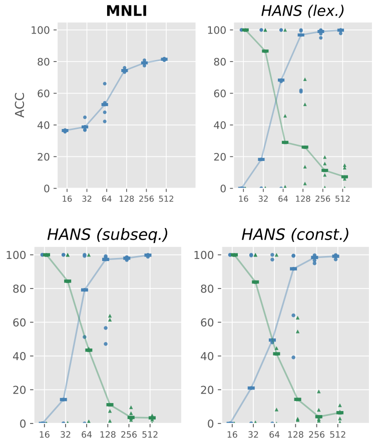


Figure 4: Results of non-prompt finetuning.

| | **MNLI** (acc.) | |
|---|---|---|
| | **IN** | HANS |
| manual | 51.1 | 62.6 |
| manual Ft-#512 | 84.3 | 54.8 |
| template-1 | 46.3 | 62.0 |
| template-1 Ft-#512 | 84.2 | 53.2 |
| template-2 | 49.9 | 61.3 |
| template-2 Ft-#512 | 83.9 | 52.7 |
| template-3 | 44.5 | 61.7 |
| template-3 Ft-#512 | 84.4 | 56.0 |

Table 5: Evaluation results of different MNLI templates provided by Gao et al. (2021). Models are evaluated against both the in-distribution (**IN**) set and corresponding challenge set of MNLI.

non-prompt finetuning with additional classifier head (CLS) can converge to that of prompt-based counterpart after certain amount of data, e.g., 512. It is then interesting to compare both finetuning paradigm in terms of their heuristics-related behavior. Figure 4 shows the results of standard finetuning using standard classifier head across varying data regimes on MNLI and the 3 subsets of HANS. We observe high instability of the results when only small amount of data is available (e.g., $\mathbf{K} = 64$). The learning trajectories are consistent across the HANS subsets, i.e., they start making random predictions on lower data regime and immediately adopt heuristics by predicting almost all examples exhibiting lexical overlap as **entailment**. We observe that standard prompt-based finetuning still performs better than CLS finetuning, indicating that prompt-based approach provides good initialization to mitigate heuristics, and employing regularization during finetuning can improve the challenge datasets (out-of-distribution) performance further.

**Impact of prompt templates** A growing number of work propose varying prompt generation strategies to push be benefits of prompt-based predictions (Gao et al., 2021; Schick et al., 2020). We

| | **MNLI** (acc.) | | | **QQP** (F1) | | | **SNLI** (acc.) | | |
|---|---|---|---|---|---|---|---|---|---|
| | **In.** | *HANS* | **avg.** | **In.** | *PAWS* | **avg.** | **In.** | *Scramble* | **avg.** |
| zero-shot `RoBERTa-large` | 51.1 | 62.6 | 56.8 | 35.4 | 51.8 | 43.6 | 49.7 | 64.7 | 57.2 |
| FT #512 `RoBERTa-large` | 84.3 | 54.8 | 69.5 | 82.1 | 29.6 | 55.8 | 88.1 | 50.1 | 69.1 |
| zero-shot `RoBERTa-base` | 48.2 | 58.1 | 53.15 | 37.3 | 41.5 | 39.4 | 48.8 | 56.4 | 52.6 |
| FT #512 `RoBERTa-base` | 74.4 | 49.9 | 62.15 | 79.0 | 26.9 | 52.9 | 83.7 | 48.5 | 66.1 |
| zero-shot `BERT-large-uncased` | 45.3 | 55.4 | 50.4 | 34.7 | 33.4 | 34.0 | 41.5 | 54.8 | 48.1 |
| FT #512 `BERT-large-uncased` | 70.9 | 50.0 | 60.4 | 77.3 | 26.3 | 51.8 | 79.9 | 49.5 | 64.7 |
| zero-shot `BERT-base-uncased` | 43.5 | 55.9 | 49.7 | 40.7 | 50.8 | 45.8 | 38.7 | 49.9 | 44.3 |
| FT #512 `BERT-base-uncased` | 63.2 | 50.1 | 56.65 | 73.9 | 29.1 | 51.5 | 74.5 | 42.6 | 58.5 |

Table 6: Evaluation results of different pretrained language models. Models are evaluated against both the in-distribution (**In.**) set and corresponding challenge set.

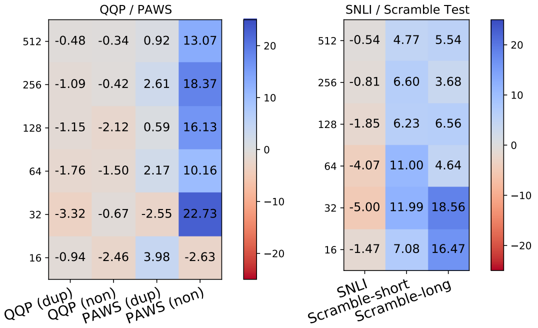


Figure 5: Relative difference between median accuracy of prompt-based finetuning across data regimes (y axis) with and without regularization on QQP / PAWS and SNLI / Scramble Test.

therefore questions whether different choices of templates would affect the model's behavior related to lexical overlap. We evaluate the 3 top-performing templates for MNLI that are obtained automatically by Gao et al. (2021) and show the results in Table 5. We observe similar behavior from the resulting models over the manual prompt counterpart, achieving HANS average accuracy of around 62% and below 55% on zero-shot and fine-tuning with 512 examples.

**Impact of learning steps** We investigate the degradation of the challenge datasets performance as the function of the number of training data available during finetuning. However, adding more training examples while fixing the number of epochs introduces a confound factor to our finding, which is the number of learning steps to the model's weights. To factor out the number of steps, we perform similar evaluation with a fixed amount of training data and varying number of training epochs. On 32 examples per label, we finetune for 10, 20, 30, 40, and 50 epochs. Additionally, we finetune on 512 examples for 1 until 10 epochs to see if the difference in learning steps results in different behavior. We plot the results in Figure 3. We observe that both finetuning settings result in similar trajectories, i.e., models start to adopt heuristics immediately in early epochs and later stagnate even with increasing number of learning steps. For instance, finetuning on 32 examples for the same number of training steps as in 512 examples finetuning for 1 epoch still result in higher overall HANS performance. We conclude that the number of finetuning data plays a more significant role over the number of training steps. Intuitively, larger training data is more likely to contain more examples that disproportionately *support* the heuristics; e.g. NLI pairs with lexical overlap are rarely of non-entailment relation (McCoy et al., 2019).

**Regularization across data regimes** Figure 5 shows the results improvement of L2 weight regularization over vanilla prompt-based finetuning on QQP and SNLI. Similar to results in MNLI/HANS, the improvements are highest on mid data regimes, e.g., 32 examples per label.

**Impact of pretrained model** In addition to evaluating `RoBERTa-large`, we also evaluate on other commonly used pretrained language models based on transformers such as `RoBERTa-base`, `BERT-base-uncased`, and `BERT-large-uncased`. The results are shown in Table 6. We observe similar pattern across PLMs, i.e., improved in-distribution scores come at the cost of the degradation in the corresponding challenge datasets.

# Chapter 8

# Data Augmentation for Robust Downstream Applications

# Falsesum: Generating Document-level NLI Examples for Recognizing Factual Inconsistency in Summarization

**Prasetya Ajie Utama**[†◇] **Joshua Bambrick**[†] **Nafise Sadat Moosavi**[‡◇] **Iryna Gurevych**[◇]

[†] Bloomberg, London, United Kingdom
[◇] UKP Lab, Technical University of Darmstadt, Germany
[‡] Department of Computer Science, The University of Sheffield
{putama,jbambrick7}@bloomberg.net

## Abstract

Neural abstractive summarization models are prone to generate summaries which are factually inconsistent with their source documents. Previous work has introduced the task of recognizing such factual inconsistency as a downstream application of natural language inference (NLI). However, state-of-the-art NLI models perform poorly in this context due to their inability to generalize to the target task. In this work, we show that NLI models can be effective for this task when the training data is augmented with high-quality task-oriented examples. We introduce Falsesum, a data generation pipeline leveraging a controllable text generation model to perturb human-annotated summaries, introducing varying types of factual inconsistencies. Unlike previously introduced document-level NLI datasets, our generated dataset contains examples that are diverse and inconsistent yet plausible. We show that models trained on a Falsesum-augmented NLI dataset improve the state-of-the-art performance across four benchmarks for detecting factual inconsistency in summarization.[1]

## 1 Introduction

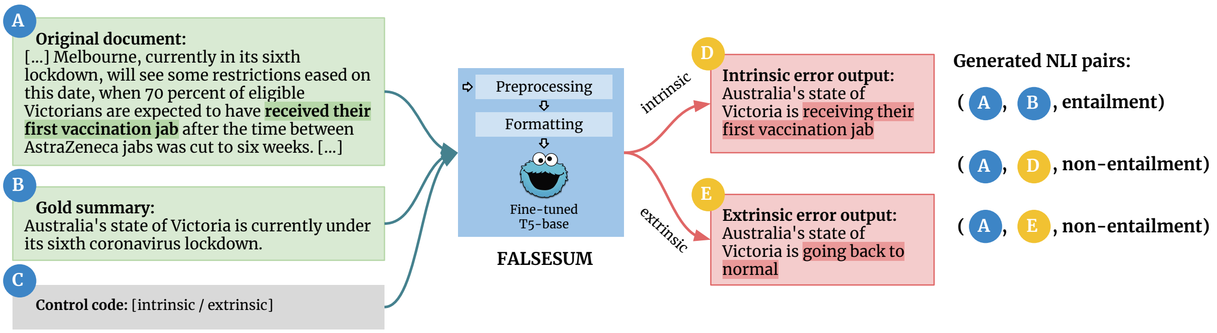
Recent advances in conditional text generation and the availability of large-scale datasets have given rise to models which generate highly fluent abstractive summaries (Lewis et al., 2019; Zhang et al., 2019). However, studies indicate that such models are susceptible to generating factually inconsistent outputs, i.e., where the content of the summary is not semantically entailed by the input document (Kryscinski et al., 2019; Goodrich et al., 2019). This motivates a new line of research for recognizing factual inconsistency in generated summaries (Kryscinski et al., 2020; Pagnoni et al., 2021; Wang et al., 2020; Fabbri et al., 2021).

This factual consistency problem is closely related to the task of natural language inference (NLI) whereby a **hypothesis** sentence is classified as either entailed, neutral, or contradicted by a given **premise** sentence (Condoravdi et al., 2003; Dagan et al., 2006; Bowman et al., 2015). Using an input document as the premise and a corresponding generated summary as the hypothesis, earlier solutions have adopted out-of-the-box NLI models to detect factual inconsistency, albeit with limited success (Falke et al., 2019; Kryscinski et al., 2020).

This poor performance largely stems from the fact that most NLI datasets are not designed to reflect the input characteristics of downstream tasks (Khot et al., 2018). Such datasets may not always capture the kinds of entailment phenomena which naturally arise from neural abstractive summarization. More importantly, there is also a discrepancy in terms of the input granularity, i.e., the premises in this consistency classification task consist of multi-sentence documents while common NLI datasets use single-sentence premises.

In this work, we introduce Falsesum, a data generation pipeline that produces NLI examples consisting of documents paired with gold summaries as **positive** examples and automatically generated inconsistent summaries as **negative** examples. We propose a novel strategy to train a text generation model to render false summaries of a given document using only supervision from an existing summarization dataset (Nallapati et al., 2016). In addition, our generator supports switchable input control codes to determine the type of factual error exhibited in the generated output. This design allows Falsesum to compose diverse and naturalistic outputs which more closely resemble the inconsistent summaries generated by summarization models (Maynez et al., 2020). This contrasts with previous solutions (e.g., Kryscinski et al., 2020; Yin et al., 2021), which synthesize NLI examples using rule-based transformations

[1] The code to obtain the dataset is available online at https://github.com/joshbambrick/Falsesum



**Original document:** [...] Melbourne, currently in its sixth lockdown, will see some restrictions eased on this date, when 70 percent of eligible Victorians are expected to have **received their first vaccination jab** after the time between AstraZeneca jabs was cut to six weeks. [...]

**Gold summary:** Australia's state of Victoria is currently under its sixth coronavirus lockdown.

**Control code:** [intrinsic / extrinsic]

**Intrinsic error output:** Australia's state of Victoria is receiving their first vaccination jab

**Extrinsic error output:** Australia's state of Victoria is going back to normal

Figure 1: Overview of the FALSESUM generation framework. FALSESUM preprocesses and formats the source document (A) and a gold summary (B) before feeding it to a fine-tuned generator model. The model produces a factually inconsistent summary, which can then be used to obtain (A, D) **or** (A, E) as the negative (non-entailment) NLI premise-hypothesis example pair. We also use the original (A, B) as a positive NLI example (entailment).

or language model-based replacements, limiting their diversity and ability to reflect realistic factual errors in summarization. Overall, our contributions in this paper are the following:

First, we present a novel training pipeline to create a text generation model which takes as input a pair of a document and a corresponding gold summary. It then perturbs the summary such that it is no longer factually consistent with the original document. Our strategy obviates the need for explicit examples of inconsistent summaries, using only an existing summarization dataset. We use this model to generate a large-scale NLI dataset for the task of recognizing factually inconsistent summaries. The resultant dataset consists of pairs with documents as the premise and naturalistic summaries as the hypotheses, each labeled as either **entailment** or **non-entailment**.

Second, we demonstrate the utility of our generated data for augmenting existing NLI datasets. We show that on four benchmark datasets, NLI models trained on FALSESUM-augmented data outperform those trained on previous document-level NLI datasets. We conduct an analysis to show that FALSESUM-generated summaries are plausible and hard to distinguish from human-written summaries. Lastly, we show that the improvement over the benchmarks is largely attributable to the diversity of factual errors that FALSESUM introduces.

## 2 Related Work

This work is related to the growing body of research into factual consistency and hallucination in text generation models, particularly for summarization (Cao et al., 2018). Research has found that around 30% of summaries generated by abstractive summarization models contain information which is inconsistent with the source document (Kryscinski et al., 2019). This motivates the development of an automatic approach to assess factual consistency in generated summaries, in addition to the benchmark datasets to measure the progress in this task (Falke et al., 2019; Kryscinski et al., 2020; Pagnoni et al., 2021; Fabbri et al., 2021).

Earlier work by Goodrich et al. (2019) proposes to use an information extraction model to extract relation tuples from the ground-truth summary text and the generated summary and then count the overlap as the measure of factuality. Eyal et al. (2019); Durmus et al. (2020); Wang et al. (2020) use a question-answering model to detect factual inconsistency by matching the predicted answers using the document and the summary as the context.

Concurrently, researchers have drawn a connection between factual consistency and natural language inference (NLI), observing that all information in a summary should be **entailed** by the source document. While this approach enables the summary to be directly evaluated without first extracting its intermediate semantic structure, earlier attempts were largely unsuccessful. Falke et al. (2019) use the probabilities assigned to the entailment label by NLI models to re-rank the summary candidates given by beam search but found no improvement in the consistency errors. Kryscinski et al. (2020) evaluate out-of-the-box NLI models on the task of inconsistency detection in a binary classification setting and show that the performance is only slightly better than majority voting.

In the same paper, Kryscinski et al. (2020) propose FactCC, a synthetic NLI data generation process which applies a set of transformation rules to obtain examples of inconsistent summaries (e.g., sentence negation, entity swapping). They demonstrate that the resulting NLI model performs well on realistic test cases which are obtained by manually annotating the output of several summarization models. This highlights the importance of NLI examples beyond sentence-level granularity and which more closely resemble the input characteristics of the downstream tasks (Mishra et al., 2021).[2]

While the FactCC model is moderately effective for detecting factual inconsistency, subsequent work indicates that it only performs well on easier test cases, where highly extractive summaries (i.e., those with high lexical overlap between a summary and the source document) tend to be factually consistent and more abstractive summaries are likely to be inconsistent (Zhang et al., 2020). Furthermore, Goyal and Durrett (2021) show that the synthetic and rule-based nature of FactCC leads to lack of diversity of consistency error types and it poorly aligns with the error distribution found in more abstractive summaries.

Falsesum addresses these limitations using controlled natural language generation to construct an NLI dataset which better targets the summarization domain. Inspired by the recent work on controllable generation (Keskar et al., 2019; Ross et al., 2021), we employ a generation model conditioned on an input code which controls the type of consistency errors induced. We further use the generated document-level NLI examples for augmentation and show that NLI models can benefit from the additional data without hurting their existing inference ability (Min et al., 2020).

## 3 Falsesum Approach

### 3.1 Design Overview

Falsesum takes as an input a source document D and a corresponding reference summary $\mathrm{S}^+$. The framework then **preprocesses** and **formats** D and $\mathrm{S}^+$ and feeds them into a generation model $\mathcal{G}$ which outputs a factually inconsistent summary $\mathrm{S}^-$. For each summarization example, we then have both positive **(entailment)** and negative **(non-entailment)** NLI tuples $(\mathrm{D}, \mathrm{S}^+, Y = 1)$, $(\mathrm{D}, \mathrm{S}^-, Y = 0)$, which consist of a document-level premise, a summary sentence, and the consistency label (1 indicates entailment).

Falsesum aims to produce a naturalistic $\mathrm{S}^-$ which is contrastive with respect to its corresponding $\mathrm{S}^+$. This means that $\mathrm{S}^+$ and $\mathrm{S}^-$ should be indistinguishable in their surface characteristics (e.g., style, length, vocabularies) and only differ in their factual consistency with respect to D. This ensures that the resulting NLI model learns the correct notion of factual consistency rather than discriminating based on surface features (McCoy et al., 2019). In addition to naturalness, we consider the diversity of the consistency error types exhibited by $\mathrm{S}^-$. We follow the **consistency error typology** introduced by Maynez et al. (2020), which categorizes consistency errors as either **intrinsic**, i.e., errors due to incorrect consolidation of information from the source document, or **extrinsic**, i.e., errors due to assuming *new* information not directly inferable from the contents of the source document.

As illustrated in Figure 1, a generation model $\mathcal{G}$ is trained to imitate the consistency mistakes of summarization models. Specifically, it generates perturbed summaries by either **(1)** incorrectly inserting pieces of information from the source document into random spans of the original summary; or **(2)** amending pieces of information in the summary by hallucinating new "facts" not present in the source document.

To this end, the framework identifies ***(◇i)* what** information or "facts" in the source document are available to the generator; and ***(◇ii)* where** the incorrect information can be inserted into the gold summary, which is indicated by span **masking**. We obtain both by subsequently performing **input preprocessing** and **formatting** steps (§3.2 and §3.3).

Next, we define the following seq2seq task to train the model $\mathcal{G}$: "Given ***(◇i)*** a list of **shuffled** and **formatted** pieces of information extracted from source document and gold summary and ***(◇ii)*** a partially **masked** gold summary, fill in the blanks and generate the original gold summary." Note that using gold summaries means that we can apply the existing summarization corpus to train $\mathcal{G}$ to generate more coherent and plausible sentences.

### 3.2 Input Preprocessing

Following Goodrich et al. (2019), "facts" in the source document and the gold summary are de-

[2] Contemporaneous work by Laban et al. (2022) attempts to improve the application of sentence-level NLI models to detect document-level factual inconsistencies using a learnable aggregation of sentence-level predictions. Our work is orthogonal since they can benefit from better quality training examples to train their aggregation weights.

fined as an open information extraction (OpenIE) tuple, which represents the predicate and argument structures found in a sentence. We denote each relation tuple as ($\text{ARG}_0, \text{PRED}, \ldots, \text{ARG}_n$), where predicate PRED describes the event (**what** happened) and its complementing semantic arguments ARG represent the **who**, **to whom**, **where**, or **how** of the event. Predicates are usually the main verb of a clause. Both predicates and their arguments consist of spans of tokens (Fader et al., 2011).

We use an OpenIE implementation of PredPatt (White et al., 2016; Zhang et al., 2017), a pattern-based framework for predicate-arguments extraction.[3] As illustrated in the top half of Figure 2, we extract the relation tuples from each source document and its corresponding reference summaries. To minimize the risk of $\mathcal{G}$ inadvertently generating consistent summaries, we corrupt each extracted "fact" by removing one randomly chosen argument from each tuple. For instance, OpenIE may extract the following tuple from a sentence:

$$(\underset{\text{ARG}0}{\underline{\texttt{Jo}}}, \underset{\text{PRED}}{\underline{\mathbf{\texttt{plans to give}}}}, \underset{\text{ARG}1}{\underline{\texttt{Alex}}}, \underset{\text{ARG}2}{\underline{\texttt{apples}}})$$

We then randomly choose $\texttt{apples}_{\text{ARG}_2}$ to be removed from the tuple. We additionally lemmatize the dependency root word of each argument and predicate span, e.g., **plans to give** ⇒ **plan to give**. This forces the model to learn to correct for grammaticality by inflecting the spans when inserting them to the **masked** spans. Once all such spans are extracted and processed, they are **grouped** and **shuffled** into two lists (predicates and arguments).

### 3.3 Input Formatting

Let P = ($\texttt{PRED}_1, \ldots, \texttt{PRED}_n$) and A = ($\texttt{ARG}_1, \ldots, \texttt{ARG}_m$) be the unordered lists of extracted predicates and arguments from a source document D and the summary sentence $S^+$. Additionally, we assume a **masked** summary sentence M (described later), derived from $S^+$, and a control code variable $c \in \{\texttt{intrinsic}, \texttt{extrinsic}\}$. Generator $\mathcal{G}$ is trained to compute $p(S^+|P, A, M, c)$. As illustrated in the bottom half of Figure 2, we encode all the conditional variables into the following format:

$$\texttt{Predicates:}P;\texttt{Arguments:}A;\texttt{Code:}c;\texttt{Summary:}M$$

In the following, we describe the key steps in the input formatting process:

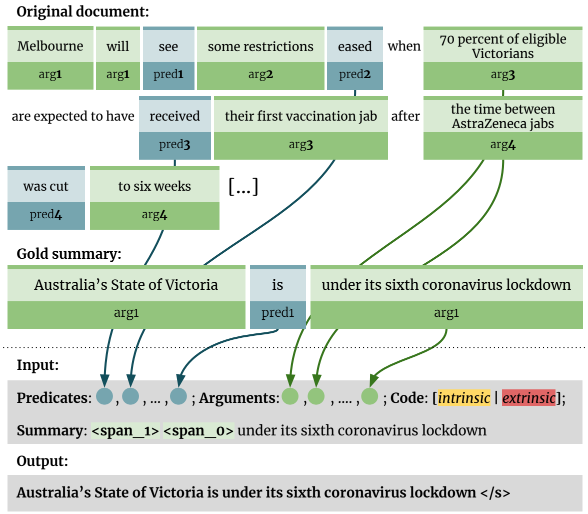

Original document:
Melbourne
arg1
will
arg1
see
pred1
some restrictions
arg2
eased
pred2
when
70 percent of eligible Victorians
arg3
are expected to have
received
pred3
their first vaccination jab
arg3
after
the time between AstraZeneca jabs
arg4
was cut
pred4
to six weeks
arg4
[...]
Gold summary:
Australia's State of Victoria
arg1
is
pred1
under its sixth coronavirus lockdown
arg1
Input:
Predicates: ● , ● , ... , ● ; Arguments: ● , ● , .... , ● ; Code: [intrinsic | extrinsic];
Summary: <span_1> <span_0> under its sixth coronavirus lockdown
Output:
Australia's State of Victoria is under its sixth coronavirus lockdown </s>

Figure 2: Input format design of FALSESUM. The framework first extracts the predicate and argument spans from the source document and the gold summary. The spans are then corrupted, lemmatized, and shuffled before being inserted into the input template.

**Step 1: Span Removal** Initially, P and A include predicate and argument spans from the original summary which may be used to reconstruct $S^+$. However, at **test** time we remove these "gold" spans from the two lists to force the $\mathcal{G}$ to make consistency mistakes. The removal is also done when training the model for control code `extrinsic` to train $\mathcal{G}$ to predict plausible unseen spans.[4] We summarize the different input formatting in Table 1.

**Step 2: Span Reduction** To encourage $\mathcal{G}$ to generate fine-grained errors (Pagnoni et al., 2021; Goyal and Durrett, 2021), we also train it to hallucinate incorrect modifiers into spans from P and A. To this end, we randomly drop adjectives and adverbs from 10% of the gold predicate and argument spans. For instance, an argument span "recently elected prime minister" will be reduced to "minister". This teaches the model to generate the remaining part of the span given only the context provided in the formatted input.

**Step 3: Control Code** To control the type of consistency errors generated by $\mathcal{G}$, we append the string "`code:`" followed by either "`intrinsic`" or "`extrinsic`" into the input tokens. The code is determined randomly with equal probability of 0.5.

[3] We note that the quality of the OpenIE extractions may impact the overall quality of our data generation framework.

[4] It is possible that some spans from the source document are duplicates of gold ones. For instance, the document may mention "The Queen of England", while the gold span from the summary is "The Queen". We use a simple heuristic to remove such duplicates by searching for other spans whose (lemmatized) dependency root token is the same.

| Mode | Input | Expected Output | Description |
|---|---|---|---|
| **train**<br>intrinsic | `Predicates` : caught, **plead guilty to**, . . . , appear before, face; `Arguments` : the corruption scandal, **Two Pennsylvania judges**, . . . , many children, the U.S. `Code` : intrinsic; `Summary` :`<span_1>` `<span_0>` federal fraud charges. | Two Pennsylvania judges plead guilty to federal fraud charges. | Model learns to combines listed spans to produce most plausible summary. |
| **test**<br>intrinsic | `Predicates` : caught, ~~plead guilty to~~, . . . , appear before, face; `Arguments` : the corruption scandal, ~~Two Pennsylvania judges~~, . . . , many children, the U.S. `Code` : intrinsic; `Summary` :`<span_1>` `<span_0>` federal fraud charges. | Many of the children face federal fraud charges. | Model consolidates incorrect information. |
| **train**<br>extrinsic | `Predicates` : ~~is pressing for~~, limit, . . . , is being erode, is fight; `Arguments` : panelist, ~~action~~, . . . , sea level, Arctic melt, ~~at the climate change conference~~ `Code` : extrinsic; `Summary` : The Alliance `<span_0>` `<span_1>` `<span_2>`. | The Alliance is pressing for action at the climate change conference. | Model learns to hallucinate new unsupported information. |
| **test**<br>extrinsic | `Predicates` : ~~is pressing for~~, limit, . . . , is being erode, is fight; `Arguments` : panelist, ~~action~~, . . . , sea level, Arctic melt, ~~at the climate change conference~~ `Code` : extrinsic; `Summary` : The Alliance `<span_0>` `<span_1>` `<span_2>`. | The Alliance is planning to impose limits on emissions. | Model hallucinates new unsupported information. |

Table 1: Examples of input formatting on two different summarization instances for both intrinsic and extrinsic error types during training and testing. Gold input spans (indicated by **boldface**), which are extracted from the gold summary, are only visible to the model during intrinsic training. They are removed from the input in all other settings, as indicated by ~~strikethrough~~ text.

Once the code is chosen, we perform the remaining formatting steps accordingly (see Table 1).

**Step 4: Summary Masking** We derive masked summary M by replacing the spans of **randomly** selected predicates and arguments with a special token **`<span_i>`**, where $i = 0$ is reserved for the predicate, and $i > 0$ for their arguments. These tokens control **where** the incorrect information should be inserted by the generator model into the original summary (see Table 1).

### 3.4 Training Falsesum

We run the Falsesum data generation pipeline on the *train* split of the CNN/DailyMail corpus (Hermann et al., 2015), originally collected for question answering, but subsequently reformulated for summarization by Nallapati et al. (2016). This dataset contains *English* news documents paired with human-written summaries, each consisting of multiple sentences. We break the summaries down such that each Falsesum example consists of the document text and a single sentence summary. We then run the **preprocessing** and **formatting** steps on each document-summary pair. The resulting pairs of formatted input and target output are subsequently split into train and test sets which consist of 394,774 and 262,692 instances, respectively.

We use the `T5-base` model (Raffel et al., 2020) as generator $\mathcal{G}$ and fine-tune it on the seq2seq task described in §3.1. The NLI examples are produced by running the fine-tuned generator on the preprocessed and formatted test split.[5] This renders an equal number of positive and negative examples. In our experiments, we randomly sample 100,000 Falsesum examples to augment the NLI dataset.

## 4 Experimental Settings

Our experiments aim to demonstrate the effectiveness of Falsesum-generated document-level examples for NLI dataset augmentation. We evaluate the downstream performance of the NLI models by testing them against several benchmarks for determining the factual inconsistency of generated summaries. In this section, we describe the training setup of the NLI models, including the model and both the sentence- and document-level datasets.

### 4.1 Training

**NLI models** We train several NLI models by fine-tuning `RoBERTa-base` (Liu et al., 2019) on *either* the original or the augmented MNLI dataset (Williams et al., 2018). The MNLI dataset consists of 392,702 train instances, each labeled

[5] See Appendix A for the hyperparameter details.

as either "*entailment*", "*neutral*", or "*contradiction*". To enable the application of NLI data to this factual consistency task, we use a binary formulation of NLI, where the *"neutral"* and *"contradiction"* labels are combined into *"non-entailment"*. The document-level inputs are formatted similarly to sentence-level examples, i.e., the document premise D and summary hypothesis ($S^+$ or $S^-$) are concatenated and a special classification token (`[CLS]`) is used (Devlin et al., 2019).

**Document-level NLI datasets** We conduct augmentation comparisons with several multi-sentence NLI datasets which obtain examples from *news* or *summarization* domains. We consider the following datasets: **ANLI** (Nie et al., 2020), a paragraph-level NLI dataset collected via an iterative and adversarial human-in-the-loop annotation protocol. It consists of mostly Wiki data but also includes a small portion of news text; **DocNLI** (Yin et al., 2021), a document-level NLI dataset containing multi-sentence premise and hypothesis sentences, collected by converting QA examples to NLI instances (Demszky et al., 2018) and replacing words and sentences in *news* summaries using a language model; **FactCC** (Kryscinski et al., 2020), a large-scale dataset specifically generated for training summary factual correctness classification models. The positive examples in FactCC are obtained by backtranslating a random sentence from a CNN/DailyMail *news* story, while negative examples are obtained by perturbing the sentence using predefined rules, e.g., entity swapping. For fair comparison, we sample 100,000 examples from each augmentation dataset in our experiments.

### 4.2 Benchmark Datasets

We evaluate these NLI models on four benchmark datasets to classify the factual consistency of abstractive summaries. These datasets differ in terms of the annotation protocol, the granularity of the summaries (single- or multi-sentence), the summarization corpus used, and the models used to generate the summaries that are annotated. The tasks are formulated as a binary classification with the labels *"consistent"* and *"inconsistent"*. We evaluate NLI models on these tasks by mapping the predicted label *"entailment"* to *"consistent"* and *"non-entailment"* to *"inconsistent"*. The benchmarks datasets are detailed in the following:

**FactCC** In addition introducing a synthetic training dataset for the task, Kryscinski et al. (2020) introduce a manually annotated test set. It contains 1,431 document and single-sentence summary pairs generated by various neural abstractive summarization models trained on CNN/DailyMail corpus.[6]

**Ranksum** Falke et al. (2019) formulate the factual consistency problem in summarization as a ranking task. They introduce a dataset consisting of 107 documents, each paired with a set of five ranked summary candidates obtained from the beam search of a summarization model. Given the manually annotated consistency label on summary candidates, the task is to re-rank the list such that the top-1 summary is factually consistent.

**Summeval** Fabbri et al. (2021) introduce a comprehensive benchmark for factual consistency detection in summarization. It includes summaries generated by seven extractive models and sixteen abstractive models, which are judged by three annotators using a 5-point Likert scale.[7]

**QAGS** The dataset collected by Wang et al. (2020) consists of 239 test set instances from XSUM (Narayan et al., 2018) and 714 instances from CNN/DailyMail.[8] Each instance consists of a pair of a source document and a single-sentence summary, which is labeled via majority voting on three annotators' labels.

## 5 Results and Discussion

### 5.1 Main Results

Performance on FactCC, QAGS, and SummEval is measured using balanced accuracy, which is suitable for class imbalanced settings, since the factually consistent label is the majority in some benchmark datasets. It is defined as the average recall of the two classes, such that majority label voting obtains only a 50% score. To measure ranking performance in Ranksum, we calculate the average Precision@1, which computes the fraction of times a factually consistent summary is ranked highest on each test instance. We perform five training runs for each setup using different random seeds and take the mean to address performance instability (Reimers and Gurevych, 2017).

[6] We merge the test and validation sets into a single test set.

[7] We aggregate the label as "consistent" if all annotators rated the summary as a 5 and "inconsistent" otherwise.

[8] This is the number of instances after we split multi-sentence summaries into separate single-sentence summary test instances, where an individual factuality judgement is available.

| Dataset | Augmentation | Benchmark Datasets<br>FactCC | Ranksum | QAGS | SummEval | Overall |
|---|---|---|---|---|---|---|
| *Majority voting* | - | 50.00 | 50.46 | 50.00 | 50.00 | 50.11 |
| **MNLI**-128 | - | 57.39 | 57.01 | 59.72 | 54.11 | 57.06 |
| `[split-doc]` **MNLI**-128 | - | 72.07 | 68.03 | 71.08 | 55.32 | 66.63 |
| **MNLI**-512 | - | 57.93 | 51.40 | 52.73 | 48.75 | 51.43 |
| **MNLI**-512 | `ANLI` | 53.91 | 55.76 | 53.54 | 49.56 | 53.19 |
| **MNLI**-512 | `DocNLI` | 58.13 | 53.58 | 57.10 | 52.59 | 55.35 |
| **MNLI**-512 | `FactCC` | 73.87 | 67.29 | 73.50 | 60.04 | 69.02 |
| **MNLI**-512 | FALSESUM (ours) | **83.52** | **72.90** | **75.05** | **65.18** | **74.17** |

Table 2: Performance of MNLI models with different augmentation data across benchmarks to classify the factual consistency of summaries. **MNLI**-128 and **MNLI**-512 are `RoBERTa-base` models trained using maximum token length of 128 and 512, respectively.

| Training Dataset | Overall | Δ |
|---|---|---|
| MNLI+FALSESUM | **74.17** | |
| MNLI+FALSESUM `-Contrastive` | 73.11 | -1.06 |
| MNLI+FALSESUM `-Extrinsic` | 71.95 | -2.22 |
| MNLI+FALSESUM `-Intrinsic` | 69.14 | **-5.03** |

Table 3: Model performance when trained on ablated FALSESUM dataset. Excluding the contrastive, extrinsic, and intrinsic examples results in lower overall performance, indicating each property is beneficial.

From the results in Table 2, we observe the following: **(1)** Models trained on sentence-level MNLI datasets perform poorly when evaluated directly on document-level benchmarks, even after we increase the maximum input token length from 128 to 512;[9] **(2)** This limitation can be alleviated by the sentence-wise prediction strategy (`[split-doc]`MNLI-128),[10] which achieves 66.63. Note, however, that this improvement comes at the expense of compute cost which is multiplied by a significant factor; **(3)** DocNLI and ANLI perform poorly even though they contain longer premise sentences, indicating that the length mismatch may not be the primary issue; **(4)** FALSESUM obtains substantial improvement over the previous state-of-the-art FactCC, despite being derived from the same summarization dataset (CNN/DailyMail). This indicates that FALSESUM provides higher quality examples and includes more types of entailment phenomena that occur naturally in this task.

### 5.2 Ablation Analysis on FALSESUM Data

We perform an ablation analysis to study how each component of our data generation pipeline contributes to the final performance. We first remove the contrastive property of the FALSESUM data by randomly including only **either** the positive $(\mathrm{D}, \mathrm{S}^+, Y = 1)$ **or** negative $(\mathrm{D}, \mathrm{S}^-, Y = 0)$ NLI examples obtained from a single $(\mathrm{D}, \mathrm{S}^+)$ pair. Next, we filter out the negative NLI instances that are generated using either `intrinsic` or `extrinsic` code. We refer to the three ablated datasets as `−contrastive`, `−intrinsic` and `−extrinsic`, respectively. We set the sampled training size to 100,000 for the three ablation setups and aggregate the results from five training runs.

Table 3 shows the performance of the ablated models. We observe that removing contrastive pairs in the augmented training data results in a 1.06% drop on the overall benchmarks score. We also see that removing `intrinsic` error examples results in the highest performance loss, −5.03% compared to −2.22% by `−extrinsic`. This is explained by the fact that intrinsic consistency errors are more dominant on benchmarks that are built on the CNN/DailyMail corpus (Goyal and Durrett, 2021). We conclude that all the above properties are important for the overall improvements obtained by FALSESUM.

### 5.3 Fine-grained Evaluation

Previous work has shown that NLI models are prone to relying on fallible heuristics which associate lexical overlap with entailment labels (McCoy et al., 2019). In the factual consistency task, this corresponds to models associating highly extractive summaries with the "consistent" label. This raises a question about whether FALSESUM data alleviates this tendency in the resulting NLI models.

To answer this question, we partition the FactCC annotated test examples into five ordered subsets based on the lexical overlap between their

[9] Average context word count is only 22 in MNLI and 546 in FactCC.

[10] See details in Appendix B

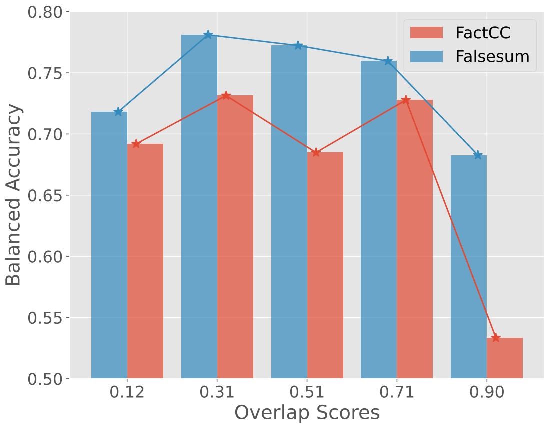


Figure 3: Comparison between NLI models augmented with Falsesum and FactCC across different measures of summary extractiveness. The x-axis shows the median overlap score of each test subset.

summary hypothesis and the source document premise. We define an overlap score using the normalized coverage and density summary extractiveness scores introduced by Grusky et al. (2018). Both measures have the range $[0.0, 1.0]$, where density $= 1.0$ indicates that all words in a summary are also present in the source document and normalized coverage $= 1.0$ indicates that the summary is obtained by copying a continuous fragment of the source document. We then define overlap $=$ normalized coverage $\times$ density.

Figure 3 shows the comparison of FactCC and Falsesum augmentation performance across varying lexical overlap scores. We see that Falsesum performs better on all subsets of the FactCC test set with the greatest performance gap appearing on the 0.9 overlap subset. Upon closer inspection, we see that the FactCC model makes mostly false positive classification errors on this subset, i.e., it tends to predict highly extractive summaries as "consistent", leading to near majority voting performance of 50%. Falsesum, on the other hand, better discriminates the factual consistency of examples without over-relying on lexical overlap.

### 5.4 Data Quality Analysis

We conduct both manual and automatic quality evaluation of the Falsesum-generated dataset. First, we sample 200 generated negative examples and manually verify whether (i) the perturbed summary $\mathrm{S}^{-}$ is indeed factually inconsistent; (ii) the type of consistency error follows the specified control code; (iii) the incorrect "fact" is inserted at the specified missing span. Following Kryscinski et al. (2020), the authors perform this annotation to avoid high disagreement by crowd annotators in this task (Falke et al., 2019). The results in Table 4 show that about 86% of intrinsic 81% of extrinsic generated error examples are factually inconsistent, which happen due to several reasons, e.g., generator model chooses a span from the list that is similar to the original span, or generator model correctly guesses the original missing span. This further suggests that pre-trained language models such as `RoBERTa-base` can be robust against the induced label noise and can still learn a performant classifier. While $\mathcal{G}$ almost always inserts the incorrect "fact" at the specified positions, we observe that it often fails to follow the specified extrinsic code correctly. We suspect that this is because the model prefers the easier task of copying the input over generating novel phrases.[11]

| Code | Label ✓ | Type ✓ | Span ✓ |
|---|---|---|---|
| Intrinsic | 86% | 94% | 94% |
| Extrinsic | 81% | 65% | 95% |

Table 4: Manual verification of Falsesum-generated NLI examples. Label, type, and span indicate the percentage of generated summaries with correct inconsistency label, error type, and error span, respectively.

| | FactCC | DocNLI | Falsesum |
|---|---|---|---|
| Majority voting | 50.84 | 53.55 | 50.00 |
| CBOW-GloVe | 60.36 | 70.38 | 56.13 |
| BiLSTM-GloVe | 68.26 | 73.04 | 57.62 |
| RoBERTa-base | 82.15 | 78.46 | 69.38 |

Table 5: Hypothesis-only model performance (accuracy) to measure the presence of artifacts and naturalness of Falsesum dataset (lower is better).

Following Gururangan et al. (2018), we also evaluate the naturalness of the generated dataset. We train an NLI model using positive examples from CNN/DailyMail and Falsesum-generated negative examples. The model receives no premise so must distinguish between entailed and non-entailed hypotheses using semantic plausibility or spurious surface features, e.g., grammatical mistakes or fluency errors. The relatively low accuracy of these models on Falsesum data (shown in Table 5) suggests that, compared to FactCC and DocNLI, Falsesum-generated summaries are relatively hard to distinguish from the gold ones.

[11] We include more examples of generated NLI instances as well as the inadvertently consistent output in Appendix D.

## Conclusion

NLI models present a promising solution for automatic assessment of factual consistency in summarization. However, the application of existing models for this task is hindered by several challenges, such as the mismatch of characteristics between their training dataset and the target task data. This mismatch includes the difference in terms of the input granularity (sentence vs. document level premises) and the types of (non-)entailment phenomena that must be recognized.

In this work, we present FALSESUM, a data generation pipeline which renders large-scale document-level NLI datasets without manual annotation. Using our training strategy, we demonstrate that it is possible to learn to generate diverse and naturalistic factually inconsistent (non-entailed) summaries using only existing (entailed) consistent summaries for training. We show that the resultant data is effective for augmenting NLI datasets to improve the state-of-the-art performance across four summary factual inconsistency benchmarks.

## Acknowledgments

We would like to thank Marco Ponza, Marco Fiscato, Umut Topkara and other colleagues from Bloomberg AI for the thoughtful discussion and feedback throughout this project. We also thank Leonardo Ribeiro for comments on the earlier version of this work and the anonymous reviewers for their constructive feedback. The authors affiliated with UKP were supported by the German Research Foundation through the research training group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES, GRK 1994/1) and by the German Federal Ministry of Education and Research and the Hessian State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

## References

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. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4784–4791. AAAI Press.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *CoRR*, abs/1809.02922.

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. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. 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 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 166–175, New York, NY, USA. Association for Computing Machinery.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. 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 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Yinhan Liu, 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. *CoRR*, abs/1907.11692.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Anshuman Mishra, Dhruvesh Patel, Aparna Vijayakumar, Xiang Lorraine Li, Pavan Kapanipathi, and Kartik Talamadupula. 2021. Looking beyond sentence-level natural language inference for question answering and text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1322–1336, Online. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gucehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Yuyang Nie, Yuanhe Tian, Yan Song, Xiang Ao, and Xiang Wan. 2020. Improving named entity recognition with attentive ensemble of syntactic information. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4231–4245, Online. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

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. Association for Computational Linguistics.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E. Peters, and Matt Gardner. 2021. Tailor: Generating and perturbing text with semantic controls. *CoRR*, abs/2107.07150.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

Sheng Zhang, Rachel Rudinger, and Benjamin Van Durme. 2017. An evaluation of PredPatt and open IE via stage 1 semantic role labeling. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*.

Yuhui Zhang, Yuhao Zhang, and Christopher D. Manning. 2020. A close examination of factual correctness evaluation in abstractive summarization.

## A Hyperparameters

**Generator model** We train a `T5-base` model for three epochs with batch size of 24 using the AdamW optimizer. We set the maximum source token length to 256 and the target token length to 42. We use a learning rate of $3e^{-5}$ and fix the random seed to 11. For decoding, we set the minimum and maximum sequence length to 10 and 60, respectively. We sample using beam search with a beam of size two. We additionally set the repetition penalty to 2.5 and the length penalty to 1.0.

**Classification model** We train `RoBERTa-base` models on augmented and original MNLI datasets for three epochs with a batch size of 32. The learning rate is set to $1e^{-5}$, while the maximum input token length is set to either 128 or 512. We use the following random seeds for the five training runs: 11, 12, 13, 14, and 15.

## B Aggregating Predictions

We follow Falke et al. (2019) to adapt out-of-the-box MNLI models to document-level input by performing a sentence-wise prediction before aggregating the output. Given a document D consisting of sentences $d_1, \ldots, d_n$, and a multi-sentence summary $S$ consisting of $s_1, \ldots, s_m$, we aggregate the probability scores given by the classifier model $F$ on each $d_i, s_j$ pair. The aggregated consistency score $\sigma(\mathrm{D}, S)$ is given by:

$$\sigma(\mathrm{D}, S) = \frac{1}{m} \sum_{j=1}^{m} \max_{d \in \mathrm{D}} F(d, s_j)$$

This means that it is sufficient for a summary sentence to be factually consistent given only a single entailing sentence in the source document. We then take the average scores across the summary sentences since each of them needs to be entailed by the source document. We use a similar aggregation method to evaluate augmented MNLI models on multi-sentence summaries from the Summeval and Ranksum benchmarks.

## C Falsesum Details

In the preprocessing steps, we only perform the predicate and argument span extraction on the first 15 sentences for computational efficiency. For training, this is not an issue since the gold spans from the reference summary are included in the input. Additionally, we may extract multiple OpenIE relation tuples from each sentence. To avoid having overlapping spans from a single input, we randomly select two tuples from each sentence.

## D Falsesum Examples

We include more examples of generated NLI instances in Table 6. We also include cases where Falsesum inadvertently generates factually consistent summaries in Table 7. Lastly, we show several examples of the formatted input and the generated output at **test** time in Table 8.

<table>
<tr><td colspan="2">Mexican federal police have arrested a fugitive on the FBI's 10 Most Wanted list, Mexican authorities said. Jorge Alberto Lopez Orozco allegedly murdered his girlfriend and her two young sons. Jorge Alberto Lopez Orozco is wanted in Elmore County, Idaho, on charges that he shot and killed three people, the FBI said. The charred remains of a woman and her sons, ages 2 and 4, were found inside a burned-out vehicle on August 11, 2002, it said. Each victim had been shot in the head or chest. The FBI was still working Friday to confirm the identity of the man in custody, said Debbie Dujanovic, a spokeswoman in the agency's Salt Lake City, Utah, field office. The Salt Lake City office has jurisdiction in the case. An extradition order was issued in January 2007, the Mexican attorney general's office said in a news release Thursday. A reward of up to $100,000 was being offered, the FBI said. Lopez, 33, was captured in Zihuatanejo, a city northwest of Acapulco on the Pacific Coast in southern Mexico, the Mexican attorney general's office said. Zihuatanejo is in Guerrero state, but Lopez was transferred to a jail in neighboring Michoacan state, officials said. The arrest came about after investigation and intelligence work by Mexican authorities, the attorney general's office said. According to the FBI, Lopez abducted his girlfriend, Rebecca Ramirez, and her two young sons from her father's house in Nyssa, Oregon, on July 30, 2002. The car he had been driving was found nearly two weeks later on a rural road near Mountain Home, Idaho, officials said. . . .</td></tr>
<tr><td>entailment<br>(intrinsic) non-entailment</td><td>FBI was still working Friday to confirm the identity of the man in custody.<br>An extradition order was issued in July 30, 2002, to determine the identity of the man in custody.</td></tr>
<tr><td colspan="2">He may have been allowed to leave the club without ever playing a league game for the first team, but Kristoffer Olsson still showed Arsenal some love as he departed. The 19-year-old Swede, whose only first-team appearance for the Gunners came off the bench in the Capital One Cup last season, has joined FC Midtjylland this week on a permanent deal. But, as the news was announced, Olsson took to Twitter to say 'Once a Gunner, always a Gunner'. Kristoffer Olsson (right) played just once for Arsenal's first team, in the Capital One cup against West Brom . Olsson expressed his love for the club on Twitter, despite being sold to FC Midtjylland . The tweet reflects Cesc Fabregas' comments when he left the club to join Barcelona, although the Spanish midfielder has sinced joined rivals Chelsea, after Arsene Wenger opted not to buy him back. Olsson has been on loan at FC Midtjylland since the beginning of the season, playing six times in the Danish top flight. The Sweden U21 international said on joining permanently: 'this is a club that believes in me and sees my potential.' Olsson has played six times on loan with FC Midtjylland and has now joined the Danish club permanently.</td></tr>
<tr><td>entailment<br>(intrinsic) non-entailment</td><td>Swedish international takes to social media to express love for Arsenal.<br>Swedish international has been on loan at Chelsea since last season.</td></tr>
<tr><td colspan="2">A teenager who was struck down with an agonising bowel condition says dancing has helped him to overcome his debilitating illness. Macaulay Selwood, 17, was diagnosed with Crohn's two years ago and was so unwell that he was often left in agony on the floor unable to move. But his determination to continue his promising dancing career gave him the spur he needed to battle through. Lord of the Dance: Macaulay at his practice studio. He was diagnosed with Crohn's in September 2010 after collapsing in agony during a dance class . Recovery: 'Dancing has helped me overcome it (Crohn's). It kept me motivated' Now the teenager from Bristol has made it to the finals of the Irish dancing world championships in Boston, USA, and is hotly-tipped for glory. He will then have a trial at the famous performing arts school, ArtsEd, in London. At shows he has been compared with Riverdance star Michael Flatley while others have taken to calling him Billy Elliot, after the film character who overcomes the odd to becoming a dancing star. Macaulay did ballet at college before focusing on Irish dancing for the world championships and works at Tesco to fund his passion. . . .</td></tr>
<tr><td>entailment<br>(extrinsic) non-entailment</td><td>Macaulay Selwood, 17, first starting suffering from Crohn's disease in 2010.<br>The 22-year-old, who was diagnosed with Crohn's in 2010, has been recovering since 2010.</td></tr>
<tr><td colspan="2">When Matthew Briggs, 32, from Huntington in North Yorkshire noticed that his father had posted a photo of them together on Facebook, he was initially pleased. But when he opened the photo and saw the image, Mr Briggs was left horrified by the sight of his 31st frame. Now, two years on, he has shed an astonishing 17st and, in November, will complete the New York marathon in memory of his mother Susan who died from multiple sclerosis when he was just 18. Pounding the pavements: Matthew Briggs, 32, has lost an impressive 17st in just two years of slimming . 'In March of 2000, she lost her battle with Multiple Sclerosis,' he says. 'She has always been my inspiration. I am the man I am today because of the woman she was.' Money raised by Mr Briggs' 26-mile run will be donated to the Multiple Sclerosis Society, a charity dedicated to beating the disease as well as supporting sufferers and their families. Mr Briggs, who has dropped from 31st to just under 14st, had piled on the pounds thanks to a diet of ready meals, takeaways and daily two litre bottles of Coca-Cola. But, after seeing the photo posted on Facebook and spurred on by a bet with his father, Mr Briggs joined his local Slimming World group and went on to shed more than 17st over two years. . . .</td></tr>
<tr><td>entailment<br>(extrinsic) non-entailment</td><td>She died in 2000 of multiple sclerosis and funds raised will go to charity.<br>She died in 2000 of multiple sclerosis and every penny she saves will go to charity.</td></tr>
</table>

Table 6: Examples of NLI pairs generated by FALSESUM. We show both the entailment and non-entailment hypotheses obtained from each source document. Green-highlighted spans indicate the information used consistently in the summary. Red-highlighted spans indicate information used or inserted by the model to generate an inconsistent summary.

| | |
|---|---|
| The Mojito, a Cuban mix of white rum, sugar, lime, mint and soda water, is the most popular cocktail in Britain according to a report . Sales of cocktails have risen by more than 10 per cent in the past two years. More than one in five of Britain's pubs and bars now serve cocktails and the Mojito – a Cuban mix of white rum, sugar, lime, mint and soda water – is the most popular, according to a report. Pina Coladas (rum, coconut and pineapple juice) and Woo Woos (vodka, peach schnapps and cranberry juice) were also popular. The Mixed Drinks Report, by consultancy firm CGA Strategy, found more women than men choose cocktails, as 54 per cent of cocktail drinkers are female. Bomb and pitcher serves remain popular, with 74 per cent of 18 to 24-year-olds admitting to have bought a bomb drink, while nine in 10 in the same age range say they drink pitchers. Cocktails are enjoyed by the core 18 to 35-year-old demographic 'in all on-trade occasions' including throughout the night, as opposed to just the start. . . . | |
| gold | Sales of cocktails have risen by more than 10 per cent in the past two years. |
| (extrinsic) generated | Cocktails have soared in popularity over the past two years. |
| From Yellowstone National Park to the Everglades, America's 391 national parks are in need of repair – and thanks to the economic stimulus signed into law, help is now underway. President Obama and his family visit the Grand Canyon in Arizona, a national park. President Obama's $787 billion economic stimulus plan passed in February and designated $750 million dollars to the national parks. But not all of the stimulus money is being used – and the parks are facing a $9 billion backlog in maintenance projects. So far, nearly 10 percent is in the pipeline. "We are picking away at it as much as we can and we've been fortunate to have the recovery act money," said Jeffrey Olson of the National Park Service. Olson said half of the $9 billion is slated to go for road repairs. "Half of that [$9 billion] is roads and about $2 billion of that are the most pressing needs – those we get some help from the stimulus. The president's budget proposal is calling for more maintenance and construction money," Olsen said. Dan Wenk, the acting director of the National Park Service says most of those pressing needs include, "camp grounds, camp sites, it's amphitheaters for evening programs. It's the bathrooms. . . . | |
| gold | Park Service is dealing with a $9 billion backlog of maintenance needs. |
| (intrinsic) generated | America's 391 national parks are facing a $9 billion backlog of maintenance needs. |

Table 7: FALSESUM-generated summaries that are unintentionally consistent with the source document. Green-highlighted spans indicate information which is consistent with the document.

| | |
|---|---|
| `Predicates` : is being offer for, were steal from, sell, Both as a solo artist and leader of the Heartbreakers, is one of , according to, where were rehearse for, contribute to, was induct into in; `Arguments` : the Heartbreakers, The band, Denise Quan, five guitars, the Recording Industry Association of America, more than 57 million albums, Petty, A 7,500 reward, a soundstage, the Rock & Roll Hall of Fame; `Code` : intrinsic; `Summary` :<span_1> <span_0> the 1960s. | |
| gold | Three of them were vintage guitars from the 1960s. |
| (intrinsic) generated | The band was inducted into the Rock & Roll Hall of Fame in the 1960s. |
| `Predicates` : : is only the second time in, How could have do with, was lace with, struggle against at, have score, expect to match, had settle into, ignite, has lost, Just as was walk into, were already circulate on, begin to filter, watch on in; `Arguments` : his chair, Anfield, clips, the stands, symbolism, 13 Premier League goals, Brendan Rodgers, through, Liverpool, the 100-plus strikes of last season, 13 games against Hull, everything, one; `Code` : intrinsic; `Summary` :Luis Suarez took three minutes to <span_0> <span_1>. | |
| gold | Luis Suarez took three minutes to get his first assist for Barcelona. |
| (intrinsic) generated | Luis Suarez took three minutes to ignite symbolism. |
| `Predicates` : allegedly know, supposedly write, in ' was underway, is investigate, file against in by, file in, forbid, was toss by in, wait for, fire at, accuse of, decide to fire based on, new information state, told, allegedly sent to, was complicate by, Even though was toss, allegedly made, hold no more, expose to; `Arguments` : the case, new information states, his sexual abuse, more recent damages, people, the blog posts, 2011, him, This week, her, allowing at one of his Los Angeles stores to post naked photos of Morales on a blog that was meant to appear as though it belonged to Morales, American Apparel, The Post, a settlement, The clothing company, Charney, new information saying he allowed an employee to impersonate and post naked photos online of an alleged victim of his sexual abuse who filed a case against him in 2011, a settlement 'in the low six-digits' was underway, the company title, employee, 2012, The $260 million lawsuit, a report from March 25, 2011 that said Morales allegedly sent nude photos of herself to Charney after she stopped working at the store, nude photos of herself, Morales; `Code` : extrinsic; `Summary` :Women in the video <span_0> <span_1>. | |
| gold | Women in the video have been identified as current or former American Apparel workers. |
| (extrinsic) generated | Women in the video were allegedly sexually assaulted by Morales. |

Table 8: Examples of the formatted input at test time and the real output of the FALSESUM generation model. Blue-highlighted spans show the formatted input predicates. Green-highlighted spans show the formatted input arguments. Yellow-highlighted spans show the formatted input control code. Gray-highlighted spans show the formatted input masked gold summary. Red-highlighted spans show the information inserted by the model to render inconsistent summaries.

# Part III

# Epilogue

# Chapter 9

# Conclusion and Future Work

## 9.1 Conclusion

The robustness of Natural Language Understanding (NLU) models is paramount to enabling reliable and trustworthy real-world applications of the resulting systems. In this thesis, we present methods and analysis which address several avenues to mitigate the reliance of models on spurious correlations in the training data and gain more robust generalization to out-of-distribution data.

In Chapter 5, we first propose a method to prevent models from exploiting spurious features that are known apriori. The proposed strategy addresses the limitation in the existing work where there is a substantial trade-off between the in-distribution and the out-of-distribution performance upon the "debiasing" process. In the next chapter, we discuss realistic settings where the prior knowledge about the spurious correlation is not explicitly available. We provide an analysis that demonstrates how the training dynamics of the Pre-trained Language Model (PLM) can explain the learning of spurious correlation during task-specific fine-tuning. We then propose a novel strategy to incorporate this insight to effectively "debias" models with only implicit knowledge of the spurious features. That is, we only need to know which examples are supporting the spurious correlation (and to what degree) without knowing the specifics of the features. Later in Chapter 7, we look at the importance of the inherent capabilities of PLM to resist using inference shortcuts. Using the new paradigm of prompting, we show that zero-shot PLMs perform initially well on the out-of-distribution test data, but this performance is gradually degraded by task-specific fine-tuning. We then propose a regularization method for *low resource fine-tuning* that improves the task-specific performance while still maintaining the out-of-distribution generalization. Finally, in Chapter 8, we study the data augmentation strategy to address the data discrepancy between NLU training and the downstream applications. Specifically, we look at the task of factual inconsistency detection in summarization as an out-of-the-box application of Natural Language Inference (NLI) models. We discuss the limitation of the current synthetic data generation and propose a novel method that generates diverse and more naturalistic examples for data augmentation. Our evaluation shows that the proposed data augmentation improves the robustness of NLI models on the downstream tasks application.

Overall, we show that the problem of spurious correlation learning can be attributed to various components of the NLU model training including the dataset, pre-training, fine-tuning and learning dynamics. Our proposed strategies then address these components to facilitate the development of NLU models with minimized adverse effects of the spurious correlation.

## 9.2 Future Work

We highlight a few remaining challenges in the improvement of robustness against spurious correlation that deserve further study by the community:

**Combining mitigation approaches** In this thesis, we discuss several directions from which the NLU community is tackling the robustness issues. While the resulting improvement is promising, the existing work evaluates their proposed approaches in isolation. In practice, refinement in various parts of the model development pipeline can be applied at the same time to increase their effects. It is therefore crucial to study how to effectively combine the existing mitigation approaches and measure their compounded effects.

**Addressing multiple biases** Multiple types of bias often co-occur across datasets of NLU tasks. They may interact with each other and so it is difficult and distinguish and explicitly characterize them (Shah et al., 2020). Models trained on these datasets are also likely to rely on multiple spurious correlations in their predictions. For instance, an NLI model can adopt both hypothesis-only (Gururangan et al., 2018) and lexical overlap (McCoy et al., 2019) at the same time. A possible approach is to ensemble multiple "bias-only" models where each captures a specific type of bias. Their predictions can then be combined to identify training examples that exhibit biases. Alternatively, our proposed approach to address *unknown* biases in Chapter 6 may reduce the effects of multiple biases, as suggested by the datasets generalization evaluation. However, more systematic evaluation and studies are required to precisely characterized how each bias is alleviated.

**Improved pre-training for robustness** Our zero-shot analysis using prompt-based NLU formulation in Chapter 7 demonstrates that pre-training plays a significant role in determining the robustness to spurious correlations. Our results suggest that models that are pre-trained on refined training loss and larger text corpora, e.g., RoBERTa (Liu et al., 2019b), acquire higher out-of-distribution performance compared to the standard BERT model (Devlin et al., 2019). This warrant further studies to look at other existing pre-trained language models or to develop a novel pre-training approach oriented toward improving robustness. Several possible directions include explicit injection of linguistics and world knowledge, architecture design changes, training loss refinement, or better curation of pre-training data.

# Bibliography

Abujabal, A., Saha Roy, R., Yahya, M., and Weikum, G. (2019). ComQA: A community-sourced dataset for complex factoid question answering with paraphrase clusters. 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 307–317, Minneapolis, Minnesota. Association for Computational Linguistics.

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., and Wiebe, J. (2014). SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A. (2012). SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Arpit, D., Jastrzundefinedbski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. (2017). A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 233–242. JMLR.org.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

Belinkov, Y., Poliak, A., Shieber, S., Van Durme, B., and Rush, A. (2019a). Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Belinkov, Y., Poliak, A., Shieber, S., Van Durme, B., and Rush, A. (2019b). On adversarial removal of hypothesis-only bias in natural language inference. In *Pro-*

*ceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 256–262, Minneapolis, Minnesota. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Bentivogli, L., Clark, P., Dagan, I., and Giampiccolo, D. (2010). The sixth PASCAL recognizing textual entailment challenge. In *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*. NIST.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (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. Association for Computational Linguistics.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Camburu, O.-M., Rocktäschel, T., Lukasiewicz, T., and Blunsom, P. (2018). e-snli: Natural language inference with natural language explanations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Caruana, R. (1997). Multitask learning. *Mach. Learn.*, 28(1):41–75.

Cengiz, C. and Yuret, D. (2020). Joint training with semantic role labeling for better generalization in natural language inference. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 78–88, Online. Association for Computational Linguistics.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (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. Association for Computational Linguistics.

Chen, J., Choi, E., and Durrett, G. (2021). Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chen, J. and Durrett, G. (2021). Robust question answering through sub-part alignment. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1251–1263, Online. Association for Computational Linguistics.

Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., and Inkpen, D. (2017). Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Clark, C., Yatskar, M., and Zettlemoyer, L. (2019). Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):2493–2537.

Conneau, A. and Kiela, D. (2018). SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Dagan, I., Glickman, O., and Magnini, B. (2006). The pascal recognising textual entailment challenge. In Quiñonero-Candela, J., Dagan, I., Magnini, B., and d'Alché Buc, F., editors, *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

de Marneffe, M.-C., Rafferty, A. N., and Manning, C. D. (2008). Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio. Association for Computational Linguistics.

Demszky, D., Guu, K., and Liang, P. (2018). Transforming question answering datasets into natural language inference datasets. *ArXiv*, abs/1809.02922.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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. Association for Computational Linguistics.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

Dolan, W. B. and Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Du, M., Manjunatha, V., Jain, R., Deshpande, R., Dernoncourt, F., Gu, J., Sun, T., and Hu, X. (2021). Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online. Association for Computational Linguistics.

Falke, T., Ribeiro, L. F. R., Utama, P. A., Dagan, I., and Gurevych, I. (2019). Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Feng, S., Wallace, E., and Boyd-Graber, J. (2019). Misleading failures of partial-input baselines. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5533–5538, Florence, Italy. Association for Computational Linguistics.

Ferreira, W. and Vlachos, A. (2016). Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California. Association for Computational Linguistics.

Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1180–1189. JMLR.org.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.

Gao, T., Fisch, A., and Chen, D. (2021). Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Gardner, M., Artzi, Y., Basmov, V., Berant, J., Bogin, B., Chen, S., Dasigi, P., Dua, D., Elazar, Y., Gottumukkala, A., Gupta, N., Hajishirzi, H., Ilharco, G., Khashabi, D., Lin, K., Liu, J., Liu, N. F., Mulcaire, P., Ning, Q., Singh, S., Smith, N. A., Subramanian, S., Tsarfaty, R., Wallace, E., Zhang, A., and Zhou, B. (2020). Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. (2007). The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Glockner, M., Shwartz, V., and Goldberg, Y. (2018). Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Han, D., Kim, J., and Oh, A. (2020). Reducing annotation artifacts in crowdsourcing datasets for natural language processing. In *The eighth AAAI Conference on Human Computation and Crowdsourcing*. AAAI.

He, H., Zha, S., and Wang, H. (2019). Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Hendrycks, D., Liu, X., Wallace, E., Dziedzic, A., Krishnan, R., and Song, D. (2020). Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Hu, H., Richardson, K., Xu, L., Li, L., Kübler, S., and Moss, L. (2020). OCNLI: Original Chinese Natural Language Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3512–3526, Online. Association for Computational Linguistics.

Jia, R. and Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Kamath, A., Jia, R., and Liang, P. (2020). Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Karimi Mahabadi, R., Belinkov, Y., and Henderson, J. (2020). End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Kaushik, D., Hovy, E., and Lipton, Z. (2020). Learning the difference that makes a difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, 26 April - 1 May, 2019*. OpenReview.net.

Kaushik, D. and Lipton, Z. C. (2018). How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Khashabi, D., Khot, T., and Sabharwal, A. (2020). More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 163–170, Online. Association for Computational Linguistics.

Kryscinski, W., Keskar, N. S., McCann, B., Xiong, C., and Socher, R. (2019). Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Kryscinski, W., McCann, B., Xiong, C., and Socher, R. (2020). Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Laban, P., Schnabel, T., Bennett, P. N., and Hearst, M. A. (2022). SummaC: Revisiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. (2017). RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Lai, Y., Zhang, C., Feng, Y., Huang, Q., and Zhao, D. (2021). Why machine reading comprehension models learn shortcuts? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 989–1002, Online. Association for Computational Linguistics.

Le Bras, R., Swayamdipta, S., Bhagavatula, C., Zellers, R., Peters, M. E., Sabharwal, A., and Choi, Y. (2020). Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Liu, N. F., Schwartz, R., and Smith, N. A. (2019a). Inoculation by fine-tuning: A method for analyzing challenge datasets. 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 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. (2020). Early-learning regularization prevents memorization of noisy labels. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Liu, X., Shen, Y., Duh, K., and Gao, J. (2018). Stochastic answer networks for machine reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1694–1704, Melbourne, Australia. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Lovering, C., Jha, R., Linzen, T., and Pavlick, E. (2021). Predicting inductive biases of pre-trained models. In *International Conference on Learning Representations, ICLR 2021, Virtual Conference, 3 May - 8 May, 2021*. OpenReview.net.

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

McCann, B., Keskar, N. S., Xiong, C., and Socher, R. (2018). The natural language decathlon: Multitask learning as question answering. *CoRR*, abs/1806.08730.

McCoy, T., Pavlick, E., and Linzen, T. (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. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Min, J., McCoy, R. T., Das, D., Pitler, E., and Linzen, T. (2020). Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Mishra, A., Patel, D., Vijayakumar, A., Li, X. L., Kapanipathi, P., and Talamadupula, K. (2021). Looking beyond sentence-level natural language inference for question answering and text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1322–1336, Online. Association for Computational Linguistics.

Moosavi, N. S., de Boer, M., Utama, P. A., and Gurevych, I. (2020). Improving robustness by augmenting training sentences with predicate-argument structures. *arXiv preprint arXiv:2010.12510*.

Moosavi, N. S., Utama, P. A., Rücklé, A., and Gurevych, I. (2019). Improving generalization by incorporating coverage in natural language inference. *arXiv preprint arXiv:1909.08940*.

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., and Allen, J. (2016). A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Naik, A., Ravichander, A., Sadeh, N., Rose, C., and Neubig, G. (2018). Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T., and Verspoor, K. (2017). SemEval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, Vancouver, Canada. Association for Computational Linguistics.

Nakov, P., Màrquez, L., Moschitti, A., Magdy, W., Mubarak, H., Freihat, A. A., Glass, J., and Randeree, B. (2016). SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 525–545, San Diego, California. Association for Computational Linguistics.

Nangia, N. and Bowman, S. R. (2019). Human vs. muppet: A conservative estimate of human performance on the GLUE benchmark. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4566–4575, Florence, Italy. Association for Computational Linguistics.

Nie, Y., Wang, Y., and Bansal, M. (2019). Analyzing compositionality-sensitivity of nli models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6867–6874.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (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. Association for Computational Linguistics.

Parikh, A., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

Parrish, A., Huang, W., Agha, O., Lee, S.-H., Nangia, N., Warstadt, A., Aggarwal, K., Allaway, E., Linzen, T., and Bowman, S. R. (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. Association for Computational Linguistics.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1310–III–1318. JMLR.org.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (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, New Orleans, Louisiana. Association for Computational Linguistics.

Peters, M. E., Ruder, S., and Smith, N. A. (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. Association for Computational Linguistics.

Poliak, A., Haldar, A., Rudinger, R., Hu, J. E., Pavlick, E., White, A. S., and Van Durme, B. (2018a). Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium. Association for Computational Linguistics.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018b). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Prost, F., Thain, N., and Bolukbasi, T. (2019). Debiasing embeddings for reduced gender bias in text classification. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 69–75, Florence, Italy. Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. Technical report, OpenAI.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Rozen, O., Shwartz, V., Aharoni, R., and Dagan, I. (2019). Diversify your datasets: Analyzing generalization via controlled variance in adversarial datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 196–205, Hong Kong, China. Association for Computational Linguistics.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. (2020). Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 8732–8740. AAAI Press.

Schick, T. and Schütze, H. (2021). Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Schuster, M. and Nakajima, K. (2012). Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Schuster, T., Shah, D., Yeo, Y. J. S., Roberto Filizzola Ortiz, D., Santus, E., and Barzilay, R. (2019). Towards debiasing fact verification models. 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 3419–3425, Hong Kong, China. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (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, Berlin, Germany. Association for Computational Linguistics.

Sgall, P. (1982). Natural language understanding and the perspectives of question answering. In *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*.

Shah, D. S., Schwartz, H. A., and Hovy, D. (2020). Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online. Association for Computational Linguistics.

Sharma, R., Allen, J., Bakhshandeh, O., and Mostafazadeh, N. (2018). Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, Melbourne, Australia. Association for Computational Linguistics.

Stacey, J., Belinkov, Y., and Rei, M. (2022). Supervising model attention with human explanations for robust natural language inference. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11349–11357. AAAI Press.

Stacey, J., Minervini, P., Dubossarsky, H., Riedel, S., and Rocktäschel, T. (2020). Avoiding the Hypothesis-Only Bias in Natural Language Inference via Ensemble Adversarial Training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8281–8291, Online. Association for Computational Linguistics.

Stowe, K., Utama, P., and Gurevych, I. (2022). IMPLI: Investigating NLI models' performance on figurative language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5375–5388, Dublin, Ireland. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., and Fidler, S. (2015). Movieqa: Understanding stories in movies through question-answering. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4631–4640.

Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018). FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., and Suleman, K. (2017). NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Tsuchiya, M. (2018). Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tu, L., Lalwani, G., Gella, S., and He, H. (2020). An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Utama, P., Bambrick, J., Moosavi, N., and Gurevych, I. (2022). Falsesum: Generating document-level NLI examples for recognizing factual inconsistency in summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2763–2776, Seattle, United States. Association for Computational Linguistics.

Utama, P., Moosavi, N. S., Sanh, V., and Gurevych, I. (2021). Avoiding inference heuristics in few-shot prompt-based finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9063–9074, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Utama, P. A., Moosavi, N. S., and Gurevych, I. (2020a). Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.

Utama, P. A., Moosavi, N. S., and Gurevych, I. (2020b). Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vlachos, A. and Riedel, S. (2014). Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies* *and Computational Social Science*, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

Vo, N. P. A., Magnolini, S., and Popescu, O. (2015). Paraphrase identification and semantic similarity in Twitter with simple features. In *Proceedings of the third International Workshop on Natural Language Processing for Social Media*, pages 10–19, Denver, Colorado. Association for Computational Linguistics.

Voita, E. and Titov, I. (2020). Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wang, H., Gan, Z., Liu, X., Liu, J., Gao, J., and Wang, H. (2019). Adversarial domain adaptation for machine reading comprehension. 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 2510–2520, Hong Kong, China. Association for Computational Linguistics.

Welbl, J., Stenetorp, P., and Riedel, S. (2018). Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Williams, A., Thrush, T., and Kiela, D. (2022). ANLIzing the adversarial natural language inference dataset. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 23–54, online. Association for Computational Linguistics.

Wu, B., Huang, H., Wang, Z., Feng, Q., Yu, J., and Wang, B. (2019). Improving the robustness of deep reading comprehension models by leveraging syntax prior. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 53–57, Hong Kong, China. Association for Computational Linguistics.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Xu, W., Ritter, A., Callison-Burch, C., Dolan, W. B., and Ji, Y. (2014). Extracting lexically divergent paraphrases from Twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. (2018). SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Zhang, G., Bai, B., Liang, J., Bai, K., Chang, S., Yu, M., Zhu, C., and Zhao, T. (2019). Selection bias explorations and debias methods for natural language sentence matching datasets. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4418–4429, Florence, Italy. Association for Computational Linguistics.

Zhao, J., Zhou, Y., Li, Z., Wang, W., and Chang, K.-W. (2018). Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, Brussels, Belgium. Association for Computational Linguistics.

Zhu, Y., Kiros, R., Zemel, R. S., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, pages 19–27. IEEE Computer Society.

# Appendix A

# Data Handling

In accordance with DFG's "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 5: https://github.com/UKPLab/acl2020-confidence-regularization
- Chapter 6: https://github.com/UKPLab/emnlp2020-debiasing-unknown
- Chapter 7: https://github.com/UKPLab/emnlp2021-prompt-ft-heuristics
- Chapter 8: https://github.com/joshbambrick/Falsesum

All publications related to this thesis are publicly available on the ACL Anthology (aclweb.org/anthology/):

- Chapter 5: https://aclanthology.org/2020.acl-main.770/
- Chapter 6: https://aclanthology.org/2020.emnlp-main.613/
- Chapter 7: https://aclanthology.org/2021.emnlp-main.713/
- Chapter 8: https://aclanthology.org/2022.naacl-main.199/

Moreover, all research results of the aforementioned publications are documented in the present thesis, which is archived by the Universitäts- und Landesbibliothek Darmstadt.

[1] https://www.dfg.de/download/pdf/foerderung/grundlagen_dfg_foerderung/forschungsdaten/leitlinien_forschungsdaten.pdf