Representation Learning and Learning from Limited Labeled Data for Community Question Answering

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades Dr.-Ing.

vorgelegt von
Andreas Rücklé
geboren in Darmstadt

Tag der Einreichung: 10. Februar 2021
Tag der Disputation: 12. April 2021

Referenten: Prof. Dr. Iryna Gurevych, Darmstadt, Germany
Prof. Dr. Jonathan Berant, Tel Aviv, Israel
Prof. Dr. Goran Glavaš, Mannheim, Germany

Darmstadt 2021
D17

Rücklé, Andreas: Representation Learning and Learning from Limited Labeled Data for Community Question Answering
Darmstadt, Technische Universität Darmstadt
Year thesis published in TUprints: 2021
Day of the viva voce: 12. April 2021
URN: urn:nbn:de:tuda-tuprints-185080
URL: http://tuprints.ulb.tu-darmstadt.de/18508

This document is provided by tuprints,
E-Publishing-Service of the TU Darmstadt
http://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

Ehrenwörtliche Erklärung

Gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades "Dr.-Ing." mit dem Titel "Representation Learning and Learning from Limited Labeled Data for Community Question Answering" selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Berlin, den 29. April 2021

Andreas Rücklé

Wissenschaftlicher Werdegang des Verfassers

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

10/10 – 10/13	Bachelor of Science (B.Sc.) in Informatik an der Technischen Universität Darmstadt
10/13 – 02/16	Master of Science (M.Sc.) Informatik an der Technischen Universität Darmstadt
03/16 – heute	Doktorand am Fachgebiet Ubiquitous Knowledge Processing (UKP-Lab) der Technischen Universität Darmstadt

Abstract

The amount of information published on the Internet is growing steadily. Accessing the vast knowledge in them more effectively is a fundamental goal of many tasks in natural language processing. In this thesis, we address this challenge from the perspective of *community question answering* by leveraging data from web forums and Q&A communities to find and identify answers for given questions automatically. More precisely, we are concerned with fundamental challenges that arise from this setting, broadly categorized in (1) obtaining better text representations and (2) dealing with scenarios where we have little or no labeled training data.

We first study attention mechanisms for learning representations of questions and answers to compare them efficiently and effectively. A limitation of previous approaches is that they leverage question information when learning answer representations. This procedure of dependent encoding requires us to obtain separate answer representations for each question, which is inefficient. To remedy this, we propose a self-attentive model that does not suffer from this drawback. We show that our model achieves on-par or better performance for answer selection tasks compared to other approaches while allowing us to encode questions and answers independently. Due to the importance of attention mechanisms, we present a framework to effortlessly transform answer selection models into prototypical question answering systems for the interactive inspection and side-by-side comparison of attention weights.

Besides purely monolingual approaches, we study how to transfer text representations across languages. A popular concept to obtain universally re-usable representations is the one of sentence embeddings. Previous work either studied them only monolingually or cross-lingually for only a few individual datasets. We go beyond this by studying universal cross-lingual sentence embeddings, which are re-usable across many different classification tasks *and* across languages. Our training-free approach generalizes the concept of average word embeddings by concatenating different kinds of word embeddings and by computing several generalized means. Due to its simplicity, we can effortlessly extend our approach to new languages by incorporating cross-lingual word embeddings. We show that our sentence embeddings outperform more complex techniques monolingually on nine tasks and achieve the best results cross-lingually for the transfer from English to German and French.

We complement this by studying an orthogonal approach where we machine translate the input from German to English and continue monolingually. We investigate the impact of a standard neural machine translation model on the performance of models for determining question similarity in programming and operating systems forums. We highlight that translation mistakes can have a substantial performance impact, and we mitigate this by adapting our machine translation models to these specialized domains using back-translation.

In the second part, we study monolingual scenarios with (a) little labeled data, (b) only unlabeled data, (c) no target dataset information. These are critical challenges in our setting as there exist large numbers of web forums that contain only a few labeled question-answer pairs and no labeled similar questions.

One approach to generalize from small training data is to use simple models with few trainable layers. We present COALA, a shallow task-specific network architecture specialized in answer selection, containing only *one* trainable layer. This layer learns representations of word n-grams in questions and answers, which we compare and aggregate for scoring. Our approach improves upon a more complex compare-aggregate architecture by 4.5 percentage points on average, across six datasets with small training data. Moreover, it outperforms standard IR baselines already with 25 labeled instances.

The standard method for training models to determine question similarity requires labeled question pairs, which do not exist for many forums. Therefore, we investigate alternatives such as self-supervised training with question title-body information, and we propose duplicate question generation. By leveraging larger amounts of unlabeled data, we show that both methods can achieve substantial improvements over adversarial domain transfer and outperform supervised in-domain training on two datasets. We find that duplicate question generation transfers well to unseen domains, and that we can leverage self-supervised training to obtain suitable answer selection models based on state-of-the-art pre-trained transformers.

Finally, we argue that it can be prohibitive to train separate specialized models for each forum. It is desirable to obtain one model that generalizes well to several unseen scenarios. Towards this goal, we broadly study the zero-shot transfer capabilities of text matching models in community question answering. We train 140 models with self-supervised training signals on different forums and transfer them to nine evaluation datasets of question similarity and answer selection tasks. We find that the large majority of models generalize surprisingly well, and in six cases, all models outperform standard IR baselines. Our analyses reveal that considering a broad selection of source domains is crucial because the best zero-shot transfer performance often correlates with neither domain similarity nor training data size. We investigate different combination techniques and propose incorporating self-supervised and supervised multi-task learning with data from *all* source forums. Our best model for zero-shot transfer, MultiCQA, outperforms in-domain models on six datasets even though it has not seen target-domain data during training.

Zusammenfassung

Die Menge der im Internet veröffentlichten Informationen wächst stetig. Ein grundlegendes Ziel vieler Aufgaben in der natürlichen Sprachverarbeitung ist es, einen effektiven Zugriff auf dieses Wissen zu ermöglichen. In dieser Arbeit adressieren wir dies aus der Perspektive des *Community Question Answering*, indem wir Daten aus Webforen und Q&A-Communities nutzen, um Antworten auf gestellte Fragen automatisch zu finden. Insbesondere beschäftigen wir uns mit grundlegenden Herausforderungen, die sich daraus ergeben, unterteilt in (1) das Erlernen besserer Textrepräsentationen und (2) den Umgang mit Szenarien, in denen wir nur wenige oder keine annotierten Trainingsdaten zur Verfügung haben.

Zunächst untersuchen wir sogenannte Attention-Mechanismen für das Erlernen von Frage- und Antwortrepräsentationen, um einen effizienten und effektiven Vergleich der beiden Texte zu ermöglichen. Eine Einschränkung bisheriger Ansätze besteht darin, dass diese beim Lernen von Antwortrepräsentationen auf Informationen in der Frage zurückgreifen. Diese Abhängigkeit macht es erforderlich, für jede Frage eine separate Antwortrepräsentation zu lernen, was ineffizient ist. Wir schlagen als Alternative ein Modell mit Self-Attention vor, welches nicht unter diesem Nachteil leidet. Wir zeigen, dass unser Modell im Vergleich zu anderen Modellen gleichwertige oder bessere Ergebnisse bei der automatischen Antwortselektion erreicht, während es uns zudem erlaubt, Frage- und Antwortrepräsentationen unabhängig zu lernen. Aufgrund der großen Bedeutung von Attention-Mechanismen stellen wir daraufhin ein Framework vor, mit dem wir Modelle zur Antwortselektion in prototypische Frage-Antwort Systeme überführen können. Dies ermöglicht es Forschern verschiedene Attention-Mechanismen interaktiv zu untersuchen und zu vergleichen.

Neben rein monolingualen Ansätzen untersuchen wir darüber hinaus, wie wir sprachübergreifende Textrepräsentationen lernen können. Ein beliebter Ansatz sind sogenannte Sentence Embeddings, die oft als universelle Textrepräsentationen Anwendung finden. Allerdings haben bisherige Arbeiten diese nur monolingual oder nur für wenige sprachübergreifende Aufgaben untersucht. Wir gehen darüber hinaus, indem wir Sentence Embeddings sprachübergreifend für eine Vielzahl von Klassifizierungsaufgaben untersuchen. Wir schlagen einen trainingsfreien Ansatz vor, der ein effizientes Verfahren verallgemeinert, welches den arithmetischen Mittelwert über die Embeddings von Wörtern in einem Satz berechnet. Wir erweitern dies mit verschiedenen Typen von Embeddings und berechnen mehrere Arten von Mittelwerten. Wir können unseren Ansatz dabei mühelos auf neue Sprachen übertragen, indem wir sprachübergreifende Word Embeddings integrieren. Wir zeigen, dass unsere Sentence Embeddings die meisten komplexeren Techniken auf neun monolingualen Klassifikationsaufgaben übertreffen und sprachübergreifend die besten Ergebnisse für den Transfer vom Englischen ins Deutsche und Französische erzielen.

Ergänzend untersuchen wir einen orthogonalen Ansatz, indem wir den Eingabetext vom Deutschen ins Englische maschinell übersetzen und monolingual fortfahren. Wir untersuchen den Einfluss eines neuronalen maschinellen Übersetzungsmodells auf die Effektivität von Modellen zur Bestimmung der Frageähnlichkeit in

Programmier- und Betriebssystemforen. Wir zeigen, dass Übersetzungsfehler einen erheblichen Einfluss auf die Effektivität dieses Ansatzes haben können, und verbessern dies, indem wir unser maschinelles Übersetzungsmodell durch Rückübersetzung an unsere speziellen Domänen anpassen.

Im zweiten Teil untersuchen wir monolinguale Szenarien mit (a) wenigen annotierten Daten, (b) gänzlich ohne Annotationen, (c) keiner Zieldatensatzinformation. Dies sind wichtige Herausforderungen, da eine Vielzahl an Webforen nur wenige annotierte Frage-Antwort-Paare und keine annotierten ähnlichen Fragen enthalten.

Eine Möglichkeit um bereits mit wenigen Daten ein effektives Modell zu erlernen ist die Nutzung einfacher Architekturen mit wenigen trainierbaren Netzwerkschichten. Unser Ansatz COALA ist eine flache, aufgabenspezifische neuronale Architektur, welche auf die Antwortselektion spezialisiert ist und nur *eine* trainierbare Netzwerkschicht enthält. Diese lernt Repräsentationen von Wort-N-Grammen in Fragen und Antworten, die wir vergleichen und anschließend aggregieren. Unser Modell erzielt gegenüber eines komplexeren Compare-Aggregate Modells Verbesserungen von durchschnittlich 4,5 Prozentpunkten, über sechs Datensätze hinweg. Darüber hinaus übertrifft es die IR-Baselines bereits mit 25 annotierten Beispielen.

Der Standardansatz für das Training von Modellen zur Bestimmung der Fragenähnlichkeit erfordert annotierte Fragenpaare, die in vielen Foren nicht existieren. Daher beschäftigen wir uns mit Alternativen, wie dem selbstüberwachten Training mit Informationen aus dem Fragentitel und -body und schlagen die Generierung von Fragenduplikaten vor. Wir zeigen, dass mittels Nutzung größerer Datenmengen erhebliche Verbesserungen gegenüber Adversarial-Domain-Transfer erzielt werden können und wir übertreffen damit das überwachte In-Domain-Training auf zwei Datensätzen. Weiterhin zeigen wir, dass sich Modelle zur Generierung von Fragenduplikaten gut auf andere Domänen übertragen lassen, und dass wir selbstüberwachtes Training nutzen können, um effektive Modelle zur Antwortselektion zu erhalten.

Schließlich argumentieren wir, dass es unerschwinglich sein kann, separate spezialisierte Modelle für jedes einzelne Forum zu trainieren. Es ist vorteilhaft, ein einziges Modell zu erhalten, das breit wiederverwendbar ist. Um dieses Ziel zu erreichen, untersuchen wir zunächst den Zero-Shot-Transfer von Text-Matching-Modellen im Kontext des Community Question Answering. Wir trainieren 140 Modelle mit selbstüberwachtem Training unter der Nutzung von Daten aus verschiedenen Foren. Wir analysieren daraufhin die Effektivität dieser Modelle auf neun Evaluationsdatensätzen für Frageähnlichkeits- und Antwortselektionsaufgaben. Wir stellen fest, dass die große Mehrheit der Modelle überraschend gute Ergebnisse erzielt. In sechs Fällen übertreffen alle Modelle die IR-Baselines. Unsere Analysen ergeben, dass es wichtig ist eine breite Auswahl an Foren für das Training von Modellen zu berücksichtigen, da wir die besten Modelle weder mittels Domänenähnlichkeit noch mittels der Größe der Trainingsdaten zuverlässig vorhersagen können. Abschließend untersuchen wir verschiedene Techniken um Daten aus mehreren Foren zu kombinieren und schlagen vor, selbstüberwachtes und überwachtes Multi-Task-Learning mit Daten aus *allen* Foren zu kombinieren. Unser bestes Modell für Zero-Shot-Transfer, MultiCQA, erreicht bessere Ergebnisse als bisherige Modelle auf sechs Datensätzen, obwohl es nicht explizit für diese trainiert wurde.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to Prof. Dr. Iryna Gurevych for her guidance over the past five years, her advice, the insightful discussions, and for giving me the chance to advance both my scientific and management skills considerably. I would also like to thank Prof. Dr. Jonathan Berant and Prof. Dr. Goran Glavaš for investing their time in reviewing my thesis.

I want to thank my colleagues at UKP Lab for the numerous valuable discussions and constructive feedback: Dr. Steffen Eger, Jonas Pfeiffer, Dr. Nafise Sadat Moosavi, Dr. Ivan Habernal, Dr. Nils Reimers, Dr. Maxime Peyrard and many more as part of our reading groups and internal review cycles. I am also grateful to my research assistants for their valuable contributions, exceptional work, and dedication under my (co-)supervision, hoping that my guidance has also helped them advance their careers: Omnia Zayed, Xuan-Son Vu, Krishnkant Swarnkar, Xia Zeng. Likewise, I would like to thank my (thesis-)students for working with me on numerous interesting and challenging topics: Nadja Geisler, Paul Dubs, Johann Wiedmeier, Jana Vatter, Martin Lichtblau, Gregor Geigle.

Leading my own project "Intelligent Search in the Social Web" as part of the SoftwareCampus has helped me considerably advance my leadership skills, and I want to thank the SoftwareCampus team for organizing such a wonderful program. Furthermore, I would like to thank the German Federal Ministry of Education and Research for funding my project with up to 100 000 EUR (2018–2020) and the numerous employees of DATEV eG for their feedback with regards to applying my research in an industrial setting.

Finally, I am deeply grateful to my family Cornelia, Brigitte and Gerhard for their continuous and immense moral support during my PhD journey, which had a profound impact on my studies.

Funding. This work has been supported by the German Research Foundation (DFG) as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1), by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 03VP02540 (ArgumenText), by the BMBF as part of the Software Campus program under the promotional reference 01IS17050, by the BMBF 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, and by the DFG under grant EC 503/1-1 and GU 798/21-1. I gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40, Titan X Pascal, and Titan Xp Pascal GPUs used for this research. Some calculations for this research were conducted on the Lichtenberg high performance computer of the TU Darmstadt.

Contents

Chapter 1

Introduction

1.1 Motivation

The amount of information made accessible on the Internet has grown rapidly in recent years. For example, the English version of Wikipedia contains more than 6 million documents totaling more than 3.7 billion words.[1] The knowledge graph WikiData spans over more than 90 million entries, representing objects, concepts, and other artifacts of human knowledge.[2] The online discussion forum Reddit collected more than 1.7 billion user comments in 2019, after 1.2 billion in 2018.[3]

Although these are only three examples, they reveal the enormous breadth of this ongoing trend. A fundamental challenge that arises from this trend is the so-called information overload (Allen and Wilson, 2003), as it is arguably impossible for us to manually benefit from all the available information. Mitigating the information overload remains a central motivation in natural language processing (NLP). The vast amounts of information can also be seen as an unparalleled opportunity, since effective methods of accessing this knowledge can provide considerable benefits to the society.

In this thesis, we are concerned with making the large amounts of information accessible by means of automatic **question answering** (QA). This is a particularly promising direction, which is best exemplified by perhaps the most famous QA system in recent history, IBM Watson (Ferrucci, 2012). In 2011, IBM Watson publicly beat the human champions in the television game show *Jeopardy!* using large amounts of heterogeneous data. Despite the remarkable breakthrough at the time, IBM Watson was only specialized in answering trivia questions with facts. In practice, however, there exists a much wider range of questions that people ask.

Consider the following three examples:

[1] https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia; last accessed 20 Nov. 2020.

[2] https://www.wikidata.org/wiki/Wikidata:Main_Page; last accessed 07 Nov. 2020.

[3] https://redditblog.com/2018/12/04/reddit-year-in-review-2018/, https://redditblog.com/2019/12/04/reddits-2019-year-in-review/; last accessed 07 Nov. 2020.

1. How high is the Eiffel Tower?
2. Why do companies fund academic research?
3. How do I create El Capitan installer on a Catalina (or post-El Capitan) installed Mac for use on USB boot installer?

We can answer the first question by either finding the fact in a knowledge-base (e.g., Berant et al., 2013) or extracting it from the Wikipedia article of Paris (e.g., Rajpurkar et al., 2016). The second and third questions are open-ended and require other types of information, for example, experiences, explanations, or descriptions. We denote such questions as **non-factoid questions**.

Answers to non-factoid questions considerably differ from factoid answers (e.g., numbers, entities, or noun phrases). For instance, it has been shown that answers to "Why" questions typically consist of multiple sentences (Verberne et al., 2007). It has also been argued that non-factoid questions can be answered from several complementary viewpoints (Omari et al., 2016). Thus, it may be impossible to specify what would be an optimal answer. These properties illustrate that we need different methods for automatically answering factoid and non-factoid questions.

Non-factoid questions are extremely common, which we can witness by the growing popularity of web forums and Q&A communities—collectively referred to as forums here—e.g., Reddit, StackOverflow, Quora, and GuteFrage.net. Indeed many of the questions found there are non-factoid questions (Hashemi et al., 2020). Because such forums have accumulated large quantities of answers, we can re-use them for automatic question answering. This is precisely the goal of **community question answering** (cQA), as we study it in this dissertation.

For illustration, suppose that we want to build a real-world cQA system in which we are given a question that needs to be answered. A first step of this system would be to find a set of similar questions that have already been answered in forums. We denote this task as *question similarity*. From the most similar questions, we can then identify one or more suitable answers in regard to the input question and provide them as a system output. We denote this task as *answer selection*. We provide a thorough introduction to these tasks and the state of the art in Chapter 2.

Motivated by cQA question similarity and answer selection tasks, we identify two fundamental challenges that we address in this thesis. (1) Learning suitable text representations as well as transferring them across languages. (2) Dealing with limited labeled training data.

1.2 Challenges and Research Questions

1.2.1 Learning Suitable Text Representations

The so-called Lexical Gap (Berger et al., 2000; Jeon et al., 2005) has long been mentioned as a central challenge in cQA. For example, we can paraphrase questions using different words without significantly changing their meaning. Likewise, we can formulate answers that contain only synonyms of the words found in the questions.

It is straightforward to see that this negatively affects methods that count word co-occurrences for determining question similarity or selecting suitable answers.

Learning dense vectors that encode the *meaning* of questions and answers can naturally overcome the lexical gap. This technique allows us to compare the semantic similarity of questions and answers by determining their representations' proximity in a vector space. Learning such representations in an efficient *and* effective manner, however, remains challenging.

Successful approaches to learning question and answer representations utilize attention mechanisms (e.g., Dos Santos et al., 2016; Tan et al., 2016) to encode long answers more effectively. When processing the answer text, they incorporate information from the question to determine which parts of the answer are most relevant. As a consequence, we can only learn representations of answers when we are given a question, but not for a large corpus of answers independently. Despite their effectiveness, common approaches with attention mechanisms are, thus, limited in terms of efficiency. We formulate our first research question accordingly:

> **RQ1: How can we learn effective and efficient representations of questions and answers?**

Is it necessary to rely on information from the question to determine the relevance of segments within the answer? How can we determine the attention independently, thereby avoiding the limitations with regard to efficiency?

We address this in Section 3.1 and propose a self-attentive model that determines the importance of segments within the question and answer texts independently, using a dedicated LSTM (Hochreiter and Schmidhuber, 1997) component. Our experiments show that our approach performs on-par or better than other attention-based models for non-factoid and factoid answer selection, thus being both effective *and* efficient—due to independent encoding. Motivated by the success of attention mechanisms, in Section 3.2, we present an end-to-end QA system that allows researchers to explore different attention mechanisms interactively and compare them side-by-side.

Notably, we address RQ1 only from the perspective of learning question-answer representations, and only using English texts. Even though task-specific monolingual models are common in NLP, they are considerably limited in scope. The resulting models are not universal and, therefore, not widely re-usable. We address this shortcoming in our second research question:

> **RQ2: How can we obtain approaches to representation learning that transfer well to different languages?**

More accurately, we study RQ2 from two complementary perspectives.

(1) How can we obtain text representations that are universal across classification tasks, e.g., question type classification and sentiment classification? How can we transfer these representations across different languages?

One way to achieve this is by obtaining universal sentence embeddings, due to their wide range of applications (Conneau et al., 2017). Previous work, however, has either investigated *monolingual* sentence embeddings or cross-lingual sentence

embeddings *specialized* on few tasks such as translation retrieval. In Section 4.1 we therefore investigate *universal cross-lingual* sentence embeddings and propose an efficient training-free technique that combines different generalized means over pre-trained word embeddings. Our approach generalizes well across languages while better maintaining the performance in cross-lingual transfer compared to more complex techniques.

(2) How can we transfer monolingual question similarity models cross-lingually? Which challenges arise in expert domains?

Another approach for cross-lingual transfer is to translate the input text to English and continue with a monolingual model. This is particularly interesting in cQA, where we might want to leverage a large amount of English forum data to answer questions posed in other languages. In Section 4.2, we thus investigate the impact of cross-lingual transfer of question similarity models using neural machine translation. We show that for specialized questions about programming and operating systems topics, translation mistakes can have a strong influence on the question similarity performance. We find that unsupervised domain adaptation of a neural machine translation model with back-translation can mitigate such effects.

1.2.2 Dealing with Limited Labeled Training Data

The second fundamental challenge we address is obtaining good models even if we have only little or no labeled training data. As we will show later in Chapter 2, duplicate questions are commonly used to train question similarity models. However, they are only available in sufficient quantities for a small number of forums. Further, even though cQA forums often consist of sufficient quantities of answers, labeled instances of good question-answer pairs that are suitable for training models can be scarce. Obtaining such data by manual annotation is expensive, and thus other techniques are required to train neural models in such settings.

> **RQ3: How can we train cQA models in settings with limited labeled training data?**

We address RQ3 from two perspectives.

(1) Do all models need large quantities of training data? Can we design a shallow model that requires less training data?

Many neural network models are very complex and notoriously data-hungry. For instance, the compare-aggregate model of (Wang and Jiang, 2017) achieves good performances across various text matching tasks, including non-factoid answer selection. However, it consists of a large number of trainable layers. In Chapter 5, we propose a considerably simplified model, containing only *one* trainable layer within a task-specific network architecture that is specialized in non-factoid answer selection. Our model outperforms the one of Wang and Jiang (2017) by 4.5 points on average for six datasets of different domains. Moreover, we find that we can successfully train it with as little as 25 labeled instances, where it outperforms standard IR baselines.

(2) Can we train neural models to cQA tasks with only unlabeled data?

Besides obtaining shallow models that we can train with little labeled data, we are interested in methods to train neural models *without* labeled instances. We study this in Chapter 6, where we leverage the connection between question titles and question bodies for self-supervised training and for duplicate question generation. We show that both can make use of larger amounts of unlabeled data than supervised training, which can improve the model performances. We find that both methods are broadly applicable, e.g., duplicate question generation transfers well across domains, and self-supervised training can be used to train answer selection models.

One limitation of previous approaches is that they train and evaluate models for individual forums, which means that they are not widely re-usable. However, the StackExchange network alone consists of over 170 forums[4] covering a variety of topics, which we call domains. It would be desirable to obtain a single model that can be broadly reused across different domains and for both question similarity and answer selection tasks. This gives us our fourth and final research question:

> **RQ4: To which extent do text matching models generalize to unseen cQA tasks and domains?**

How well do models transfer across different cQA tasks and domains? How can we combine the training data from different domains to achieve good zero-shot transfer capabilities?

In Chapter 7, we train 140 BERT (Devlin et al., 2019) models with self-supervised training signals on different English StackExchange forums and transfer them to nine datasets of question similarity and answer selection tasks. We surprisingly find that all models transfer well in our zero-shot transfer setups and that domain similarity and the training size are often not suitable for predicting the best zero-shot transfer performances. We propose a single model that incorporates self-supervised and supervised multi-task learning on all source domains, which transfers well to our nine evaluation benchmarks and outperforms in-domain BERT in six cases. With this, we take an important step towards universally applicable cQA models.

1.3 Contributions

In the following, we summarize the most important contributions of this thesis with regard to the research questions stated above:

RQ1: How can we learn effective and efficient representations of questions and answers?

- We propose a self-attentive model for learning question and answer representations that leverages an isolated LSTM component for importance weighting. Unlike previous attention-based models in cQA, our model does not incorporate question information to derive the importance of answer segments. We can, therefore, encode all answers independently, e.g., allowing

[4] https://stackexchange.com/sites; last accessed: 12. Nov 2020.

pre-computation of all answer representations for large datasets. We demonstrate that our self-attentive model achieves on-par or better answer selection performance compared to previous attention-based models.

- We present a modular and extensible service architecture that enables researchers to transform their answer selection models into question answering systems. Our interactive visualization allows users to explore the strengths and weaknesses of attention mechanisms and to compare different models side-by-side. This can support researchers in conducting qualitative analysis and in building interactive QA prototypes.

RQ2: How can we obtain approaches to representation learning that transfer well to different languages?

- We propose concatenated power mean word embeddings as universal cross-lingual sentence representations. Our approach is training-free and generalizes average word embeddings using two computationally simple yet important ingredients: (1) the concatenation of complementary types of word embeddings to inject diverse kinds of information; (2) power means (Hardy et al., 1952) to capture more information of the sequence. We establish that, despite their simplicity, our sentence embeddings outperform many more complex methods monolingually, and thus represent a truly hard-to-beat baseline. More importantly, we extend the definition of universal sentence embeddings to the cross-lingual case: we require them to transfer well across different languages. We demonstrate that our approach achieves better results compared to three cross-lingual adaptations of InferSent (Conneau et al., 2017) when transferring from English to German or from English to French. Our versatile, cross-lingual, and extensible models can thus be re-used across several tasks and languages.

- As an alternative to obtaining cross-lingual models, we can machine translate texts to English and continue monolingually. We study such an approach for question similarity and with specialized expert domains—programming and operating systems cQA forums—where the translation quality substantially impacts the model performance. To remedy, we adapt the transformer (Vaswani et al., 2017) for neural machine translation to the idiosyncrasies of our domains using back-translation. We show that this yields considerable improvements upon the standard approach that trains neural machine translation models on parallel texts from parliament proceedings.

RQ3: How can we train cQA models in settings with limited labeled training data?

- We propose a simple and effective text matching model, compare-aggregate for long answers (COALA), which considerably simplifies the more general compare-aggregate variant of Wang and Jiang (2017). In its basic form, COALA only contains one trainable layer, thus taking a step in the opposite direction compared to notoriously deep models. This layer learns representations of text segments in questions and answers, which we compare and score

with unsupervised techniques. We demonstrate the effectiveness of COALA on seven datasets of different domains, and more importantly, find that it already performs better than unsupervised baselines with as little as 25 labeled instances. Further, we show that due to its task-specific network structure, COALA can handle long answers substantially better than other approaches.

- We study training methods that require only unlabeled questions from cQA forums. First, we propose automatic duplicate question generation, where we generate a new question title from the question's body. We then consider the generated title as a duplicate of the question's original title and use it for training question similarity models. Second, we broadly investigate the effectiveness a self-supervised training strategy with title-body pairs, i.e., predicting whether both texts belong to the same question or are from unrelated questions.[5] We show that due to the large amounts of unlabeled in-domain questions that we can leverage, models trained with these methods can outperform adversarial domain transfer in the setup of Shah et al. (2018) by more than 5.6 points AUC(0.05)[6] on average. Because the number of labeled instances is often limited, we can in some cases achieve better performances than in-domain supervised training. We also find that our question generation models are robust against domain transfer, e.g., we can train them on StackExchange Academia and generate duplicates for AskUbuntu with minimal performance decreases—0.9 points in this case—demonstrating that our method can be broadly applied. Finally, we show that self-supervised training with title-body pairs is effective for fine-tuning BERT, and yields models that can be used to perform answer selection.

RQ4: To which extent do text matching models generalize to unseen cQA tasks and domains?

- We investigate the zero-shot transfer of text-matching models on a massive scale by self-supervised training on 140 source domains from community question answering forums. We study the model performances on nine datasets of question similarity and answer selection tasks and find that all 140 models transfer surprisingly well. Although our models were not given annotated questions, answers, or data from the target domains, in six cases, all 140 models outperform standard IR baselines. We investigate whether more similar domains or larger source datasets lead to better zero-shot transfer and, surprisingly, find that neither are suitable predictors.

- Finally, we investigate techniques for combining data from our 140 source domains, namely multi-task learning and AdapterFusion (Pfeiffer et al., 2021), to obtain a model that consistently performs well across all benchmarks. Our best

[5] A Master thesis (Wiedmeier, 2017) and Bachelor thesis (Vatter, 2019), both closely supervised by me and carried out according to my task description, have given some first evidence for the effectiveness of this method. In this thesis, we go considerably beyond this by studying it on a much broader scale with different kinds of models, more tasks, and more data. We describe several important differences in Chapter 6.

[6] Area under curve with a threshold for false positives.

zero-shot MultiCQA model incorporates self-supervised and supervised multi-task learning on all source domains, and outperforms in-domain BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) on six evaluation benchmarks. We thus demonstrate the effectiveness and benefits of universal models to cQA.

The experimental source code of our work related to the contributions listed above is publicly available at `https://github.com/ukplab`. Details on our strategy to data handling are given in Appendix A.

1.4 Publication Record

Most of our work has been published in international peer-reviewed conferences or on the arXiv pre-print server. Contents from these publications are re-used (and quoted verbatim) in this thesis. All publications have joint authorship. The contributions of the thesis author are stated in the beginning of the chapters or sections that include content of the respective publications.

In the following, we list these publications and link them to the respective dissertation chapters in which verbatim quotes from these publications are to be expected. We underline abbreviations of conference names and publication years, which we will refer to in our thesis organization (Section 1.5).

- **Andreas Rücklé** and Iryna Gurevych: 'Representation Learning for Answer Selection with LSTM-Based Importance Weighting', in: *Proceedings of the 12th International Conference on Computational Semantics* (IWCS 2017), September 2017. Section 3.1
- **Andreas Rücklé** and Iryna Gurevych: 'End-to-End Non-Factoid Question Answering with an Interactive Visualization of Neural Attention Weights', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (ACL 2017; System Demonstrations), pp. 19–24, July 2017. Section 3.2
- **Andreas Rücklé**, Steffen Eger, Maxime Peyrard, and Iryna Gurevych: 'Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations', published on the *arXiv pre-print* server (arXiv 2018; arXiv:1803.01400v2), March 2018. Section 4.1
- **Andreas Rücklé**, Krishnkant Swarnkar, and Iryna Gurevych: 'Improved Cross-Lingual Question Retrieval for Community Question Answering', in: *Proceedings of the 2019 World Wide Web Conference* (WWW 2019), pp. 3179–3186, May 2019. Section 4.2
- **Andreas Rücklé**, Nafise Sadat Moosavi, and Iryna Gurevych: 'COALA: A Neural Coverage-Based Approach for Long Answer Selection with Small Data', in: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence* (AAAI 2019), pp. 6932–6939, January 2019. Chapter 5
- **Andreas Rücklé**, Nafise Sadat Moosavi, and Iryna Gurevych: 'Neural Duplicate Question Detection without Labeled Training Data, in: *Proceedings*

of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019), pp. 1607–1617, November 2019. Chapter 6

- **Andreas Rücklé**, Jonas Pfeiffer, and Iryna Gurevych: 'MultiCQA: Zero-Shot Transfer of Self-Supervised Text Matching Models on a Massive Scale', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2020), pp. 2471–2486, November 2020. Chapter 7

In the following, we list further publications that have been completed in parallel to this work. The listed works exceed the scope of this thesis, may contain significant or predominant contributions by other authors, and are *not included* in the present thesis. We merely list them to complete the overall picture.

- **Andreas Rücklé** and Iryna Gurevych: 'Real-Time News Summarization with Adaptation to Media Attention', in: *Proceedings of the International Conference Recent Advances in Natural Language Processing* (RANLP 2017), pp. 610–617. September 2017.
- Michael Bugert, Yevgeniy Puzikov, **Andreas Rücklé**, Judith Eckle-Kohler, Teresa Martin, Eugenio Martínez-Cámara, Daniil Sorokin, Maxime Peyrard, and Iryna Gurevych: 'LSDSem 2017: Exploring Data Generation Methods for the Story Cloze Test', in: *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics* (LSDSem 2017), pp. 56–61. April 2017.
- Steffen Eger, **Andreas Rücklé**, and Iryna Gurevych: 'PD3: Better Low-Resource Cross-Lingual Transfer By Combining Direct Transfer and Annotation Projection', in: *Proceedings of the 5th Workshop on Argument Mining* (ArgMin 2018), pp. 131–143, November 2018.
- Steffen Eger, Gözde Gül Şahin, **Andreas Rücklé**, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych: 'Text Processing Like Humans Do: Visually Attacking and Shielding NLP Systems', in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL 2019), pp. 1634–1647, June 2019.
- Nafise Sadat Moosavi, Prasetya Ajie Utama, **Andreas Rücklé**, and Iryna Gurevych: 'Improving Generalization by Incorporating Coverage in Natural Language Inference', published on the *arXiv pre-print* server (arXiv:1909.08940), September 2019.
- Steffen Eger, **Andreas Rücklé**, and Iryna Gurevych: 'Pitfalls in the Evaluation of Sentence Embeddings', in: *Proceedings of 4th Workshop on Representation Learning for NLP* (RepL4NLP 2019), pp. 55–60, August 2019.
- **Andreas Rücklé**, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych: 'AdapterDrop: On the Efficiency of Adapters in Transformers', published on the *arXiv pre-print* server (arXiv:2010.11918), October 2020.
- Mingzhu Wu, Nafise Sadat Moosavi, **Andreas Rücklé**, and Iryna Gurevych:

'Improving QA Generalization by Concurrent Modeling of Multiple Biases', in: *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 839–853, November 2020.

- Jonas Pfeiffer, **Andreas Rücklé**, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych: 'AdapterHub: A Framework for Adapting Transformers', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): System Demonstrations*, pp. 46–54, November 2020.
- Jonas Pfeiffer, Aishwarya Kamath, **Andreas Rücklé**, Kyunghyun Cho, and Iryna Gurevych: 'AdapterFusion: Non-Destructive Task Composition for Transfer Learning', in: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, pp. 487–503, April 2021.

1.5 Thesis Organization

The structure of this thesis follows the same order as the contributions stated before in Section 1.3 and corresponds to the timeline of the publication record given in Section 1.4. Figure 1.1 illustrates the overall structure.

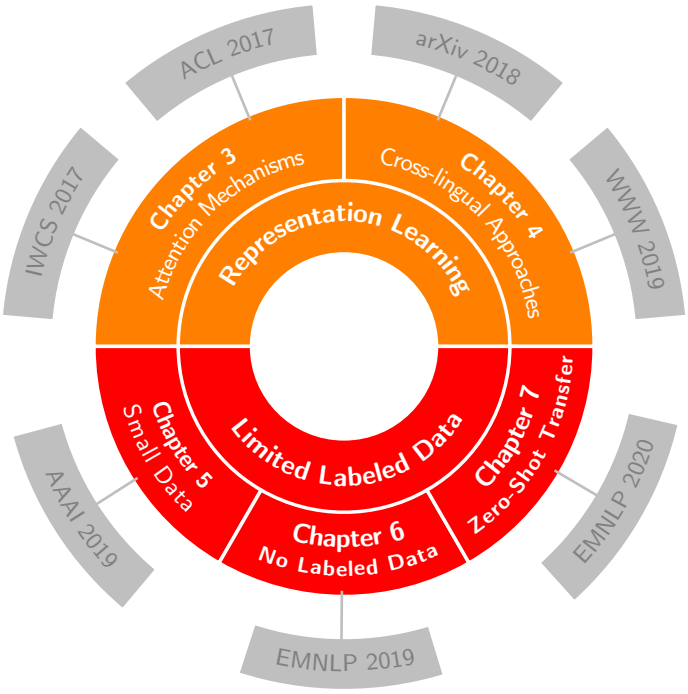

Figure 1.1: The organization of this thesis. We complement this with a chapter with background information on our work and the conclusions of the thesis, which contains an outlook on possible future work. The conference abbreviations highlighted in gray refer to the publications of Section 1.4, which the respective chapters are based on. The chapter titles are abbreviated in this figure, the full titles can be found in the text (boldface).

Chapter 2 provides the overall **background** to this thesis. We describe a prototypical cQA system architecture and relate its components to the tasks that we address throughout this dissertation. We then discuss previous work in the respective areas and compare existing research datasets.

Chapter 3 studies **attention mechanisms for learning representations of questions and answers** and introduces our self-attentive approach that learns importance weights of questions and answers independently. We present our end-to-end cQA system that enables researchers to interactively analyze and compare different attention mechanisms for answer selection tasks on real-world datasets.

Chapter 4 investigates the **cross-lingual transfer of approaches to representation learning** from two perspectives. (1) We propose concatenated power mean word embeddings as universal cross-lingual sentence embeddings and evaluate them broadly across tasks and languages. (2) We study the impact of machine translation on the cross-lingual transfer of monolingual question similarity models.

Chapter 5 is the first chapter that studies scenarios with limited training data. We study approaches to **long answer selection with small training data**, i.e., cQA scenarios where we have only few labeled instances. We propose COALA, our task-specific compare aggregate model for long answer selection, which can be trained with as little as 25 labeled instances.

Chapter 6 then takes this one step further and investigates **training cQA models without labeled data**. We investigate duplicate question generation to obtain training data, and self-supervised training with question title-body pairs. We show that due to a larger amount of unlabeled data available to these methods, we can often achieve the best results.

Chapter 7 completes this thesis by studying the **zero-shot transfer of text matching models on a massive scale**, with 140 models trained on different cQA forums with self-supervised training signals. We conduct analyses on this large sample size, investigating whether more similar domains or larger source datasets lead to better zero-shot transfer. We investigate how to combine the training data of all source domains and propose incorporating self-supervised and supervised multi-task learning on all source domains.

Chapter 8 concludes this thesis and outlines promising **future directions in cQA** and related areas for which we, and other recent work, are laying foundations.

Chapter 2

Background

This chapter provides background information on this thesis and lays the foundation for the Chapters 3 to 7. We introduce community question answering (cQA) and place it in the broader context of question answering (QA). We then summarize previous work in areas related to those covered in this thesis and present relevant research datasets. Later, we complement this in the individual chapters with additional background and related work specific to the challenges we address.

Here, we will answer the following questions:

- What is community question answering?
- What are important differences of community question answering compared to factoid question answering?
- What is the structure of a prototypical community question answering system?
- Which datasets and data sources are available?
- Which approaches have been proposed in previous work?

2.1 Community Question Answering (cQA)

We consider cQA as part of a vast field of automatic question answering (QA), in which systems and models try to understand questions posed in natural language and automatically find appropriate knowledge to answer them. It is important to note that the definition of QA varies in the literature and that the term "QA" is sometimes used synonymously with certain QA tasks. For example, machine reading comprehension, which extracts a fact from a text passage that can answer a question, is often referred to as QA (e.g., Gan and Ng, 2019; Yang et al., 2017). Partly due to the breadth of possible tasks that may be posed as question answering, Gardner et al. (2019a) argue that "[...] question answering is a format, not a task."

In this thesis, we refer to QA as an umbrella term covering a broad *variety of tasks* posed as QA. We define the central challenge of QA to find a function $f_\Psi : Q \rightarrow A$ that maps a question $q \in Q$ to an answer $a \in A$, potentially leveraging some

background information Ψ. The function f, thus, defines a question answering system. As it is arguably infeasible to obtain a single system that can answer all possible questions, there exist several potential realizations for the types of questions it can answer (Q), the answers it provides (A), and the background knowledge it can access (Ψ).

The function f may be composed of one or multiple operations. For instance, in cQA, we can identify similar questions that have been answered in a web forum and, in a separate step, assess potential answer candidates (details follow later). We refer to these operations as **QA tasks or subtasks**. They are often addressed separately and independently. Since there is a large number of QA tasks and subtasks, we briefly present the taxonomy we use throughout this thesis in Figure 2.1. Most importantly, we distinguish between two *categories* of tasks in QA: (1) tasks related to answering *factoid questions*, and (2) tasks related to answering *non-factoid questions*. Whereas factoid questions require concise facts as answers, non-factoid questions go beyond that and can often only be answered with descriptions, explanations, or advice. Table 2.1 provides examples of these two types of questions.

Common tasks in **factoid QA** are answer sentence selection (e.g., Yang et al., 2015), where we are given a question and a list of sentences that state a fact from which we must choose the correct one; knowledge-base question answering (e.g., Berant et al., 2013), often concerned with semantic parsing of questions and matching them to a knowledge graph; machine reading comprehension (e.g., Rajpurkar et al., 2018; Gardner et al., 2019b), in which models read and reason over a passage and typically extract a fact to answer the given question.

Common tasks in **non-factoid QA**, on the other hand, are non-factoid answer selection (e.g., Cohen et al., 2018), which is similar to answer sentence selection but with long passages as answers; community QA and its subtasks, which we discuss in detail in the following; long-form QA (e.g., Fan et al., 2019), in which we generate an answer passage for a question (with or without background information).

Further tasks exist that do not directly fall into this schema, up to exotic scenarios that formulate other NLP tasks as QA (McCann et al., 2018)—e.g., machine translation ("What is the translation of ...?"). We neglect such extremes here.

One possible distinction between tasks in factoid QA and non-factoid QA is given by the **challenges** they pose. For instance, a central challenge in factoid QA is to deal with complex questions that contain multiple relations (Sorokin and Gurevych, 2018) and how to break them down into smaller units (Talmor and Berant, 2018; Wolfson et al., 2020). Challenges in machine reading comprehension are how to reason over the text in a passage or document, e.g., to perform numerical reasoning (Geva et al., 2020) or implicit multi-step reasoning (Geva et al., 2021). In contrast, central challenges in non-factoid QA are how to learn suitable representations of long answers (Tan et al., 2016; Cohen et al., 2018) and how to transfer learned knowledge to different expert domains (Shah et al., 2018; Poerner and Schütze, 2019).

These differences partly stem from the fact that tasks in factoid and non-factoid QA use different kinds of **background information** Ψ. For instance, answers to non-factoid questions often cannot be found in datasets that encode general factual

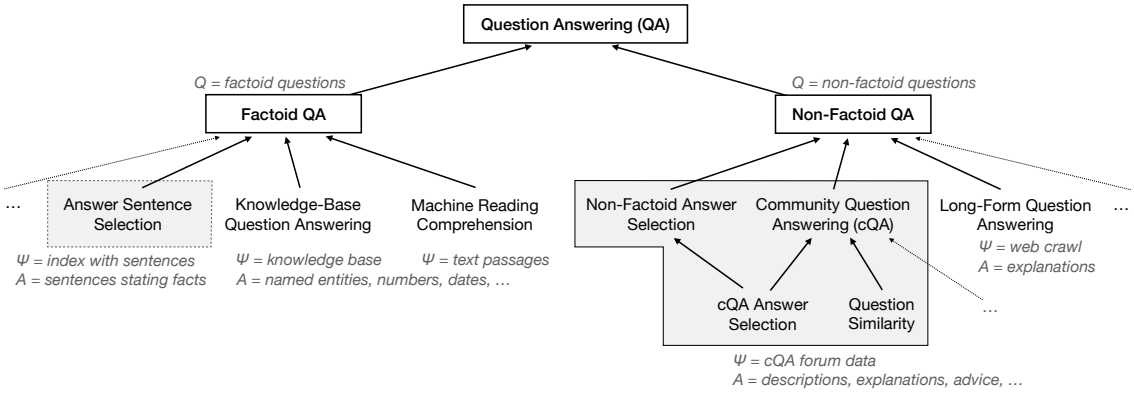

Figure 2.1: Our QA taxonomy, which includes the tasks and subtasks we refer to throughout this thesis. We distinguish between factoid QA and non-factoid QA. Gray boxes indicate which tasks we encounter in different parts of this dissertation. Our primary focus is on cQA and its subtasks. Q, A, and Ψ provide examples for the different question types, answer types, and data sources.

Question Types (Q)	**Examples**
Factoid questions	ⓘ How many atoms combine to form dioxygen? ⓘ What is the Dutch word for the Amazon rainforest? ⓘ Which coastline does Southern California touch?
Non-factoid questions	ⓘ Why do companies fund academic research? ⓘ Do I have to learn music theory to learn piano? ⓘ How does a 'rainbow table' hacker obtain password hashes in the first place?

Table 2.1: Examples of factoid and non-factoid questions. Factoid questions can be answered with facts from a knowledge base, Wikipedia, etc. Non-factoid questions, in contrast, often require explanations, advice, or descriptions as answers. The examples of factoid questions are taken from the Stanford Question Answering Dataset (SQuAD) 2.0 (Rajpurkar et al., 2018). Non-factoid questions are taken from StackExchange (SE) travel, SE music, and SE security.

knowledge such as *knowledge bases*. Resources that include such content—e.g., descriptions, explanations, and advice—are *web forums*. In these forums, users can ask questions that other users, often experts in their respective fields, answer. Some of the best-known forums are Quora, Reddit, and StackOverflow (which is part of the StackExchange network). They are widely popular: both Reddit and StackOverflow are among the 50 most visited websites globally, according to Alexa.com statistics[1]. We refer to them as *cQA forums*.

The goal of **community question answering (cQA)** is to re-use data from cQA forums to automatically answer non-factoid questions—thus being a particular instantiation of QA with a distinct set for Q (non-factoid questions), A (descriptions, explanations, advice, ...), and Ψ (cQA forum data). We include an excerpt of a typical cQA forum thread in Figure 2.2. Important properties that we can identify

[1] https://www.alexa.com/topsites; last accessed 24 June 2020.

in this example are that questions are often composed of titles and bodies, and that we need to handle long answer texts. In addition to data from the widely popular forums mentioned above, cQA may also leverage data from a large quantity of more specialized expert forums, some of which are available in different languages. One example is the DATEV-community[2], in which tax consultants answer questions about the German tax policy and automation in accounting. A common limitation of such expert forums is that they do not offer enough labeled data to train commonly used machine learning models. We present some of the most important challenges in more detail when we discuss the prior work in Section 2.3 and Section 2.4.

We note that there are many other tasks and application scenarios related to cQA forums, many of which are not concerned with automatic question answering but other aspects of these communities. The different tasks have been surveyed in great detail by Hoogeveen et al. (2018) and Srba and Bielikova (2016). Examples are question routing and expert finding (Li et al., 2019b) summarizing forum threads (Chan et al., 2012), determining the quality of questions (Li et al., 2012), and learning to ask clarification questions (Rao and Daumé III, 2018a). We note that this thesis aims to address the fundamental challenges that arise from the cQA setup that we outline here in this chapter. Therefore, we will not detail other tasks that exist in the broader context of cQA.

Automatically answering questions with cQA forum data requires addressing several subtasks. In the following, we first present the structure of a prototypical cQA system to motivate these subtasks regarding a practical setting. Later in this chapter, we then review previous work in cQA relevant in this context and present common data sources.

2.2 Structure of a Prototypical cQA System

We define the goal of a cQA system to automatically find suitable already existing answers to a user question—which we denote as *query question* here—in cQA forums. As in our example from Figure 2.2, existing questions in cQA forums usually consist of a title and a body. In contrast, in many practical setups the query question is short and concise, i.e., it does not contain a body. This would be the case, for example, when using a cQA system as part of a conversational agent, in a search engine, or as part of a digital assistant (to answer compact query questions such as "How can I live a healthy life?"). We believe that these are realistic use-case scenarios, and thus design the structure of our prototypical system accordingly.

To locate, assess, and output relevant content in cQA forums regarding the query question, we divide the cQA system (which is a realization of the function f described before) into three operations, visualized in Figure 2.3. This yields a structure that corresponds to the subtasks of cQA that we address in different chapters of this thesis, and it is compatible with other research in this area, e.g., the SemEval 2017 shared task on cQA (Nakov et al., 2017) and related cQA systems (Hoque et al., 2016; Romeo et al., 2018).

[2] https://datev-community.de/; last accessed 24 June 2020.

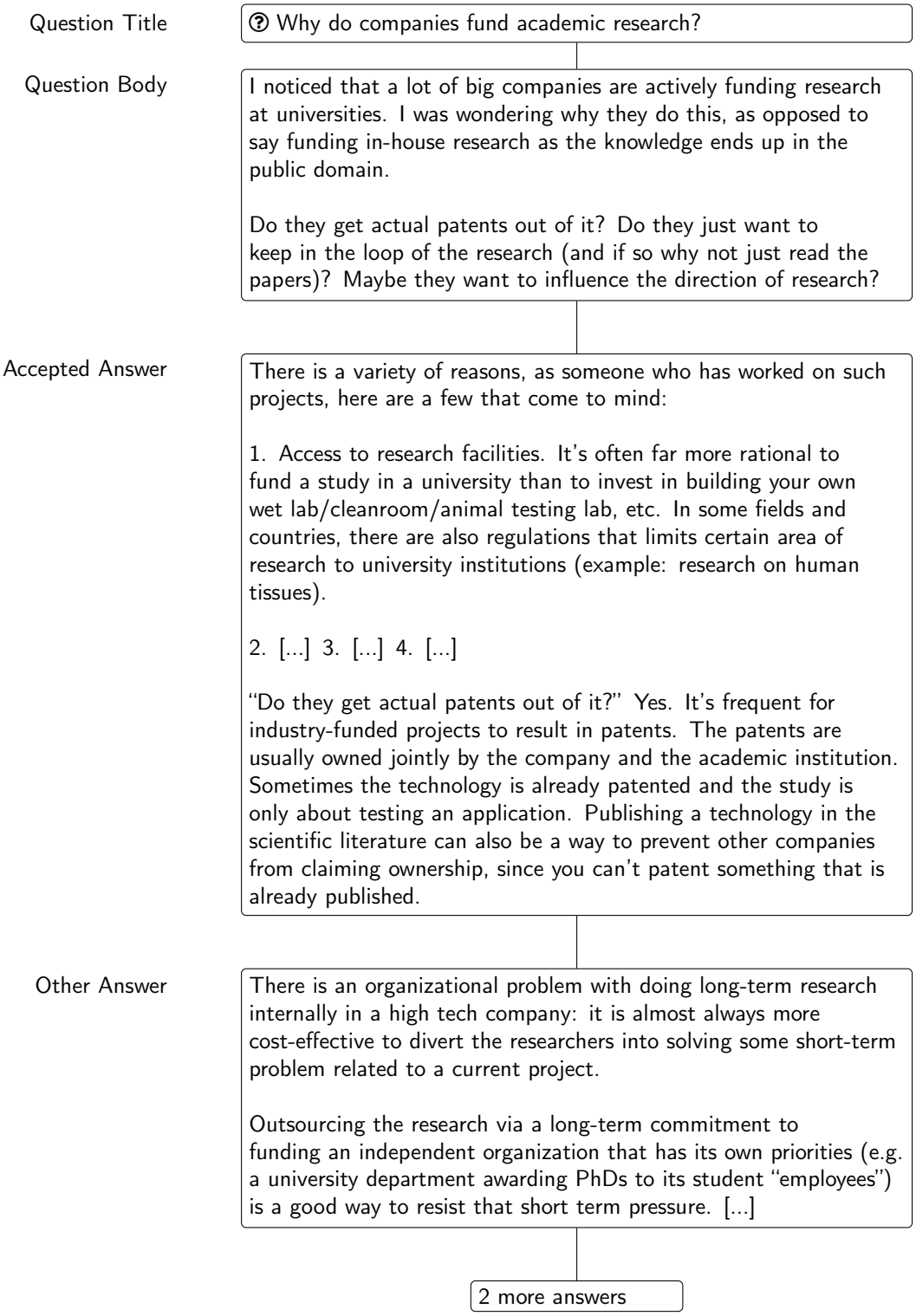

Figure 2.2: An example of a question in a cQA forum and two of four answers from community members. Questions usually consist of a short title and a longer body that includes additional details. Answers may be long texts and are not necessarily of high quality. The person asking a question can mark an answer as correct in most cQA forums. Source: https://academia.stackexchange.com/questions/43375/why-do-companies-fund-academic-research.

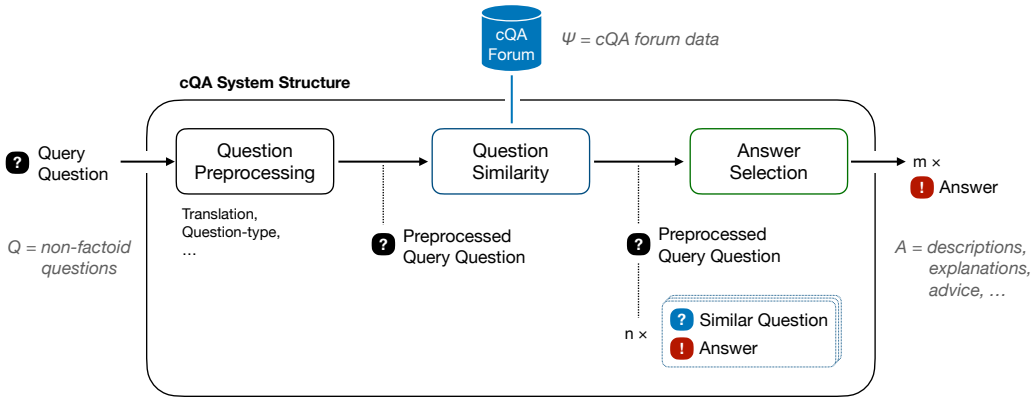

Figure 2.3: A visualization of the operations/components of our prototypical cQA system with an abstract view on the input and output data. The operations/components correspond to the subtasks in cQA that we deal with in this thesis.

Question Preprocessing. Several preprocessing steps such as tokenization, tagging (e.g., part of speech), or dependency parsing can be performed using standard NLP tools such as NLTK (Bird and Loper, 2004). More important in the context of this thesis are (1) question type classification, which could be used to select specialized models in the subsequent steps; and (2) the automatic translation of the query question to other languages—e.g., from German to English—in order to bridge the language gap between the system input and the cQA forum data.

Question Similarity. Given a preprocessed query question, the system now finds similar questions in cQA forums that have already been answered—e.g., a question such as the one shown previously in Figure 2.2. Usually, we perform a retrieval step followed by a re-ranking of the results (e.g. Lei et al., 2016a; Nakov et al., 2017). For instance, we can use a standard search engine such as ElasticSearch[3] with BM25 (Robertson and Zaragoza, 2009) to retrieve a set of candidate questions from the cQA forum. These candidate questions are then re-ranked according to a question similarity model with regard to the input question. The reason for relying on these two subsequent steps is computational efficiency; while BM25 is very efficient and can be applied to a large question corpus, more effective neural models for question similarity are much more computationally demanding.

Answer Selection. Based on the preprocessed query question and the most similar questions identified in the previous step, we now select relevant answers from those similar questions. The answers are evaluated and ranked with respect to the query question. The most suitable answers—the exact number depends on the use case—are then returned to the user. We only consider high-quality answers here in this thesis, e.g., those marked as accepted by the forum community. Therefore, we do not deal with a separate step to automatically identify and discard low-quality content (e.g., spam).

We have chosen this structure for demonstration purposes, and point out that we

[3] https://elastic.co; last accessed 29 Jan. 2021.

Question Similarity	Answer Selection	Semantic Textual Similarity
• Compares a query question with several candidate **questions**.	• Compares a query question with several candidate **answers**.	• Compares two **sentences**.
• Real information need, **long questions** (multiple sentences, title/body, subquestions).	• Real information need, **long/partial answers** (detailed descriptions, explanations, advice etc.).	• Typically **declarative sentences** (some instances may include short questions).
• Tests whether two questions express the **same information need**.	• Tests whether an answer expresses information **relevant to** the information need of a question.	• Tests for semantic equivalence and **meaning overlap**.
• Measures **ranking performance** with binary judgements.	• Measures **ranking performance** with binary judgements.	• Measures **correlation** with graded human judgements.

Table 2.2: Some of the most important similarities and differences of question similarity, answer selection, and semantic textual similarity tasks.

could design cQA systems in various ways. Examples can be found in the submissions to the TREC LiveQA shared task (Agichtein et al., 2015) where systems answer questions from Yahoo! Answers in (near) real-time. Besides, one might add additional operations—e.g., to summarize multiple answers (Song et al., 2017a; Chan et al., 2012). Our system structure nevertheless represents a realistic setting and can be used in practice: we have implemented an end-to-end cQA system based on this structure as part of our BMBF-funded SoftwareCampus project "Intelligent Search in the Social Web" (3/2018–3/2020, led by Andreas Rücklé). A master thesis (Lichtblau, 2020) has successfully used it as a basis for a user-centered study to determine important properties of cQA systems from a user perspective.

The central advantage of the structure presented here is that the different operations correspond to distinct subtasks of cQA, which we address throughout this thesis. In the following, we go into more detail on **question similarity** and **answer selection** and outline common challenges and some of the previous work. We further contrast these subtasks in Table 2.2. More specific background information is given in the respective thesis chapters.

2.3 Question Similarity

We define the task of question similarity as follows. Given a query question $q \in Q$, a set of n candidate questions $Q' = \{q'_0, q'_1, \ldots, q'_n\} \subset Q$, and a function $rel : Q, Q \rightarrow \{0, 1\}$ indicating whether two questions are similar or not. Our goal is to find a model that produces an optimal ranking of the candidates in Q' with respect to rel.

This raises the question of what *similarity* actually means and how it is defined. We find that this depends on the specific dataset and how it was created. This heterogeneity is also evident in the different names used to refer to the task (often with slightly nuanced task formulations). For example, "Question Similarity", "Duplicate

Question Detection", "Question Retrieval" or "Question Re-Ranking" are common names, sometimes used synonymously. In the interest of better comprehensibility, we refer to them as *question similarity* in this thesis (arguably the most universal of the four names) and will briefly outline the nuanced task formulations when we present common datasets in Section 2.3.1.

As question similarity performs a semantic comparison of two texts, it is also related to the task of (general) semantic textual similarity (STS, Cer et al., 2017). However, there exist several differences. (1) STS compares individual sentences for meaning overlap, whereas question similarity compares multiple questions to find pairs that express the same information need. (2) Question similarity often deals with long and complex questions (which we discuss in more detail later) that contain detailed descriptions, whereas STS typically compares declarative sentences (with few instances containing short questions). (3) Question similarity is often performed for specialized topics such as programming questions. Table 2.2 includes additional differences. This can make it difficult to apply standard STS techniques to the task of question similarity. For example, sentence embeddings such as the ones induced by a pre-trained InferSent model are very effective when applied to STS (Conneau et al., 2017), but in Chapter 5, we find that they may not be directly applicable to cQA. Nevertheless, sentence embeddings can achieve good results in question similarity tasks if the models are adapted to the specialized cQA target domain (Poerner and Schütze, 2019) and if they are trained on forum data (Cer et al., 2018).

A particularly important concept, which is necessary for understanding most work on question similarity, is the definition of so-called **duplicate questions**. Despite the connotation that the word "duplicate" may imply within technical or programming domains, duplicate questions rarely represent exact string matches. Rather, they refer to two questions that require the same—or sufficiently similar—answers, as it might be the case for questions that are merely paraphrases of each other. As one can imagine, there is no universal definition of duplicate questions, but rather annotation guidelines. For example, within the StackExchange network, there exist several resources on how to identify duplicate questions manually and how to annotate them. They suggest that there exist different variants of duplicate questions, including "borderline duplicates" which may or may not be considered duplicates, and a decision is to be made by the community.[4] We present an example of duplicate questions in Figure 2.4. It is important to note that we commonly use such data to train question similarity models, and often rely on it to evaluate such models as well—depending on the used dataset.

2.3.1 Data Sources and Datasets

We will now turn to the description of question similarity datasets and common data sources on which such datasets are based. This list is by no means exhaustive. There is a large body of works on question similarity—which we will review later—many of which are based on datasets that have never been used again or have not

[4] See, e.g., https://stackoverflow.blog/2009/04/29/handling-duplicate-questions/ and https://meta.stackexchange.com/questions/10841/; last accessed 16 Sept. 2020.

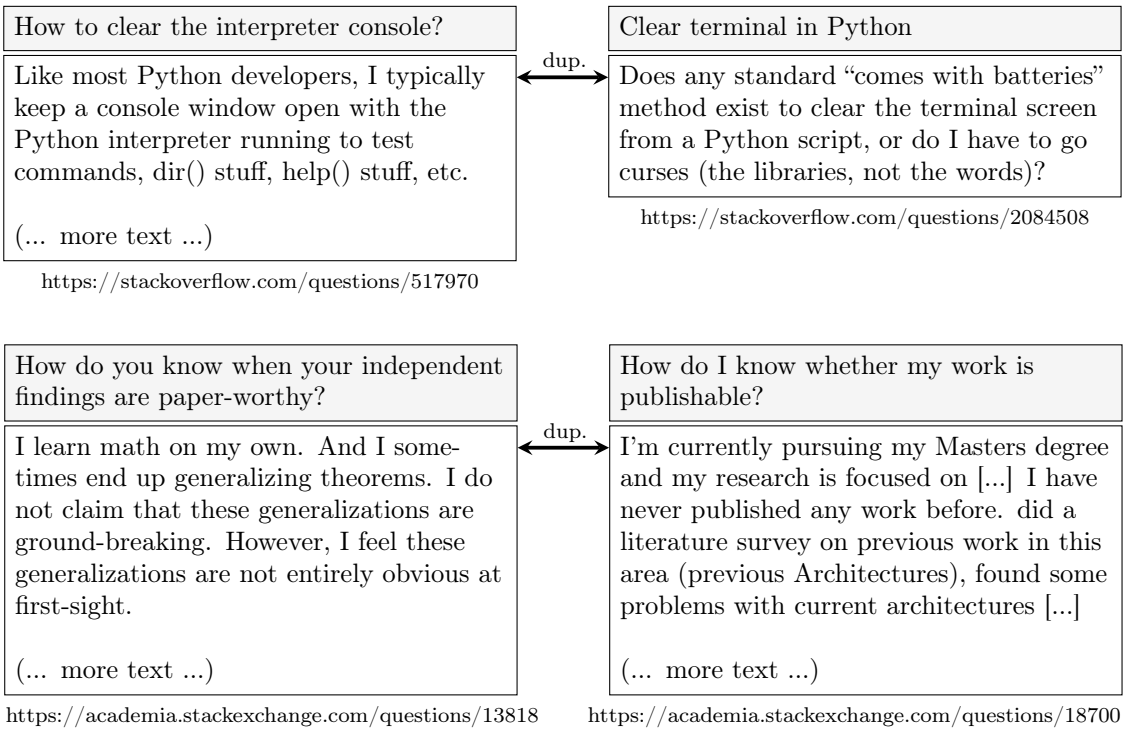

Figure 2.4: An example of two pairs of duplicate questions (each consisting of a title and a body) from different platforms annotated by the community.

been published for licensing[5] or other (unknown) reasons. Furthermore, the number of cQA forums is large, e.g., the StackExchange network alone consists of more than 170 forums on different topics,[6] which illustrates the wide variety of possible data sources. Even more specialized forums are available in other languages, e.g., DATEV Community, as we have described previously.

For conciseness reasons, we concentrate here within this overview only on datasets and data sources that have been frequently used in recent years. This overview will provide a succinct picture for this thesis—in which we do not focus on dataset creation. We list the most common data sources below.

- **The StackExchange Network** consists of more than 170 forums (as of September 2020) with ones dedicated to programming and technical topics (e.g., StackOverflow, AskUbuntu), culture and recreation (e.g., travel, history), life and arts (cooking, photography), and many more. The number of questions in forums is highly imbalanced, e.g., StackOverflow (the most popular one) contains more than 20 million questions, whereas the forum on Linguistics contains only around 8.5 thousand questions.[7] Questions always contain a title and a body. One particular advantage of StackExchange is the high quality of the questions and answers: an active community moderates

[5] In the course of this work, we considered, for example, working with data from the popular German cQA forum GuteFrage.net. We discarded such plans because our datasets' publication would not have been possible due to licensing restrictions.

[6] https://stackexchange.com/sites; last accessed: 17 Sept. 2020.

[7] As of September 2020. https://stackexchange.com/sites; last accessed 17 Sept. 2020.

the content.[8] Besides discarding low-quality content, community members also annotate questions as duplicates and label accepted answers (i.e., indicating that they fulfill the information need of the question). Because such annotations are time-consuming, they are only available in larger quantities for the more popular technical forums. A publicly available benchmark dataset that uses duplicate annotations from 12 large cQA forums (mostly technical) is CQADupStack (Hoogeveen et al., 2015). Similarly, Dos Santos et al. (2015) published a question similarity dataset based on questions and duplicate annotations from the AskUbuntu forum. This has been later extended with manual annotations for the evaluation splits by Lei et al. (2016a) to ensure a more accurate model evaluation. Several other works have used similar data from StackExchange (e.g., Zhang et al., 2017a; Karan and Šnajder, 2018; Shah et al., 2018; Poerner and Schütze, 2019; Shirani et al., 2019).

- **Yahoo! Answers** is another highly popular cQA forum, which contained around 300 million questions in 2012 (on a large number of topics) according to some reports.[9] It considerably differs from StackExchange in at least two attributes. First, Yahoo! Answers' quality can be low (depending on the topics). For instance, one can easily find a plethora of questions titled "Is this funny?"[10] and some questions are seemingly only asked to start a conversation, see figure 1.1 in (Hoogeveen et al., 2018). Further, Adamic et al. (2008) wrote in their analysis on Yahoo! Answers user activities that "*One may dispute the validity of the knowledge in Alternative Science and even the degree of knowledge in Celebrities. However, the YA participants believe this is knowledge, and they are certainly exchanging it.*" The investigation of the question and answer quality on Yahoo! Answers has thus been an important research topic (e.g., Fichman, 2011; Liu and Agichtein, 2008; Harper et al., 2008). Secondly, Yahoo! Answers does not contain annotated duplicate questions. This makes it hard to train and evaluate machine learning models for question similarity, and manual annotations are usually collected to assess the model performance. Despite these shortcomings, and due to the large size, many works have used data from this forum (e.g., Zhou et al., 2015; Wang and Chua, 2010; Zhou et al., 2011, 2013; Wang et al., 2009).

- **Quora** is a relatively new cQA forum with 61 million questions as of November 2019.[11] Similar to StackExchange, it is highly moderated, and the (perceived) quality of the questions and answers is very high. The most notable differences to other platforms are that there exist only question titles (no bodies). Further, the community can edit questions and answers, if deemed appropriate. Users are not anonymous, which further reduces spam content. Quora had become popular in the research community in 2017, when they released a large

[8] See, e.g., https://stackoverflow.blog/2009/05/18/a-theory-of-moderation/; last accessed 17 Sept. 2020.

[9] https://searchengineland.com/yahoo-answers-hits-300-million-questions-but-qa-activity-is-declining-127314; last accessed 17 Sept. 2020.

[10] https://answers.search.yahoo.com/search?p=is+this+funny; last accessed 17 Sept. 2020.)

[11] https://www.quora.com/How-many-questions-have-been-asked-on-Quora-1; last accessed 17 Sept. 2020.

corpus of manually annotated duplicate questions.[12] This has subsequently been integrated into the popular GLUE benchmark for natural language understanding (Wang et al., 2018), and other researchers have used derivatives of this data (e.g., Shah et al., 2018; Uva et al., 2018).

- **Qatar Living Forums** is relatively well known in the community because it has been the data source for the SemEval shared tasks on cQA (Nakov et al., 2017) from 2015–2017. The forum is otherwise relatively limited in its topics, i.e., it is aimed at expatriates who work and live in Qatar (or plan to do so). It is comparable to Yahoo! Answers in that it is not (or only lightly) moderated. The SemEval challenge organizers provide a small number of high-quality annotations for question similarity and other cQA tasks (267 query questions for question similarity in 2017).

As we can see, the data sources alone are extremely heterogeneous in terms of their quality, available annotations, and size. We also note that other data sources exist and have been used before, e.g., Reddit[13], WikiAnswers (Fader et al., 2013), QA data from Amazon products (McAuley and Yang, 2016), Baidu Zhidao (He et al., 2018), and more. We do not use any of these other data sources in this thesis and refer to the detailed surveys of (Hoogeveen et al., 2018; Srba and Bielikova, 2016) for more details.

Based on the above listed data sources, several research datasets have been created. In Table 2.3, we present some of the most recent ones that are most relevant in regard to this thesis. Here we also see the nuanced formulations of the question similarity tasks. In the course of this thesis, we do not focus rigidly on a few datasets. Rather, we have chosen to select the datasets for our experiments according to (1) whether they are openly available, (2) if there exist reproducible baselines, (3) the timeliness of the datasets for better comparability with more recent work, and (4) which challenges the datasets present.

2.3.2 Challenges

There are various challenges related to question similarity, and we outline here the most relevant ones regarding this thesis.

Overcoming the lexical gap. The most common challenge mentioned in the context of question similarity is overcoming the so-called lexical gap, sometimes also referred to as lexical chasm (see, e.g., Berger et al., 2000; Jeon et al., 2005). This means that two questions can express the same information need with an entirely different vocabulary. It is straightforward to see that the lexical gap has a large impact on bag-of-words approaches, where we fundamentally count word co-occurrences. Modern approaches naturally overcome this to a certain extent, e.g., by leveraging and learning dense vectors that can be used to infer semantic

[12] It contains roughly 400 thousand question pairs with annotations. https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs; last accessed 18 Sept. 2020.

[13] The extensive data can be accessed (not officially supported and subject to prohibitive licenses) on https://pushshift.io/; last accessed 18 Sept. 2020.

Dataset	Description
CQADupStack Source: StackExchange (Hoogeveen et al., 2015) License: Apache 2.0	This is one of the first openly available question similarity datasets with a large number of annotations. It contains data from 12 cQA forums, each with 12 000–52 000 annotated question pairs for training. This dataset relies on duplicate question annotations from the StackExchange community. It is primarily used for *question retrieval*: given a query question and the whole cQA forum, the task is to retrieve a duplicate.
⋆AskUbuntu Source: StackExchage (Lei et al., 2016a) License: Apache 2.0	This is an extension of an earlier dataset based on the AskUbuntu data of Dos Santos et al. (2015). In contrast to CQADupStack, it contains data from one forum only (with 12 000 annotated question pairs for training). The training splits are likewise based on duplicate question annotations. The biggest differences to CQADupStack is that during evaluation, one is given a query question and 20 potentially related questions (which were retrieved with a standard search engine). The 20 candidates were manually annotated for similarity to the query question, and the task is then question *re-ranking*. This makes it suitable to apply computationally demanding machine learning models.
⋆SemEval-2017 (3b) Source: Qatar Living (Nakov et al., 2017) License: "The datasets are free for general research use."	The SemEval challenge on cQA has first started in 2015. The organizers have subsequently annotated different parts of the Qatar Living Forums (which we described in more detail before). Task 3b contains training and evaluation data for question similarity. Similar to AskUbuntu, this is a re-ranking setup, i.e., for a query question, they retrieved ten potentially similar questions with a search engine and manually annotated them for similarity. Different from AskUbuntu, both the training and evaluation splits were manually annotated. The training set is thus relatively small (containing 267 query questions with ten annotations each).
⋆StackExchange DQD Source: StackExchange (Shah et al., 2018) License: Apache 2.0	Similar to AskUbuntu and CQADupStack, this dataset uses duplicate annotations (from four forums, each with 9106 annotated question pairs for training). During evaluation, we are given a query question and 100 potentially similar questions. In contrast to AskUbuntu, where potentially relevant questions were retrieved with a search engine, this dataset draws them randomly from the corpus (with no additional annotations). The challenge is now to re-rank the potentially similar questions such that we "detect" the duplicate. This task formulation is referred to as *duplicate question detection*.
Quora Dataset Source: Quora License: Subject to Quora Terms of Service (proprietary).	Contains 400 000 manually annotated question pairs (as duplicate or no-duplicate), which are not specific to any particular topic. Because this is based on Quora data, it contains only question titles (no question bodies). Tasks based on this dataset (e.g., Wang et al., 2018) are often to classify whether two questions are duplicates or not (which is also commonly referred to as *duplicate question detection*). https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

Table 2.3: Question similarity datasets that we consider as most important in the context of this thesis (re-usable ones of sufficient size and quality). Datasets that we experiment with are marked with ⋆. We do not include Yahoo! answers datasets here as they are extremely heterogeneous and often cannot be re-used as they contain manual annotations specific to the retrieved results for one particular model (see, e.g., section 2 in (Hoogeveen et al., 2018)).

relationships (embeddings of words or texts). Nevertheless, the lexical gap remains a central challenge in question similarity, which requires a rich knowledge of similar words, synonyms, antonyms, etc. The lexical gap is also a fundamental challenge for identifying paraphrase sentences (Dolan and Brockett, 2005), or when determining semantic textual similarity.

Long questions. As we have seen earlier, questions in cQA forums usually consist of titles and bodies. Combining information from both the title and the body can provide additional signals to better determine the similarity of two questions (Lei et al., 2016a). However, this can also pose challenges to models, e.g., these additional descriptions might include unnecessary details, which may require more fine-grained approaches (Wang et al., 2010). In this thesis, rather than developing specialized architectures for this purpose, in Chapter 6, we *leverage* the connection between question titles and question bodies in cQA to train question generation models and for self-supervised training of question similarity models.

Complex questions. Complex questions are a common phenomenon in question answering research but they pose different challenges to factoid QA and cQA. In factoid QA, complex questions are often ones that consist of many relations, which makes it challenging to parse them (Sorokin and Gurevych, 2018). Talmor and Berant (2018) provide an example of a complex question: "What city is the birthplace of the author of 'Without end', and hosted Euro 2012?". Such questions are referred to as complex because of their compositionality. One may need to answer several sub-questions before inferring a final answer. This challenge can be addressed with sequential approaches applied to already decomposed questions (Iyyer et al., 2017), or by automatically learning to decompose complex questions into simpler operations that can be solved separately (Talmor and Berant, 2018; Wolfson et al., 2020). Challenges that arise from the compositionality of questions can also be addressed by learning span-based semantic parsers, where questions are transformed into so-called "span trees" that represent programs composed over a complex question (Herzig and Berant, 2020). When executed, these programs answer the question.

In contrast, complex questions in cQA may take other forms. For instance, we often have to deal with underspecified questions that do not provide enough information to precisely answer them (e.g., asking for the cause of a program crash without specifying the used program version). This motivates the task of learning to ask clarification questions (Rao and Daumé III, 2018a), but it also illustrates that we need to be able to handle such noise when determining question similarity. Questions in cQA may also explicitly state several sub-question, see the question body in Figure 2.2. This is different from sub-questions in factoid QA, as it may not be possible (or required) to explicitly infer knowledge by re-combining their answers. Handling such complex cQA questions can, e.g., be addressed by segmentation (Wang et al., 2010).

Expert domains. There are many cQA forums with different topics—which we refer to as domains in this thesis. For instance, the StackExchange network consists of more than 170 forums of different domains, many of which cover expert topics such as programming, cryptography, astronomy, academia, finance, and cooking.

Approaches	Advantages	Disadvantages
Bag-of-words. Query Likelihood.	Fast to compute. Training-free.	Affected by the lexical gap.
Translation-based models.	Determine the semantic similarity of words.	Self-translation problem.
Tree Kernels.	Leverage the syntactic structure of texts. Work with small training data.	Need a separate component generating parse trees. Outperformed by neural networks.
Word embedding similarity. Average word embeddings.	Training-free or easy to fine-tune. Determine the semantic similarity of individual words or texts.	Ignore word order and relations between words in a text.
Representation learning with CNN/LSTM/RCNN/Transformers etc.	Good performance. Learn local (CNN) or global relations (LSTM, ...) between words in a text.	Require a large amount of training data.

Table 2.4: An overview contrasting some of the most common types of approaches to question similarity.

This variety of highly specialized domains can make it challenging to obtain universally applicable and, thus, re-usable models. However, this also provides us with the opportunity to broadly study the transfer of approaches. We address this in Chapter 7, where we investigate the zero-shot transfer capabilities of models with 140 different source domains.

No training data. Some of the expert domains in cQA do not contain enough annotated duplicate questions for supervised training of question similarity models. Furthermore, some widely popular forums such as GuteFrage.net or Yahoo! Answers do not contain duplicate annotations *at all*. Data scarcity is a central bottleneck for neural networks and makes it necessary to leverage alternative training methods. We investigate methods for training question similarity models without labeled data in Chapter 6.

Overcoming the language gap. Large amounts of data from cQA forums are often only available in English. However, this data could also be useful to answer questions from users in other languages. To make the large amounts of information in the cQA forums more widely accessible, we need to overcome the language gap, i.e., dealing with query questions in a language other than the data source's language. Overcoming the language gap can be even more difficult in expert domains where machine translation might not work well. We address this problem in Chapter 4 by obtaining cross-lingual sentence representations and improving neural machine translation for question similarity with back-translation.

2.3.3 Previous Approaches

In the following, we give an overview of what we consider to be preliminary question similarity work for this thesis. We summarize previous approaches in two categories: (1) methods based on bag-of-words and word similarities; (2) approaches that use neural networks. Table 2.4 contrasts the most common types of approaches.

2.3.3.1 Bag-of-Words and Word Similarities

Approaches that are still considered strong baselines are bag-of-words approaches based on term and document frequencies, e.g., BM25 (Robertson and Zaragoza, 2009). In essence, they count and weight co-occurring words in two questions. Similarly, language modeling approaches such as query likelihood (Ponte and Croft, 1998) predict how likely a query will be generated for a document (here: another question). These approaches are strongly affected by the lexical gap, i.e., if two questions are semantically equivalent but use a different vocabulary, the determined similarities will be small. The reason is that they fail to model the semantic similarity of words that are not co-occurring in the two texts.

Translation-based models (Berger and Lafferty, 1999; Berger et al., 2000) try to overcome this and leverage monolingual statistical translation models to determine the probability that one sentence (or passage) is a translation of another sentence. Instead of considering two natural languages such as English and German, translation-based approaches in cQA learn monolingual word similarities by considering a question language and an answer language (both usually in English). Such methods have been applied to determine both question similarity and question-answer similarity (e.g., Jeon et al., 2005; Riezler et al., 2007; Bernhard and Gurevych, 2009; Zhou et al., 2011). A significant disadvantage of this concept is the self-translation problem, where a term in the question might be translated into the same term with a low probability. Xue et al. (2008), therefore, suggest combining both the language model and the translation approach to determine question similarity. Another promising extension is the inclusion of question category information, i.e., learning specific models for certain parts of a cQA forum representing different topics (Cao et al., 2012). Furthermore, some works have learned topic models representing latent topics of questions and combine them with translation and language models (Zhang et al., 2014; Ji et al., 2012).

Finally, other approaches determine question similarity based on tree kernels, which compare syntactic substructures of parse trees, i.e., they determine the *syntactic* similarity of questions based on pairwise patterns (Filice et al., 2016; Da San Martino et al., 2016).

Most importantly, these approaches have recently been outperformed by neural networks, if sufficient training data is available (e.g., Uva et al., 2018; Wang et al., 2017a; Lei et al., 2016a). The availability of training data is, however, a critical bottleneck of neural approaches.

2.3.3.2 Approaches Based on Neural Networks

Neural networks have achieved unprecedented advances in many areas of NLP, including cQA and question similarity tasks. Early methods are based solely upon word embeddings that encode the meaning of words in dense vectors—e.g., the skip-gram model (Mikolov et al., 2013a) that yields word2vec embeddings which are suitable for determining semantic (word) similarities. For instance, Zhou et al. (2015) learn word embeddings with a variant of the skip-gram model that is regularized such that embeddings of words that usually occur in questions with the same topics (determined by category annotations) are closer in the embedding space. They then aggregate the embeddings of all words in the questions with a Fisher Kernel (Sánchez et al., 2013) and compute the dot product between the resulting representations to measure question similarity. They thereby outperformed many of the previously described more traditional approaches, e.g., combined language and translation based models. Gillick et al. (2018) learn word embeddings with a deep averaging network, which improves upon average word2vec embeddings for end-to-end question retrieval. Others compute soft cosine similarity (Sidorov et al., 2014) between questions based on word embedding similarities (Charlet and Damnati, 2017)—which outperforms many feature-based approaches in the question similarity task of the SemEval 2017 cQA shared task (Nakov et al., 2017).

The skip-gram model and word embeddings are simple and shallow examples of approaches based on neural networks. The large majority of more complex models use embeddings as inputs to a neural network that learns representations of the question text, e.g., learning the local or global relationships between words—or, more recently, sub-word units. Many of them learn dense vector representations, which are then compared with some similarity measure, e.g., cosine similarity. We utilize some of these approaches later in Section 4.2 and Chapter 6.

For instance, Dos Santos et al. (2015) map the words of a question to their word embeddings and use them as an input to a convolutional neural network (CNNs, e.g., Kim, 2014) that learns local context features over word n-grams. This is followed by global max-pooling to infer a fixed-size dense vector representation for the question. They then determine the similarity of two questions by calculating the cosine similarity between their representations. This model has later been extended by Lei et al. (2016a) with a gating mechanism over convolutions—i.e., adding a recurrent operation and thus including global contextual information during representation learning. In combination with an unsupervised pre-training method, they achieve considerable improvement over several bag-of-words approaches and neural baselines. Likewise, one can use long short-term memory networks(LSTMs, Hochreiter and Schmidhuber, 1997) instead of CNNs for representation learning (Shah et al., 2018). Similar approaches have been widely adopted in cQA with different extensions whose explanations go beyond the scope of this section (Gupta et al., 2018; Bonadiman et al., 2017; Nicosia and Moschitti, 2017; Romeo et al., 2016). Importantly, these models are typically trained with duplicate questions and require sufficient amounts of training data, typically thousands of instances. Such data is, however, not available in many realistic scenarios, e.g., for a large quantity of smaller cQA forums.

When in-domain training data is scarce, other strategies to train neural models are required. If there exist some labeled question pairs, e.g., thousands, one can first train a less data-hungry non-neural model and apply it on unlabeled question pairs to generate large amounts of (noisy) labels for training (Uva et al., 2018). Another alternative is to use labeled question-answer pairs instead of question duplicates for weakly supervised training (Wang et al., 2017a; Qiu and Huang, 2015). However, even thousands of labeled question pairs can be difficult to obtain, and some forums do not contain enough labeled question-answer pairs (which we describe and address in greater detail in Chapter 5). Other methods thus use unsupervised training with encoder-decoder architectures (Zhang and Wu, 2018; Lei et al., 2016b), or adversarial domain transfer where the model is trained on a source domain and adversarially adapted to a target domain (Shah et al., 2018). However, such approaches typically fall short of the performances that are being achieved with in-domain supervised training. We note that this is one key challenge for current neural models to question similarity, and we address this in Chapter 6 and Chapter 7.

Another crucial challenge is to perform question similarity cross-lingually, i.e., to determine the similarity of two questions posed in different languages. Previous approaches extend question similarity models with cross-lingual components, e.g., Da San Martino et al. (2017) use a combination of a cross-lingual tree-kernel and machine translation, and Joty et al. (2017) learn language invariant representations with a feed-forward network and adversarial training. Importantly, none of them have studied cross-lingual question retrieval in highly specialized cQA forums of programming and operating systems topics, which are extremely popular. Technical cQA forums pose crucial challenges to cross-lingual question similarity due to the highly specialized vocabulary. We investigate this in Section 4.2.

In summary, there has been considerable research in recent years towards obtaining better models for question similarity, from early translation-based models to current neural networks. Nevertheless, several key challenges remain for us to address in this thesis, such as to train models without labeled data and to study realistic cross-lingual scenarios. We have now outlined the previous work, and will provide more details in the course of this thesis as needed.

2.4 Answer Selection

After finding similar questions in the cQA forum, we select one or more of their answers, i.e., ones that can answer our query question. Typically, we re-rank a list of candidate answers according to our query question and select the top-n answers to be returned (often only the best one). We define the task of answer selection accordingly as follows. Given a query question $q \in Q$, a set of n candidate answers $A' = \{a'_0, a'_1, \ldots, a'_n\} \subset A$, and a function $rel : Q, A \rightarrow \{0, 1\}$ indicating whether a candidate answers the question. Our goal is to find a model that produces an optimal ranking of the candidates in A' with respect to rel.

As it is the case for question similarity, there exist different nuanced task formulations for answer selection. For example, this task has also been formulated such that a model identifies the best answer in a single cQA forum thread or automatically

predicts the quality of different users' answers (e.g. Shah and Pomerantz, 2010; Zhu et al., 2009; Hu et al., 2013; Suggu et al., 2016). However, this is more related to the scenario of automatic content moderation than to the automatic question answering, and it can be heavily impacted by metadata, e.g., how many accepted answers a user has posted in the past (Agichtein et al., 2008). In addition, as described before in Section 2.3, many modern cQA forums successfully moderate the content, and quality prediction would then only be helpful to identify spam posts more quickly. Finally, the task formulation we have outlined above is the same as in other non-factoid answer selection datasets that do not rely on cQA forum data, which means that we can more broadly study our approaches' generalization capabilities.

Finally, this task formulation is closely related to question similarity, as introduced before. However, there exists at least one significant difference: Question similarity determines whether two questions express the same information need whereas answer selection determines if an answer expresses information relevant to the question (see our task overview in Table 2.2). Thus, different approaches may be better suited to solve these two tasks. For instance, unidirectional attention mechanisms have been primarily applied to answer selection (Tan et al., 2016; Wang et al., 2016a), leveraging the directional question-answer relationship. One can also learn separate neural network components to infer more accurate representations of question and answer texts (e.g., Feng et al., 2015). Both are neither optimal for question similarity nor general semantic textual similarity.

2.4.1 Data Sources and Datasets

Common data sources for cQA answer selection are mostly the same as in Section 2.3.1—popular ones are StackExchange, Yahoo! Answers, and Qatar Living Forums.[14] We, therefore, omit a detailed description here.

We present the research datasets that are most relevant in the context of this thesis in Table 2.5. Once again, we see the heterogeneity of the datasets. This concerns, for example, the quality and annotations and the test set creation. Some datasets chose candidate answers randomly (which is arguably far from realistic), others predict the quality of answers given to the same question, and, most realistically in our context, some obtain candidate answers by retrieving them with a search engine.

Throughout this thesis, we choose the datasets for our experiments according to how well they fit the setup we have outlined before in Section 2.2. For example, we also investigate the effectiveness of our approaches on WikiPassageQA (Cohen et al., 2018), even though it does not contain cQA forum data, because it models a realistic related scenario. Notably, many of the newer datasets were created in parallel to the work we conducted in this thesis, which illustrates that realistic cQA and non-factoid answer selection datasets were often not (publicly) available in the past or existing datasets addressed different challenges. For instance, some of the earlier datasets are not public (e.g., Bogdanova et al., 2017; Severyn and Moschitti, 2012; Verberne et al., 2010) or not accessible anymore (e.g., Jansen et al., 2014).

[14] Quora, on the other hand, has not released any publicly available forum data with answers.

Dataset	Description
⋆**InsuranceQA** Source: insurancelibrary.org (Feng et al., 2015) License: "This dataset is provided as is and for research purpose only."	This benchmark is based on data from a specialized forum in which only insurance practitioners provide answers to user questions. The domain is very narrow (targeting US insurance customers) but the quality of answers is high. For each question, this dataset contains 500 candidate answers, of which typically one is correct. There exist two versions. Version 1 contains randomly chosen candidate answers (plus correct answers), and version 2 models a more realistic setup in which the candidate answers were retrieved with a search engine (BM25) using the question as a query. Both versions contain roughly 12k questions in the training splits.
SemEval-2017 (3a) Source: Qatar Living (Nakov et al., 2017) License: "The datasets are free for general research use."	In addition to question similarity, the SemEval shared task also includes a subtask for answer selection. The dataset contains manually annotated answers from Qatar Living. There are two limitations with this dataset in the context of our setup: (1) candidates are answers from the forum thread of the query question, (b) many answers are of low quality. Thus, the task is more related to quality prediction.
Yahoo! QA Source: Yahoo! Answers (Tay et al., 2017) License: Unknown / subject to the "Yahoo Data Sharing Agreement."	This dataset has been created based on 63k questions and answers from Yahoo! Answers. For each question, four unrelated candidate answers have been randomly sampled from the top-1k results of a search engine (using BM25). It thus has two shortcomings in our settings: (1) the sampling of candidate answers does not represent a realistic setting, (2) the quality of answers can be low due to the data source being Yahoo! Answers. A more realistic variant is nfL6, which uses the same data but does not perform random sampling to select candidates (Cohen and Croft, 2016).
⋆**WikiPassageQA** Source: Wikipedia (Cohen et al., 2018) License: Unknown	A non-factoid answer selection dataset containing explanatory passages that answer crowd-sourced questions. Its structure is otherwise similar to InsuranceQA. There are several candidate answers (manually labeled) that correspond to the passages of a Wikipedia article for a question. WikiPassageQA contains 3332 questions in the training split.
⋆**StackExchange LAS** Source: StackExchage (Rücklé et al., 2019a) License: Apache 2.0	To complement the few existing datasets containing non-factoid questions with high-quality answers, in Chapter 5, we create five new datasets from apple, cooking, academia, travel, and aviation StackExchange forums. For a question, we consider its accepted answer as correct and collect negative candidates by retrieving accepted answers of similar questions (with BM25). We have created these datasets to provide a more thorough evaluation throughout this thesis, and other researchers have also re-used them. We provide more details in Chapter 5.
ANTIQUE Source: Yahoo! Answers (Hashemi et al., 2020) License: "ANTIQUE is publicly available for research purposes."	This very recent dataset adds rich annotations to Yahoo! Answers, with four labels from "does not make any sense" to "looks reasonable and convincing". The training set contains 2.4k questions with approximately ten annotated answers each (the highest voted answers of each question). Ten candidate answers were first retrieved from the collection with a union of different models and then manually annotated to construct realistic test sets. The recency of this dataset illustrates the need for high quality, publicly available cQA datasets.

Table 2.5: Non-factoid answer selection datasets that we consider as most important in the context of this thesis. Datasets that we experiment with are marked with ⋆.

Besides, there are other datasets that model distantly related settings and are worth mentioning for the sake of completeness. MS MARCO (Nguyen et al., 2016b) includes a passage re-ranking task (starting from version 2.1, released at the end of 2018) where questions are popular queries from the Bing search log and passages are from general web documents. Similarly, the Chinese DuReader dataset (He et al., 2018) uses Baidu search logs to obtain questions and the corresponding answers are summaries based on Chinese web pages or answers from the popular Baidu Zhidao cQA forum. NaturalQuestions (Kwiatkowski et al., 2019) contains factoid questions from the Google search log, and answers are extracted facts and long answer sentences annotated from top-ranked Wikipedia pages. TREC LiveQA (Agichtein et al., 2015) contains annotations from a research challenge in which end-to-end QA systems answered questions from real users on Yahoo! Answers—the answers were manually judged and are thus specific to individual heterogeneous systems.

2.4.2 Challenges

Some of the challenges that applied to question similarity are also present in answer selection, e.g., the lexical gap (questions and answers may use different vocabulary), handling a large number of expert domains, and dealing with different languages. In addition to that, we identify the following additional challenges.

Real information need and partial answers. In cQA we deal with real user questions and answers, which often represent complex information needs. Therefore, questions can sometimes only be partially answered, and even "good" answers might disregard certain aspects of a question. This makes it challenging to re-use approaches from other types of QA where the answers are clearly either correct or incorrect (e.g., factoid QA or for multiple-choice questions). For instance, the question in Figure 2.2 contains three additional (sub-)questions in the question body, and a relevant answer might not address all of them. Further, the accepted answer shown in this example could be relevant to other questions as well but might not match them perfectly. Dealing with such *partial answers* requires robust methods, e.g., models that identify the most important aspects of a question that should be included in a relevant answer. The existence of partial answers also implies that we can find multiple answers providing complementary information to a question (Omari et al., 2016).

Detailed answer texts. Answers to questions in cQA are often complex texts consisting of several sentences and containing detailed information. One reason for this is that *non-factoid questions* often cannot be answered concisely and require longer descriptions, explanations, or advice. Dealing with long answers is not exclusive to cQA and a more general challenge for non-factoid QA. For example, the average length of answers in StackExchange Aviation and WikiPassageQA is 281 and 153 tokens, respectively (words and punctuation). This is in sharp contrast to factoid answer *sentence* selection, e.g., in WikiQA (Yang et al., 2015) the answers contain an average of 25 tokens.[15] Long answers can pose crucial challenges for

[15] For more details see Table 3.1 and Table 5.1.

previous models that are often derived from factoid QA. We discuss and address such challenges in Chapter 3 and Chapter 5.

It is important to note that dealing with long *passages or documents* also poses challenges to other types of QA such as machine reading reading comprehension. Here, it is difficult to read and reason over long texts in order to *extract or generate* an answer. One possible solution is to automatically summarize the document prior to reading (Choi et al., 2017). Interestingly, this is similar to approaches in cQA that summarize texts of long answers prior to answer selection (Deng et al., 2020a).

Small training data. In cQA there is a considerable number of specialized expert domains, which leads to challenges in the availability of training data. Although the problem is less pronounced in answer selection as compared to question similarity—we rarely deal with the cases in which the community does not label good answers—we nevertheless often deal with *small* training data. This is especially problematic for data-hungry neural models. We address this in Chapter 5.

2.4.3 Previous Approaches

In the following, we give an overview of what we consider to be preliminary answer selection work for this thesis. We categorize this in: (1) feature-based approaches, and (2) approaches based on neural networks. Table 2.6 gives an overview.

It is important to note that many approaches to *question similarity* may also be applied to answer selection. Since we have already summarized them before, we omit their description here. The outputs—e.g., similarities determined by bag-of-words approaches, query likelihood, etc.—are often combined with other features and used in learning-to-rank models.

2.4.3.1 Feature-based Approaches

Many previous approaches leverage feature-based learning-to-rank models. For instance, Surdeanu et al. (2008) combine several hand-crafted features in a ranking perceptron (Shen and Joshi, 2005), including lexical similarity features, translation features, and correlation features with respect to web corpora. Higashinaka and Isozaki (2008) propose a corpus-based approach that automatically collects patterns of causal expressions for why-QA, which are then used as features for a rank SVM (Joachims, 2002). Verberne et al. (2010) propose several structural features such as main-verb overlap and determining the focus of the question. Surdeanu et al. (2011) explore linguistically motivated features, e.g., bag of semantic role tuples based on PropBank (Palmer et al., 2005). Severyn and Moschitti (2012) propose using tree kernels (Shawe-Taylor and Cristianini, 2004) to automatically encode relational features between syntactic trees of questions and answers. Yih et al. (2013) leverage lexical semantic relationships of individual words in question and answer sentences, e.g., synonymy and antonymy. Jansen et al. (2014) complement lexical semantics with features based on discourse markers and rhetorical structures for why and how-questions. Fried et al. (2015) extend this with higher-order lexical alignments, i.e., word associations over multiple hops.

Approaches	Advantages	Disadvantages
Feature-based (e.g., SVMs)	Integrate multiple diverse features, e.g., bag-of-words similarity, translation similarity, discourse features, metadata features. Work with small data.	Require manual feature engineering. Potentially very complex pipelines. No end-to-end learning.
Representation learning *without attention* (CNN, LSTM, ...)	Independent encoding of questions and answers. Better performance than feature-based approaches.	Do not scale well to long answers. Require a large amount of training data.
Attention-based representation learning (attentive CNN, attentive LSTM, ...)	Scale well to long answers. Typically better performance than representation learning without attention.	No independent encoding of questions and answers. Require a large amount of training data.
Relevance matching (e.g., compare-aggregate)	Often achieve the best performance by explicitly comparing all question-answer words.	No independent encoding of questions and answers. Computationally demanding. Require a large amount of training data.

Table 2.6: An overview contrasting some of the most common types of approaches to answer selection.

As we can see, there is a rich body of work regarding feature-based approaches, and similarly, complex feature sets including additional metadata features such as social user-based features and conversational features have been commonly used for general answer quality prediction (Agichtein et al., 2008; Shah and Pomerantz, 2010; Blooma et al., 2010; Nakov et al., 2015, 2016, 2017). Finally, it is important to mention that other work has also applied translation-based models to answer selection, which is similar to what we have previously outlined in the context of question similarity (e.g., Riezler et al., 2007; Bernhard and Gurevych, 2009).

This clearly illustrates that modeling complex relationships between questions and answers is crucial for achieving improvements in answer selection. We now turn to neural approaches, which can learn such relationships automatically without requiring manual feature engineering. These approaches have shown to achieve better performance scores as compared to many feature-based or bag-of-words approaches (Yu et al., 2014; Feng et al., 2015; Shen et al., 2015).

2.4.3.2 Approaches Based on Neural Networks

One of the first works that apply deep learning to answer selection is (Yu et al., 2014), who propose learning representations of questions and answers with a convolutional neural network (CNN) followed by mean pooling. Question and answer representations are then compared with a learned similarity function for scoring. This has later been extended by concatenating additional features to the question-answer representations and the similarity score, which are then fed to an additional

multi-layer perceptron for prediction (Severyn and Moschitti, 2015). Others instead compare learned representations with cosine similarity (Zhou et al., 2016a) or other similarity measures (Feng et al., 2015; Tay et al., 2018a), as well as different composition layers (Tay et al., 2017). Previous research also experimented with stacked models, which showed that deeper architectures can achieve improved performance scores (Wang and Nyberg, 2015; Rao et al., 2016). Especially for non-factoid answer selection, it has been shown that bidirectional LSTMs are more effective for learning question-answer representations as compared to CNNs (Tan et al., 2015). Importantly, dealing with long answer texts poses crucial challenges to such models, and as a solution, attention-based models have been proposed (Dos Santos et al., 2015; Tan et al., 2016; Wang et al., 2016a; Zhang et al., 2017b), which are also effective for answer sentence selection (Yin et al., 2016). These learn to focus on segments within the answer that are most related to the question and are therefore less affected by unnecessary details within the answer text. In Chapter 3, we formalize and compare such attention-based representation learning approaches in greater detail and propose a self-attentive model that learns to focus on text segments within the answer text without relying on information from the question.

Others apply matching approaches that first compare question and answer tokens and then aggregate this information to determine a score. For instance, Shen et al. (2015) compare all word embeddings of the question and answer with cosine similarity, thereby yielding a soft-alignment matrix. They then aggregate this information with a CNN followed by a multi-layer perceptron for prediction. Researchers have also extended this to explicitly model dissimilar segments (Wang et al., 2016b), various similarity functions (Wang and Jiang, 2017), and multiple merged attention mechanisms (Tay et al., 2018b). We go into more detail on such *compare-aggregate* approaches later in Chapter 5. Importantly, they often achieve better performances as compared to representation learning models, however, they can be more computationally demanding.

As we can see, there exists a plethora of different models and approaches to answer selection. However, with the increased effectiveness of neural models, new challenges arise. For instance, it is particularly challenging to train such approaches with small data. Furthermore, attention-based representation learning models cancel one of such approaches' most important properties: the independent encoding of questions and answers that allows pre-computing representations of all answers within a large corpus.

2.5 Chapter Summary

In this chapter, we introduced cQA and put it into the broader context of automatic question answering. We defined cQA such that it uses the data of web forums to automatically answer non-factoid questions. This implies some significant differences compared to other types of QA. For example, we deal with detailed questions and answers, often from specialized expert domains. Importantly, there exist a considerable number of other heterogeneous tasks that use forum data for other purposes, e.g., for summarization, quality prediction, and more.

We illustrated three essential operations on the example of a prototypical cQA system design: (1) preprocessing, (2) question similarity, and (3) answer selection. While preprocessing covers various NLP tasks—e.g., machine translation or question type classification—question similarity and answer selection refer to dedicated subtasks in the context of cQA.

Question similarity is used to find relevant other questions that have already been answered in cQA forums. We highlight that existing data sources and datasets are heterogeneous and that there exist many nuanced task formulations. Nevertheless, the critical challenges are often similar, e.g., dealing with the lexical gap, handling long and complex questions, addressing many expert domains, training models without labeled data, and overcoming the language gap.

Answer selection then assesses answers of these similar questions regarding the query question. This yields challenges such as dealing with real user questions that may be difficult to answer, properly handling long and detailed answers, and training models with small data.

We summarized the previous work, providing a compact overview for both cQA subtasks. Most notably, we found that more recent work leverages neural networks, which achieve better results than previous methods. Neural networks, however, impose critical bottlenecks. For example, it can be difficult to fine-tune them for our task when we have little labeled data, and it can be hard to learn suitable representations for long texts. These bottlenecks represent critical challenges in the context of cQA, and we address them in the remainder of this thesis.

Chapter 3

Attention Mechanisms for Learning Question and Answer Representations

In this chapter, we study approaches to learning dense vector representations of questions and answers. As we have outlined before in Section 2.4, we can improve neural models such as LSTMs and CNNs by incorporating attention mechanisms that allow the models to focus on the most important text segments. Attention mechanisms are especially useful for non-factoid answer selection in which we deal with long answers, often descriptions or explanations. However, common attention mechanisms introduce a dependency between the learned question and answer representations, thereby imposing crucial computational limitations. We discuss these limitations in this chapter and address RQ1 of this thesis:

> **RQ1: How can we learn effective and efficient representations of questions and answers?**

In Section 3.1, we study representation learning from the perspective of answer selection and propose an approach that overcomes the limitations of attention mechanisms (Dos Santos et al., 2015; Tan et al., 2016) through self-attentive importance weighting. In Section 3.2 we present an end-to-end cQA system, which allows researchers to explore and compare different attention mechanisms for answer selection interactively.

3.1 Self-Attentive Importance Weighting

Parts of this section have been previously published as listed below. Verbatim quotes from this publication are included in this section.

Andreas Rücklé and Iryna Gurevych: 'Representation Learning for Answer Selection with LSTM-Based Importance Weighting', in: *Proceedings of the 12th International Conference on Computational Semantics* (IWCS 2017), September 2017.

My contributions: Neural network design ("LW" importance weighting), implementation, experimentation, analysis.

Approaches to answer selection commonly rely on representation learning models to encode questions and answers as dense vectors, i.e., semantic representations of the texts. These representations are comparable and can be used to determine whether an answer expresses information relevant to the question by calculating their representations' similarity. We visualize this common approach in Figure 3.1 and note that we deal with representation learning models frequently throughout this thesis.

Recall from Chapter 2 that we define the task of answer selection such that we are given a query question $q \in Q$, a set of n candidate answers $A' = \{a'_0, a'_1, \ldots, a'_n\} \subset A$, and a function $rel : Q, A \rightarrow \{0, 1\}$ indicating whether a candidate answers the question. Our goal is to find a model that produces an optimal ranking of the candidates in A' with respect to rel. Therefore, we learn a score $\sigma_{q,a}$ for each $a \in A'$ with respect to q. For models without attention mechanisms, we usually encode the question and candidate answer separately and independently:

$$\sigma_{q,a} = sim\Big(enc_\Theta(q), enc_\Theta(a)\Big) \tag{3.1}$$

where $\text{enc}_\Theta : \mathbb{V}^* \rightarrow \mathbb{R}^d$ is an encoder ($\mathbb{V}^*$ denotes all sequences over a vocabulary) with network parameters Θ, and $\text{sim} : \mathbb{R}^d, \mathbb{R}^d \rightarrow \mathbb{R}$ is a similarity function, typically cosine similarity.

The choice of similarity function can have an impact on the model's performance scores, e.g., in an isolated setup on one dataset with a CNN encoder the more complicated GESD—the geometric mean of inverse euclidean distance and sigmoid dot product—has been shown to outperform cosine similarity (Feng et al., 2015). However, this does not generalize to other encoders such as attention-based models (Tan et al., 2016). Due to its effectiveness in many scenarios and the mathematical simplicity—cosine similarity is equal to the dot product of two l_2-normalized vectors and can be efficiently computed on a large scale (Johnson et al., 2019)—it is commonly used in recent research (e.g., Tan et al., 2016; Wang et al., 2016a; Lei et al., 2016a).

The procedure outlined above corresponds to a siamese neural network, which uses the same network architecture and parameters for encoding both the question and the candidate answer. Relying on siamese networks has been shown to perform considerably better compared to approaches with separate encoders, or encoders with separate sets of network parameters (Feng et al., 2015). Popular choices for such independent encoders with shared parameters are CNNs and LSTMs. In addition to their good performance scores, they allow us to *pre-compute* the representations of all answers within a corpus. When we want to answer a question, we can re-use our pre-computed answer representations and only need to newly encode the question. This can be much more efficient at runtime than newly encoding all candidate answers with regard to each new question. For example, such independent encoding allows us to perform large-scale semantic similarity search with optimized libraries for dense vector retrieval such as FAISS (Johnson et al., 2019).

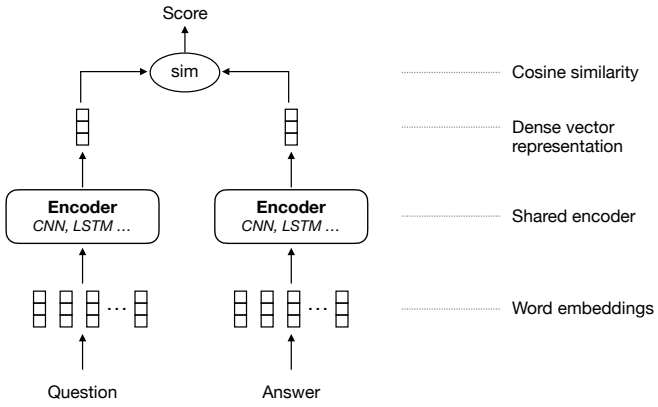

Figure 3.1: Representation learning approaches often encode questions and answers independently with a shared CNN or LSTM encoder. Both encoders, thus, share the same architecture and parameters. The output is a fixed-size dense vector representation that encodes the meaning of a text. To score an answer candidate regarding the question, we compute the similarity between the two independent representations (usually cosine similarity).

A central limitation of CNNs and LSTMs is that they do not explicitly model the importance of segments within the input texts (tokens, phrases, sentences). This is especially important when dealing with long answer texts that contain detailed information. As a solution, attention-based encoders have been proposed that explicitly model importance weightings on a token-level by introducing a dependency between question and answer encoders (Dos Santos et al., 2016; Tan et al., 2016; Wang et al., 2016a). This means that the importance of the tokens within the answer is determined by some notion of similarity to the question (learned as part of the network). However, this procedure results in the loss of a central advantage of approaches for learning dense vector representations: we can no longer pre-compute answer representations independently from the question. Moreover, methods that determine attention with such a dependent encoding might infer misleading attention weights for unrelated question-answer pairs that have a significant lexical overlap.

In this section, we propose to overcome these limitations with *LSTM-based importance weighting*—a self-attentive approach that allows for an independent encoding while at the same time explicitly learning attention weights for question and answer texts. In contrast to other approaches that determine importance weights with dependent encoding, our model uses a separate and independent LSTM component to identify important text segments. On two versions of the non-factoid answer selection benchmark InsuranceQA (Feng et al., 2015), our model performs on-par or better than the best attention-based representation learning model of Dos Santos et al. (2016). In addition, we also show that our model learns suitable representations for factoid answer selection, in which we deal with short answer *sentences*.

3.1.1 Background: Attention Mechanisms

3.1.1.1 Attention Mechanisms in NLP

Attention mechanisms in NLP have originally been proposed for neural machine translation (NMT). NMT models typically follow an encoder-decoder structure (Sutskever et al., 2014; Cho et al., 2014b; Kalchbrenner and Blunsom, 2013). The earliest NMT models first learn a fixed-size dense vector representation of the sentence in a source language (encoder) and then use this representation to generate the translation in a target language (decoder). A downside of this approach is that the encoder needs to compress all the source sentence information in its learned representation, which, similarly to our setup, often fails for long source sentences (Cho et al., 2014a). Attention-based models overcome this limitation by attending to relevant tokens in the input sentence during decoding, thus eliminating the need of including all information in the learned representation of the source sentence (Bahdanau et al., 2015).

Attention mechanisms have led to considerable improvements over encoder-decoder models without attention, e.g., they achieve better performances when translating long sentences (Bahdanau et al., 2015; Luong et al., 2015a). Due to their success in NMT, attention mechanisms have subsequently also been introduced to other NLP tasks such as machine reading comprehension (Hermann et al., 2015), image captioning (Xu et al., 2015), text summarization (Rush et al., 2015), recognizing textual entailment (Rocktäschel et al., 2016), and non-factoid answer selection—which we now describe in more detail.

3.1.1.2 Attention Mechanisms in Non-Factoid Answer Selection

While the motivations for using attention mechanisms in non-factoid answer selection and NMT are similar—i.e., to better handle long texts—both setups considerably differ. In particular, in non-factoid answer selection (including cQA answer selection) we want to learn representations that are more focused on the input text segments that are important in comparing questions and answers. NMT, in contrast, uses attention to attend to parts of the input text during decoding (i.e., translation). Therefore, our setup requires adaptations to the attention mechanisms that have been originally proposed in NMT.

For instance, consider an LSTM encoder (Hochreiter and Schmidhuber, 1997) that processes word embeddings w^a of an answer over time steps t to learn an answer representation r^a:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{w}_t^a, \mathbf{h}_{t-1}) \tag{3.2}$$

$$\mathbf{r}^a = \sum_t \alpha_t \mathbf{h}_t \tag{3.3}$$

With $\alpha_t = \frac{1}{|a|}$, Equation 3.3 corresponds to mean pooling ($|a|$ is the length of the answer). Effective attention mechanisms determine α such that important segments within the question or answer receive more weight. We outline two variants of attention mechanisms in non-factoid answer selection below, one unidirectional and one bidirectional approach.

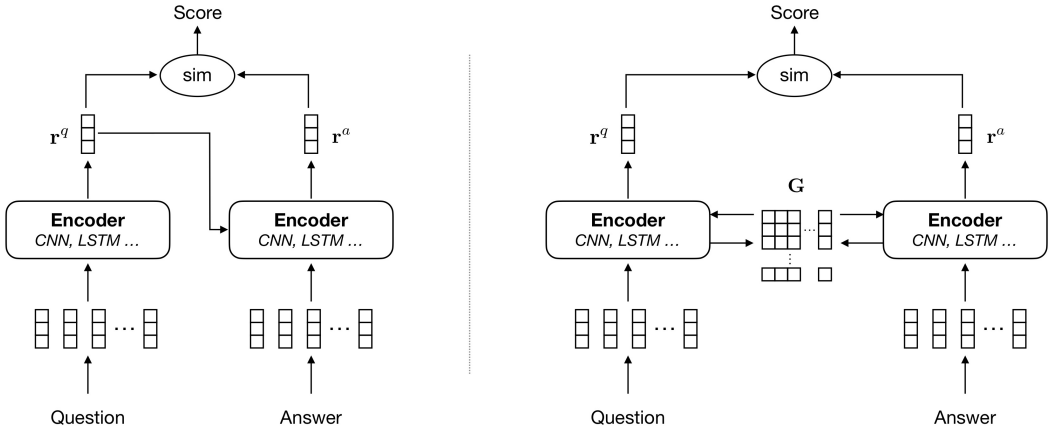

Figure 3.2: An abstract visualization of unidirectional attention (left) and bidirectional attention (right).

Unidirectional attention. Unidirectional attention mechanisms leverage attention for learning only a more focused answer representation. The flow of information is, thus, from question to answer. We show a schematic visualization of this approach in Figure 3.2 on the left. To determine importance in the input, a weight is learned for each token (e.g., word) in the answer by leveraging question information.

For instance, Tan et al. (2016) use an LSTM encoder (see Equation 3.2) to learn the question representation $\mathbf{r}^q$, and then leverage this representation to determine an attention weight α_t for the LSTM hidden state $\mathbf{h}_t^a$ that corresponds to the t^{th} token in the answer:

$$\mathbf{m}_t = \mathbf{w}^\intercal \tanh\left(\mathbf{W}^a \mathbf{h}_t^a + \mathbf{W}^q \mathbf{r}^q\right) \tag{3.4}$$

$$\alpha = \text{softmax}\left(\mathbf{m}\right) \tag{3.5}$$

where $\mathbf{W}^a$, $\mathbf{W}^q$, and $\mathbf{w}$ are network parameters. Therefore, if the t^{th} token of the answer is more important with respect to the question, it likely has a stronger influence on the resulting answer representation.

There exist different variations of unidirectional attention for answer selection. For instance, Wang et al. (2016a) explore applying attention at three different positions within a GRU (gated recurrent unit). Their best model achieves performance scores comparable to the ones reported by Tan et al. (2016).

Bidirectional attention. In contrast to unidirectional attention mechanisms, bidirectional attention mechanisms learn an importance weighting for both questions and answers jointly. Thus, the information flow is not only from question to answer but also from answer to question. A popular approach is "Attentive Pooling" (AP; Dos Santos et al., 2016), which we visualize abstractly in Figure 3.2 on the right. Let $\mathbf{h}^q$ and $\mathbf{h}^a$ be LSTM hidden states for the question and answer, respectively. AP first computes a soft-alignment $\mathbf{G}$ (a matrix expressing similarities) between all tokens in the question and answer:

$$\mathbf{G} = \tanh\left(\mathbf{h}_q \mathbf{U} \mathbf{h}_a^\intercal\right) \tag{3.6}$$

where $\mathbf{U}$ are network parameters that correspond to a learned similarity function. AP now selects the strongest alignment for each word in the question and answer input by computing the column-wise and row-wise maximum over $\mathbf{G}$:

$$\mathbf{m}_t^q = \max_m \mathbf{G}_{t,m} \qquad \mathbf{m}_t^a = \max_l \mathbf{G}_{l,t} \tag{3.7}$$

The attention weights α^q and α^a for the question and answer are then:

$$\alpha^q = \text{softmax}(\mathbf{m}^q) \qquad \alpha^a = \text{softmax}(\mathbf{m}^a) \tag{3.8}$$

AP achieves considerable improvements over approaches without attention in factoid and non-factoid answer selection and it can be used with both LSTMs and CNNs. Further, it performs better than the unidirectional attention mechanism in the non-factoid answer selection benchmark InsuranceQA. There exist several variations of bidirectional attention mechanisms. For instance, ABCNN proposes using similar soft-alignments as additional feature maps to a convolution operation for factoid answer selection (Yin et al., 2016). Yin and Schütze (2017) extend Attentive Pooling with k-min and k-max pooling over the soft alignment matrix, which are task-specific improvements for recognizing textual entailment and factoid answer selection, respectively.

3.1.1.3 Limitations of Previous Approaches

Both unidirectional and bidirectional attention suffer from two conceptual drawbacks: (1) questions and answers cannot be encoded independently, and (2) using question information to determine the importance of answer text segments may, in some cases, result in misleading attention weights.

Dependent encoding. As we have seen previously, common approaches calculate attention by incorporating information from both the question and the answer. Therefore, our independent encoding of Equation 3.1 does not hold anymore, e.g., meaning that we cannot pre-compute answer representations for efficient dense vector search (Johnson et al., 2019). For each new question and n candidate answers from our dataset, we need to compute $n + 1$ representations with unidirectional attention, and $2n$ representations with bidirectional attention. This is in sharp contrast to the constant 1 representation—the representation of the question—that we need to compute with independent encoding and pre-computed answer representations. This conceptual and computational limitation means that previous attention mechanisms can only be used with a small number of candidate answers.

Potentially misleading importance weights. In non-factoid answer selection, we often need to discard many unrelated candidate answers with a high lexical overlap to the question, e.g., when they all address similar topics. Therefore, attention weights that we determine based on a question-answer alignment may not necessarily correspond to segments that constitute the text's most important information. Moreover, segments of an *incorrect* candidate answer that carry important information not matching the question may receive low attention weights when there

is otherwise high lexical overlap. Such misleading importance may, in some cases, result in artificially high scores for incorrect candidate answers. Figure 3.5 includes an example, where the aspect of *getting cash* is central to the answer but not related to the information need of the question (asking for *when* one can borrow something). However, we note that the extent of this phenomenon likely depends on the dataset at hand and may not be globally observed across all possible cQA datasets.

3.1.2 Self-Attention with LSTM-based Importance Weighting

We argue that humans can often identify the topic and the most important aspects of a long description or explanation without having seen any particular question. Moreover, humans can also independently identify which parts of a long answer text are, in general, *not* relevant—e.g., conversational phrases. We thus hypothesize that neural question and answer encoders, likewise, can learn a suitable importance weighting when independently encoding questions and answers.

To test our hypothesis, we propose *LSTM-based Importance Weighting* (LW), a self-attentive approach that learns importance weights with a separate network component that does not require combining question and answer information. We visualize LW in Figure 3.3.

First, we obtain an unpooled representation of the input text, which is the same procedure for most representation learning models (compare Equation 3.2). This can, e.g., correspond to the hidden states $\mathbf{h}$ of a bidirectional LSTM (BiLSTM):[1]

$$\overrightarrow{\mathbf{h}}_t = \overrightarrow{\text{LSTM}}_{\overrightarrow{\Theta}}(\mathbf{w}_t, \overrightarrow{\mathbf{h}}_{t-1}) \tag{3.9}$$

$$\overleftarrow{\mathbf{h}}_t = \overleftarrow{\text{LSTM}}_{\overleftarrow{\Theta}}(\mathbf{w}_t, \overleftarrow{\mathbf{h}}_{t-1}) \tag{3.10}$$

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t] \tag{3.11}$$

where arrows denote the direction in which the individual unidirectional LSTMs process the text, and Θ denotes the network parameters.

We then learn the importance (and only the importance) of each $\mathbf{h}_t$ with an additional and separate BiLSTM:

$$\overrightarrow{k}_t = \overrightarrow{\text{LSTM}}_{\overrightarrow{\Phi}}(\mathbf{h}, \overrightarrow{k}_{t-1}) \tag{3.12}$$

$$\overleftarrow{k_i} = \overleftarrow{\text{LSTM}}_{\overleftarrow{\Phi}}(\mathbf{h}_i, \overleftarrow{k}_{t-1}) \tag{3.13}$$

$$\mathbf{k}_i = [\overrightarrow{k}_i, \overleftarrow{k}_i] \tag{3.14}$$

Our motivation for using LSTM is that they can learn dependencies within the input through their recurrent structure. With this, they can learn which text segments

[1] We can also obtain $\mathbf{h}$ with CNNs. This then corresponds to the outputs after applying the CNN filters over the input text.

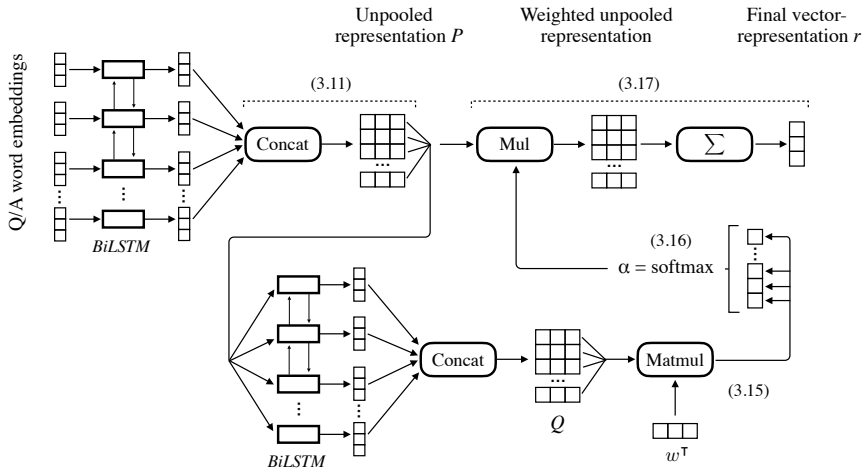

Figure 3.3: The network structure of LW with BiLSTM (LW-BiLSTM). Numbers in parentheses refer to the related Equations.

are most relevant for the task at hand and encode this knowledge in $\mathbf{k}$. We then transform each $\mathbf{k}_i$ to a scalar value ($\mathbf{v}_i$) to obtain attention weights by applying the softmax over $\mathbf{v}$:

$$\mathbf{v}_i = \mathbf{w}^\intercal \mathbf{k}_i \tag{3.15}$$

$$\alpha = \text{softmax}(\mathbf{v}) \tag{3.16}$$

where $\mathbf{w}$ are network parameters.

We finally obtain the fixed-size dense vector representation $\mathbf{r}$ of a question or an answer by weighted pooling over $\mathbf{h}$:

$$\mathbf{r} = \sum_t \alpha_t \mathbf{h}_t \tag{3.17}$$

As we have outlined before, using shared parameters is more effective when learning the unpooled representation of the question and answer texts (which corresponds to $\mathbf{h}$ in our example). We follow this standard procedure and additionally explore using separate network parameters for questions and answers in the LW components (Equation 3.12–3.15), i.e., learning different weighting behavior for questions and answers. This may allow the network to attend to different patterns in the short questions and long answers while still learning compatible representations through the shared encoders that infer the unpooled representation.

It is important to note that conceptually similar approaches have been proposed, partly parallel to our work. They have not used LSTM for importance weighting over long text sequences, and they have applied their methods to different tasks. Parikh et al. (2016) compare premises and hypotheses for natural language inference on a word level with bidirectional attention, and optionally also add a self-attentive component that compares each word in a sentence with all other words of the same sentence. For the same task, Liu et al. (2016) derive the importance based on the comparison of tokens to an average-pooled representation of the same text. Within

	Questions			Candidate Answers per Question		Answer Length
	Train	Valid	Test	All	Correct Answers	Tokens
InsuranceQA v1	12 887	1000	3600	500	1.4	96.5
InsuranceQA v2	12 889	1592	1625	500	1.6	111.8
WikiQA	873	126	243	9.8	1.2	25.2

Table 3.1: Dataset statistics. The number of tokens includes words, numbers, punctuation marks etc.

machine reading comprehension, Li et al. (2016) weight the importance of each token in the question with a simple feedforward network. Lin et al. (2017) extend this by calculating multiple attention values for each word that correspond to different learned attention patterns, which results in multiple representations per sentence. They apply their methods to author profiling, sentiment analysis, and recognizing textual entailment.

3.1.3 Experimental Setup

Training. During training we minimize the max-margin hinge loss:

$$\mathcal{L} = max\left(0,\ m - \text{sim}(\mathbf{r}^q, \mathbf{r}^{a+}) + \text{sim}(\mathbf{r}^q, \mathbf{r}^{a-})\right) \tag{3.18}$$

where $\mathbf{r}^q$ is the learned question representation, $\mathbf{r}^{a+}$ and $\mathbf{r}^{a-}$ are learned representations of correct and incorrect candidate answers, sim is cosine similarity, and m is the desired margin between the similarities.

Datasets often only provide pairs of $(q, a+)$ for training, and thus, we select incorrect candidates $a-$ at training time. For a pair of question and correct answer we randomly sample 50 incorrect candidate answers from the whole training set and select the candidate with the highest similarity to the question according to our currently trained model. This strategy results in more difficult negative examples compared to random sampling, which can lead to better models (Tan et al., 2016).

Datasets. We study the effectiveness of our models on the two non-factoid answer selection datasets InsuranceQA v1 and InsuranceQA v2 (Feng et al., 2015). To test whether our models generalize to other QA settings, we additionally evaluate them on the *factoid* answer selection dataset WikiQA (Yang et al., 2015), which contains short answer sentences. All three datasets require a model to re-rank a set of candidate answers in regard to a given question. We list important dataset statistics in Table 3.1, including the number of train/valid/test questions, the number of candidate answers per question (dev/test), the average number of correct answers, and the average text length of all answers answer.

InsuranceQA v1 and v2 were crawled from the same forum, with two different evaluation setups. v1 includes randomly chosen incorrect candidate answers during evaluation (a model needs to find the correct answer among 500 candidates of which 499 were chosen randomly). In contrast, v2 contains 500 candidate answers that

were retrieved with the Lucene[2] search engine. More details on InsuranceQA are included in Table 2.5 (from our background chapter).

The WikiQA dataset was constructed by means of crowd-sourcing through the extraction of sentences from Wikipedia articles (Yang et al., 2015). We use this dataset to study our models' effectiveness within the different scenario of factoid answer selection that deals with significantly shorter texts. Both, WikiQA and InsuranceQA v2 contain questions without correct candidate answers in the candidate set.[3] We follow the common practice of discarding such examples during evaluation (Dos Santos et al., 2016; Yin et al., 2016).

Evaluation measures. We follow common practice on the datasets and measure the answer selection accuracy on both versions of InsuranceQA. Let $top_k(q)$ be the top-k ranked candidate answers for a given question q, and recall that *rel(q, a)* gives us a binary value indicating whether q is answered by a.

$$precision@k(q) = \frac{1}{k} \sum_{a \in top_k(q)} rel(q, a) \tag{3.19}$$

Accuracy is then equal to precision@1, averaged over all questions—i.e., the ratio of questions for which we found a correct answer at the first position.

On WikiQA, it is common practice to measure the mean reciprocal rank (MRR) and mean average precision(MAP). Let $bestR(q)$ be the best rank of a candidate that answers q.

$$reciprocal\ rank(q) = \frac{1}{bestR(q)} \tag{3.20}$$

Let $allR(q)$ be all ranks of candidates that answer q.

$$average\ precision(q) = \frac{1}{|allR(q)|} \sum_{i \in allR(q)} precision@i(q) \tag{3.21}$$

The mean reciprocal rank (MRR) and mean average precision (MAP) are then the above values, averaged over all questions.

Models and baselines. We combine LW with both BiLSTM (LW-BiLSTM) and CNN (LW-CNN). As baselines we study BiLSTM and CNN with 1-max pooling and the stacked models CNN+BiLSTM and BiLSTM+BiLSTM, which use a BiLSTM with 1-max pooling to process the unpooled representation of the prior component.

The comparison of LW with stacked models is particularly important, since both use the same components within a different network structure. The only added complexity in LW is the projection vector in Equation 3.15. This, however, adds only little model capacity (depending on the representation size, 141–200 additional

[2] https://lucene.apache.org/; last accessed 11.01.2021.

[3] Around 20% of questions in InsuranceQA; Table 3.1 lists the number of questions with correct answers.

parameters). While LW uses the second BiLSTM to learn importance weights, stacked models use the second BiLSTM to adjust the entire representation.

In addition, we compare against the published results of the attention-based models outlined in §3.1.1.2. Because InsuranceQA v2 was created after InsuranceQA v1 and only briefly before we conducted our work, no published results existed on this dataset when we conducted our experiments. Since the code for none of the attention-based models on InsuranceQA v1 was publicly accessible, we re-implemented the best-performing attention model AP-BiLSTM (Dos Santos et al., 2015) for a better comparison.

Neural network setup. We performed grid search over reasonable hyperparameter choices and found the optimal values to be similar to the hyperparameters reported in previous work.

In *InsuranceQA*, the cell size of all LSTMs is 141 (each direction), and the number of filters for all CNNs is 400 with size 3. The only exception is CNN+BiLSTM with 282 filters and a cell size of 282. We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of $4 \cdot 10^{-4}$, a margin of $m = 0.2$, and a batchsize of 20. We initialize the word embeddings with off-the-shelf 100-dimensional uncased GloVe embeddings (Pennington et al., 2014) and optimize them further during training. We apply dropout with a rate of 0.3 on the representations before comparison.

For *WikiQA*, we use 300 dimensional cased GloVe embeddings, an LSTM cell size of 200, 600 CNN filters, a dropout rate of 0.6, and following Dos Santos et al. (2015), we use SGD with a decaying learning rate of $\frac{1.1}{\text{epoch\_index}}$. We do not use different filter sizes for the stacked models CNN+BiLSTM and BiLSTM+BiLSTM. The remaining hyperparameters are the same as in InsuranceQA.

We repeat our experiments three times for InsuranceQA and report the results of the run with the best development score. We apply early stopping with a patience of 10 epochs on InsuranceQA (determined by development accuracy) and for 5 epochs on WikiQA (determined by development MRR).

3.1.4 Experimental Results

We report the experimental results of InsuranceQA in Table 3.2 and the results for WikiQA in Table 3.3.

LW vs. CNN and BiLSTM. We observe that introducing LW to both CNN and BiLSTM substantially improves the answer selection performance on all three datasets. For example, on InsuranceQA v1, LW improves the answer selection performance of CNN and BiLSTM by 9.6% and 4.3%, respectively (test). We observe similar improvements on WikiQA, which shows that our approach generalizes well to factoid answer selection. Neither the reduced length of the answers nor the significantly reduced size of the training data have a noticeable influence on these trends. The largest gain with LW is typically achieved for CNN encoders, likely because CNN only considers local context information to which LW adds global context information. In comparison, the improvements with BiLSTM encoders are smaller,

Model	InsuranceQA v1		InsuranceQA v2	
	Valid	Test	Valid	Test
Attentive BiLSTM (Tan et al., 2016)	68.9	66.9	-	-
IABRNN (Wang et al., 2016a)	69.1	67.0	-	-
AP-BiLSTM (Dos Santos et al., 2016)	68.4	69.1	-	-
AP-BiLSTM (our implementation)	69.8	69.6	32.2	31.9
CNN	60.5	58.3	24.4	24.4
BiLSTM	68.2	65.7	32.4	31.1
CNN+BiLSTM	68.5	67.3	33.0	31.4
BiLSTM+BiLSTM	67.5	66.3	31.2	32.0
LW-CNN shared	69.0	67.8	33.2	34.0
LW-CNN separate	70.0	67.9	33.5	33.7
LW-BiLSTM shared	**70.9**	68.5	**35.4**	36.1
LW-BiLSTM separate	**70.9**	**70.0**	**35.4**	**36.9**

Table 3.2: Experimental results on InsuranceQA v1 and v2. On both datasets we measure the answer selection accuracy (which equals Precision@1).

Model	Dev		Test	
	MAP	MRR	MAP	MRR
AP-CNN (Dos Santos et al., 2016)	-	-	0.6886	0.6957
ABCNN (Yin et al., 2016)	-	-	0.6921	0.7127
ABRNN (Wang et al., 2016a)	-	-	**0.7341**	**0.7418**
CNN	0.6473	0.6531	0.6204	0.6365
BiLSTM	0.6792	0.6828	0.6174	0.6310
CNN+BiLSTM	0.6580	0.6570	0.6560	0.6737
BiLSTM+BiLSTM	0.7037	0.7120	0.6735	0.6789
LW-CNN shared	0.7292	0.7319	0.6992	0.7112
LW-CNN separate	**0.7372**	**0.7463**	**0.7102**	**0.7240**
LW-BiLSTM shared	0.7254	0.7260	0.6854	0.6954
LW-BiLSTM separate	0.7340	0.7434	0.6941	0.7039

Table 3.3: Experimental results on WikiQA.

with overall better absolute performance scores on InsuranceQA compared to CNN encoders.

LW vs. stacked models. Our self-attentive models with LW are more effective compared to the *stacked* CNN+BiLSTM and BiLSTM+BiLSTM models on all datasets. This is important because both LW and stacked models use the same network components. This shows that the improvements of LW are due to its network structure and the weighting mechanism.

Shared vs. separate LW weights. Using separate LW parameters yields improvements in the majority of cases. Although this is intuitive because the question and answer texts are syntactically different, previous work has shown that using separate parameters for learning question and answer representations often leads

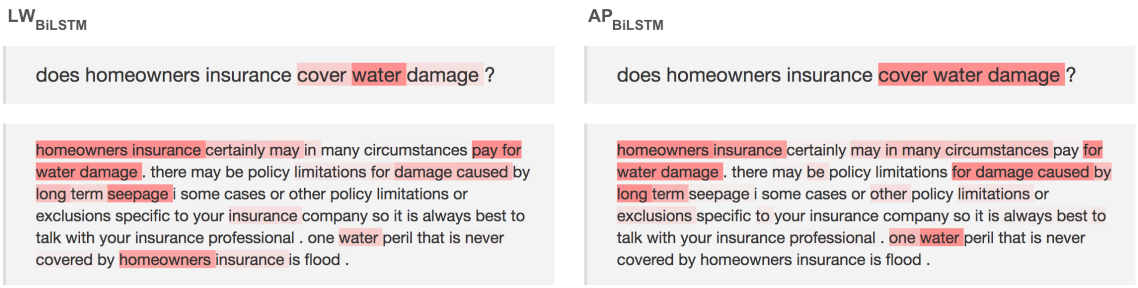

Figure 3.4: A visualization of the attention weights of LW-BiLSTM and AP-BiLSTM for a question and a correct answer (from InsuranceQA). Red colors visualize high relative importance.

to performance degradations (Feng et al., 2015). We suspect that one reason may be that previous encoders mapped questions and answers into two separate (but necessarily compatible) vector spaces, using two differently initialized subnetworks. This can make it difficult to learn suitable representations. In contrast, here we only learn separate *weighting behavior* for questions and answers, and otherwise use shared parameters for learning the unpooled question and answer representations, which mitigates the above mentioned problem—i.e., both texts are mapped into the same vector space.

LW vs. other attention models. LW-BiLSTM achieves better performance scores than other attention models in most of our comparisons, e.g., on InsuranceQA our best model performs better than traditional unidirectional and bidirectional attention models. Notably, AP-BiLSTM only achieves minor improvements compared to BiLSTM on InsuranceQA v2. The most important difference between v1 and v2 is the more realistic selection of candidate answers in v2—all incorrect candidates are lexically similar to the question instead of being chosen randomly. Because AP-BiLSTM relies on an explicitly learned measure of similarity between questions and candidate answers to determine the importance weights, it assigns high scores to lexically similar segments in a question and incorrect candidate answers. LW, in contrast, determines them for questions and answers independently.

In summary, we show that LW learns an effective importance weighting in question and answer texts. At the same time, our approach is not affected by the two crucial shortcomings of attention mechanisms that we have outlined before: (1) our model allows *independent encoding* of questions and answers because importance weights are determined independently for each text; (2) for the same reason, our models do not attend to segments within question and answer texts when they are similar, but rather if these segments define the text's overall topic and meaning.

3.1.5 Analysis

We inspect the results on InsuranceQA v2 to better understand the reasons for the good performance scores of LW, and the limitations of this approach.

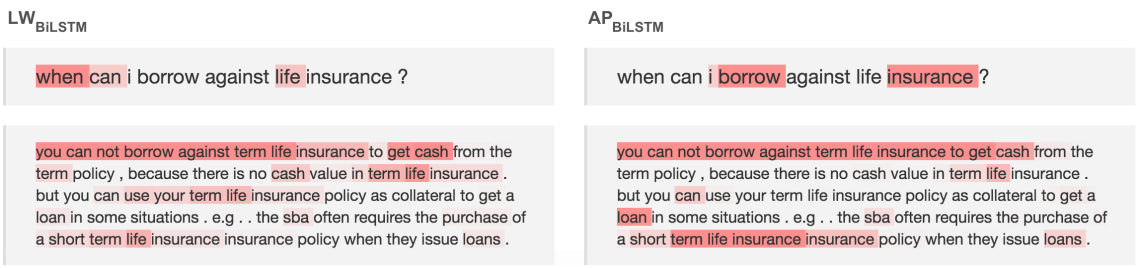

Figure 3.5: A visualization of the attention weights of LW-BiLSTM and AP-BiLSTM for a question and an incorrect candidate answer (with high lexical similarity; both from InsuranceQA). Red colors visualize high relative importance.

3.1.5.1 Attention Weights

We manually probe the attention weights of LW-BiLSTM and compare them to the bidirectional attention of AP-BiLSTM. For pairs of questions and correct candidate answers, we find that both LW-BiLSTM and AP-BiLSTM often learn attention weights that consider similar parts of the texts as most relevant. An example is given in Figure 3.4. We further notice two important attributes that differentiate LW-BiLSTM from AP-BiLSTM, which likely contribute to the differences we observed in our experiments.

(1) For incorrect candidate answers with high lexical similarity to the question, LW-BiLSTM focuses more on segments that are unrelated. This can lead to different representations, which is desired in such cases. In contrast, AP-BiLSTM, by design, focuses strongly on the similar segments and thereby learns similar representations for questions and incorrect candidate answers. We show an example in Figure 3.5. LW focuses strongly on a segment within the question that corresponds to the interrogative adverb *when*. This requires candidate answers to have a similar focus to achieve a high score (e.g., by describing a date).[4] Since this is not the case for the given incorrect candidate, the representations are dissimilar, and the score of this pair is low.

(2) LW-BiLSTM strongly focuses on few highly relevant segments that are well-suited to describe the overall topic of the text. This leads to representations that are less affected by noise because irrelevant segments receive lower relative importance. We quantitatively measure the strength of the importance weights for all answers in InsuranceQA v2. For each question-answer pair (correct or incorrect), we determine the maximum values of the importance weights with LW-BiLSTM and AP-BiLSTM. LW-BiLSTM derives at least one importance weight greater or equal to 0.10 within 77% of all answers, and one importance weight greater or equal 0.20 within 24% of all answers.[5] We find that AP-BiLSTM does not learn such a strong focus—in almost no cases (0%; a very small number) it derives at least one importance weight greater or equal to 0.10.

[4] Our approach sometimes focuses on words indicative of the question type (interrogative pronouns, adverbs, etc.), but this is not always the case. If an important noun is present in the question, LW most often focuses strongly on the noun (e.g. *fire*, *water*, *electricity*).

[5] An importance weight of 0.10 means that the associated token has a high influence on the representation (10%).

3.1.5.2 Limitations of LW

The most common mistake of LW models we observed is that important aspects of the question are not addressed in the selected answer. For instance, the question *"What is a renters insurance declaration page?"* contains the aspects *what* (question type), *renters insurance*, and *declaration page*. When LW-BiLSTM selects a wrong answer, it often differs in only one aspect. For the previous question, our approach selects an answer that describes what the *auto insurance* declaration page is (a similar topic). The reason is the inability of LW to match all important aspects of the question to all important aspects of the answer separately. This can also be observed in our previous example in Figure 3.4, where LW focuses on the aspects *cover* and *water damage* but ignores *homeowners insurance*. Our approach can thus not effectively differentiate between candidate answers that describe *renters insurance* instead of *homeowners insurance*.

We address this shortcoming later in Chapter 5.

3.1.6 Discussion

Should we always encode questions and answers independently? Even though we have argued that this procedure has advantages, we note that there exist various cases in which dependent encoding can be more effective. For instance, when using transformers (Vaswani et al., 2017), it has been shown that performing full attention over a pair of sequences generally performs better than independently encoding both texts and comparing them with cosine similarity or dot product (Humeau et al., 2020; Thakur et al., 2020). However, transformers are vastly different architectures than the ones explored in this chapter: (1) dependent encoding with transformers (termed cross-encoders) learn one joint representation of both texts for scoring, instead of dependently encoding *both* texts and comparing the output representations; (2) cross-encoders effectively combine both approaches explored in this chapter, i.e., self-attentive importance weighting and bidirectional attention; (3) they repeat attentive weighting over a larger number of layers (e.g., 12 or 24) and various so-called attention heads that learn different attention behavior.

The performance gains for transformer-based cross-encoders (mentioned above) show that independent encoding has no inherent modeling advantage over dependent encoding per se, i.e., our observed improvements in §3.1.4 do not necessarily generalize to the most recent neural network architectures. Moreover, transformer models can learn specific attention heads that only rely on information from one input (Kovaleva et al., 2019), which is similar to LW. Finally, we argue that independent encoding still has computational advantages, e.g., it is necessary for dense retrieval.

How does independent encoding relate to dense information retrieval? Dense retrieval is closely related to independent encoding as we have studied it in this chapter. For instance, Gillick et al. (2018) proposed encoding questions by averaging over learned word embeddings for end-to-end *question retrieval*. They showed that dense question representations, in combination with nearest neighbor search, can yield better results than BM25 (Robertson and Zaragoza, 2009). Dense

retrieval may, thus, be a viable alternative to bag-of-words approaches typically used for the first-stage retrieval in cQA (see our system description in §2.2). Most importantly, these approaches are only computationally feasible when pre-computing representations of all documents independently from the query questions. That means if we want to incorporate attention mechanisms, we can only leverage self-attentive approaches such as LW.

Even though there has recently been considerable interest in dense retrieval and independent encoding with transformer-based models (e.g., Yang et al., 2020; Karpukhin et al., 2020; Xiong et al., 2020; Ma et al., 2020; Liang et al., 2020), it has also been shown that these models do not always generalize well to diverse tasks and domains (Thakur et al., 2021). Further work needs to be done to realize the full potential of these approaches and make them more widely applicable.

Finally, we see various opportunities for cQA to benefit from recent advances in dense retrieval. For example, USE-QA (Yang et al., 2020), which has been shown to perform well for factoid answer retrieval (Guo et al., 2020), encodes questions and answers within different networks. Answer sentences are encoded within a "context" that consists of the surrounding text from the passage in which the sentence appears. In cQA, we could apply this idea to encode the context of an answer, which may be represented by the discussion thread in the forum. Besides, Ding et al. (2020) propose several ideas to obtain more difficult negative instances for model training, yielding considerable gains in retrieval tasks. We may adapt their techniques to also train better cQA models.

Can we learn attention at different granularities? An interesting extension of LW would be to *incorporate* sentence-level attention. Under the assumption that long descriptions, explanations, and advice—similar to news articles—follow an inverted pyramid scheme (Pöttker, 2003), the first answer sentences may often be the most important ones. Sentence-level attention could automatically learn such patterns, while also learning to ignore conversational sentences (e.g., "that is a great question!"; see Figure 3.8). However, we believe that token-level attention may still be necessary, e.g., for effectively processing long sentences. Combining both token and sentence-level attention may be beneficial during training, e.g., for more effectively learning token-level attention by applying it to only a few important sentences instead of the whole answer text.

3.1.7 Summary

In this section, we described common attention mechanisms for non-factoid answer selection, and outlined two important shortcomings of previous work. (1) The dependent encoding of questions and answers yields models that require us to compute new answer representations for each question. (2) Using question information to determine the importance in answers may, in some cases, result in misleading attention when questions and incorrect answers have a high lexical overlap.

We argued that identifying which segments within an answer are important does not necessarily require information from the question, i.e., we can determine suit-

able importance weights without requiring the dependent encoding of questions and answers. To validate our hypothesis, we proposed LSTM-based importance weighting (LW), a self-attentive approach with a separate BiLSTM component for learning the importance of segments in question and answer texts. Our experimental results on two variants of InsuranceQA and on the WikiQA dataset demonstrate the effectiveness of LW, which considerably outperforms several baselines such as CNN, BiLSTM models as well as stacked variants thereof. LW achieves on-par or better results compared to other attention-based models with dependent encoding and outperforms Dos Santos et al. (2016)'s bidirectional attention model on InsuranceQA v2, in which incorrect answers have a high lexical overlap to the question.

Most importantly, we have established that self-attention is effective in non-factoid answer selection. In contrast to previous models with dependent encoding, self-attentive models allow us to pre-compute all answer representations within a large corpus independently from the question, which may yield improvements in the computational efficiency at run-time and can enable large-scale dense vector retrieval.

We have shown that our approach can also be applied beyond non-factoid answer selection to the short factoid answers in WikiQA. Other work has confirmed that LW generalizes well to even more distant tasks such as identifying optimal food and wine pairings based on wine reviews and other textual data (Hu, 2018).

3.2 An Interactive Non-Factoid QA System for Visualizing Neural Attention

> Parts of this section have been previously published as listed below. Verbatim quotes from this publication are included in this section.
>
> **Andreas Rücklé** and Iryna Gurevych: 'End-to-End Non-Factoid Question Answering with an Interactive Visualization of Neural Attention Weights', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (ACL 2017; System Demonstrations), pp. 19–24, July 2017.
>
> My contributions: System architecture, the overall concept, implementation of the backend, implementation of the frontend, neural network models, training procedures, data processing pipelines, framework design.

As we have seen in Section 3.1, advanced attention mechanisms are an essential part of successful approaches to non-factoid answer selection. They allow the models to focus on a few important segments within long answer texts and learn better representations. Analyzing attention mechanisms is thus crucial for understanding the models' strengths and weaknesses, which requires researchers to compare different approaches. Due to the lack of tool-support to aid this process, such analyses are complex and require substantial development effort. To close this gap, we created an integrated solution for helping researchers better understand different attention-based models' capabilities and aid qualitative analyses.

We thereby complement our previous section and present an extensible and mod-

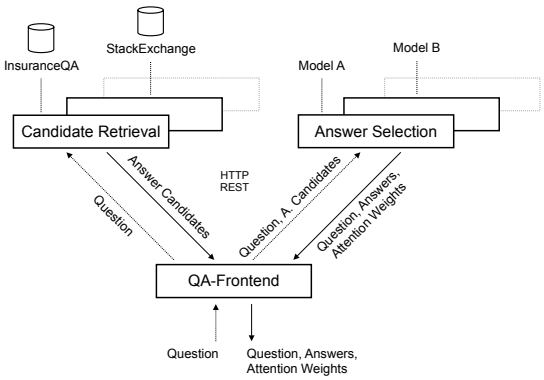

Figure 3.6: A high-level view on our service architecture. The three services are loosely coupled, communicate over HTTP REST APIs, and are easily extensible. The loose coupling also means that datasets can be easily exchanged and models can be used in parallel within the same frontend (allowing for a side-by-side comparison of attention weights).

ular service architecture that transforms models to non-factoid answer selection into fully-featured end-to-end question answering systems. Our system enables researchers to explore and compare attention-based models for answer selection interactively, by visualizing the associated attention weights dynamically for any given question. Researchers can explore different models simultaneously and compare two attention mechanisms side-by-side within the same view. Indeed, for our analyses in §3.1.5 we relied on an early prototype of the full system presented here.

3.2.1 System Overview

We propose a service orchestration that incorporates multiple web services with separate responsibilities. All services communicate using well-defined HTTP REST APIs, thus achieving strong extensibility properties. This simplifies researchers' efforts to replace our services with their own implementations, e.g., to add models implemented with different machine learning frameworks.

Figure 3.6 shows a high-level view of our system architecture. For each question, we retrieve a list of candidate answers from a given dataset (candidate retrieval). We then rank these candidates with the answer selection component, which integrates our attention-based model. The result contains the top-ranked answers in regard to the question and all associated attention weights, which are then visualized by our user interface.

Our architecture is similar to the pipelined structures of earlier work in question answering that rely on a retrieval step followed by a more expensive supervised ranking approach (Hoque et al., 2016; Surdeanu et al., 2011; Higashinaka and Isozaki, 2008). In particular, we simplify our system structure that we introduced in Section 2.2 by removing the question similarity component. This allows researchers to relate the results directly to the answer selection models that are being analyzed. Including a separate question similarity step, or using more advanced components such as query expansion or answer merging would introduce additional complexity, making

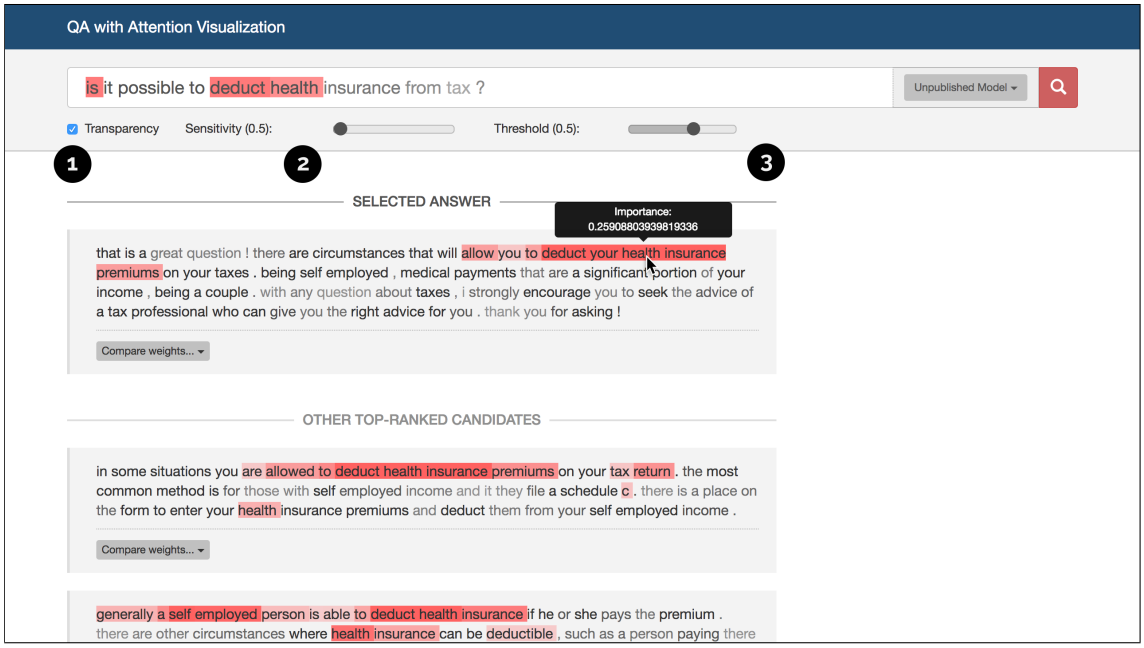

Figure 3.7: The user interface of our question answering system with an interactive visualization of neural attention weights. The UI includes several options to adapt the attention visualization to the researchers' needs. We can (1) optionally render text segments transparent (grayed out) if they receive low attention weights; (2) adapt the sensitivity for computing the opacity of attention highlights; (3) change the threshold that is used to determine when to show attention highlights. The texts are from InsuranceQA.

it difficult to analyze the results of the answer selection model individually.

Because all components in our extensible service architecture are loosely coupled, it is possible to integrate multiple answer selection services with different attention mechanisms at the same time. The user interface leverages this and allows researchers to seamlessly switch between different models. Furthermore, we include a method to interactively compare two models side-by-side within the same view. This allows researchers to directly align and qualitatively compare two attention mechanisms for the same input example, thus potentially reducing the effort required for error analysis. We show a screenshot of our question answering interface in Figure 3.7, and an example of a side-by-side comparison in Figure 3.8.

In the following sections, we describe the individual services and briefly discuss their technical properties.

3.2.2 Candidate Retrieval

The efficient retrieval of answer candidates is an important component of our system, allowing us to pre-select a small number of answers from the whole dataset that are lexically similar to the question. We index all existing answers of a dataset with ElasticSearch[6], an open-source high-performance search engine. Our service

[6] https://elastic.io; last accessed 27 Jan. 2021.

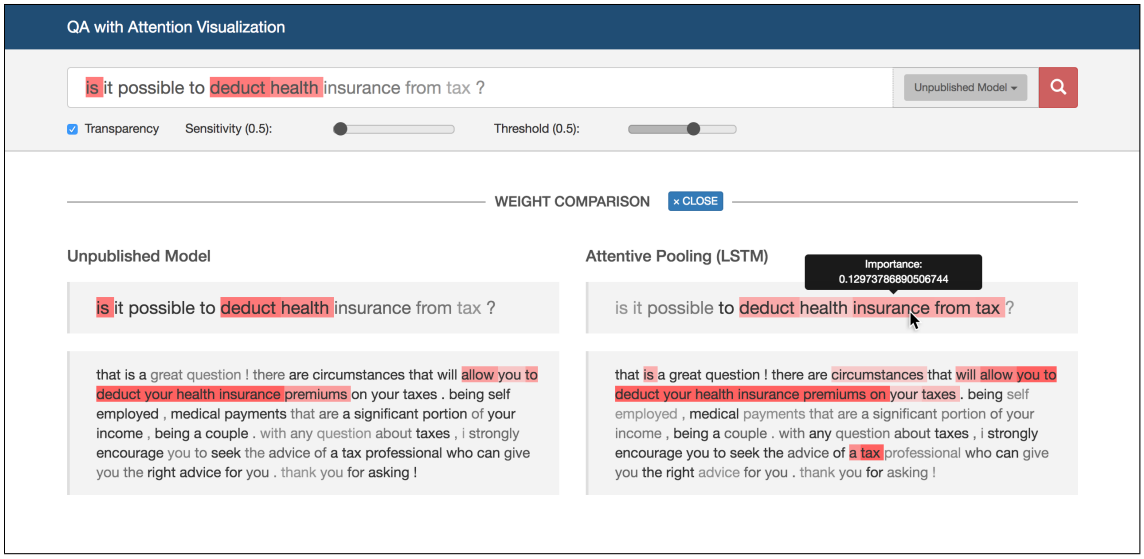

Figure 3.8: A side-by-side comparison of two different attention-based models. Our user interface allows researchers to spot the used models' differences quickly, and to better analyze their behavior. For instance, the left model focuses more strongly on a few individual words in the question than the model to the right. Our user interface makes it easy to analyze such differences. The texts are from InsuranceQA.

provides a unified interface for the retrieval of answer candidates that we query with BM25 (Robertson and Zaragoza, 2009).

The service implementation is based on *Scala* and the *Play Framework*.[7] We provide a data reader to index InsuranceQA (Feng et al., 2015) and a data reader that can index all publicly available dumps of StackExchange.[8] In addition, researchers can add new datasets by implementing a single data reader class.

3.2.3 Answer Selection

The answer selection service provides an interface to the attention-based models, which rank a list of candidate answers according to a question. It extracts attention weights from the model, which we include in the result of the service call. It uses *Flask*[9] for the service implementation.

Due to the research focus of our system, we include a fully configurable and modular *framework* for experimentation that includes different modules for training and evaluating answer selection models. This framework is based on TensorFlow (Abadi et al., 2016) and leverages our experimental source code that we created as part of our research in Section 3.1, and thus, includes tested models both with and without attention. Furthermore, our framework includes different training procedures, including random and difficult negative sampling (see §3.1.3).

Our framework is:

[7] https://www.playframework.com/; last accessed 3 Dec. 2020.
[8] https://archive.org/details/stackexchange; last accessed 15 Jan. 2021.
[9] https://flask.palletsprojects.com; last accessed 27 Jan. 2021.

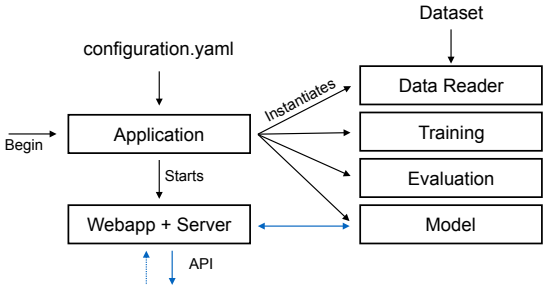

Figure 3.9: Our experimental framework is fully configurable trough YAML configuration files. The data reader, training procedure, model, and evaluation procedure, are dynamically instantiated with the hyperparameters and options specified in the experiment configuration.

- **Fully configurable:** We use *YAML* files to define the whole experiment setup, including all hyperparameters of the data reader, training module, evaluation procedure, and model. We provide an excerpt of a configuration in Listing 3.1.
- **Dynamically composable:** New data readers, training methods, evaluation procedures, and, most importantly, models can be seamlessly integrated. Our framework requires no complicated setup; the model classes are dynamically instantiated according to the given configuration file. Due to our well-defined APIs and class structure, all components are (by design) compatible.
- **Extensible**: Extending our framework with new data sources, training and evaluation procedures, and models is as simple as implementing a single class.

We show a high-level view of the framework structure in Figure 3.9.

3.2.4 QA-Frontend and User Interface

The QA-Frontend coordinates all other services and integrates them into a question answering system. A particular focus lies in the user interface to explore and compare attention-based models. It fulfills the following requirements:

- We visualize attention with heatmaps similar to (Hermann et al., 2015).
- Our UI supports unidirectional and bidirectional attention mechanisms as well as self-attention.
- Researchers can query different models within the same view.
- We provide researchers with a side-by-side comparison of different attention-based models.

Figure 3.7 shows a screenshot of the query view. In the top row, researchers enter their questions and select the model to be used for answer selection. Below the input, we offer multiple ways of adapting the attention visualization to the researchers' needs. For instance, we can highlight only the most important text segments by increasing the threshold t. Furthermore, we can change the sensitivity s of the highlights' opacity with regard to the attention strength.

```
1   data-module: data.insuranceqa.v2
2   model-module: model.lw_lstm
3   training-module: training.dynamic
4   evaluation-module: evaluation.default
5
6   data:
7     map_oov: true
8     embeddings: data/glove.6B.100d.txt
9     insuranceqa: data/insuranceQA
10    ...
11
12  model:
13    lstm_cell_size: 141
14    margin: 0.2
15    trainable_embeddings: true
16    ...
17
18  training:
19    negative_answers: 50
20    batchsize: 20
21    epochs: 100
22    save_folder: checkpoints/lw_lstm
23    dropout: 0.3
24    optimizer: adam
25    scorer: accuracy
26    ...
```

Listing 3.1: An excerpt of a *YAML* configuration file for the candidate ranking framework. Lines 1–4 show how the dataset, model, training procedure, and evaluation are referenced. Models can either be trained with our framework or served for prediction. The individual configuration sections define hyperparameters and other options for the individual components.

We calculate the opacity of an attention highlight o_i that corresponds to the weight w_i in position i as follows:

$$a = min\left(w_{std},\ w_{max} - w_{avg}\right) \tag{3.22}$$

$$o_i = \begin{cases} s \cdot \frac{w_i - w_{avg}}{a}, & if \quad w_i \geq w_{avg} + a \cdot t \\ 0, & otherwise \end{cases} \tag{3.23}$$

where w_{avg}, w_{std} and w_{max} are the average, standard deviation, and maximum of all attention weights. We use a instead of w_{std} because in rare cases $w_{std} > w_{max} - w_{avg}$, which would lead to visualizations without fully opaque positions.

When the user hovers over an answer and the selected model employs a two-way attention mechanism, the *question* view visualizes the associated attention weights.

Finally, we can use each answer as a basis for comparing the attention weights to another model in our side-by-side view (Figure 3.8).

We implemented the user interface with modern web technologies, such as *Angular*, *TypeScript*, and *SASS*. The QA-Frontend service is implemented in *Python* with *Flask*. It is fully configurable and allows multiple answer selection services to be used at the same time.

3.2.5 Impact

We were able to successfully adapt this system for our SoftwareCampus project "Intelligent Search in the Social Web" (2018–2020, in collaboration with DATEV eG). This project implements the full cQA pipeline introduced in Section 2.2. Furthermore, our experimental framework has served as the technical basis for several other research projects, e.g., our work in Chapter 5 and Chapter 7. It has also been adapted for research beyond QA, e.g., the story cloze task (Bugert et al., 2017) and natural language generation (Puzikov and Gurevych, 2018). Although we are not aware of published research *systems* that were *built upon* our source code, we do note that our project has received some attention on GitHub. With 18 forks and two external code contributors (as of 12.01.2021), there is some evidence that our system has been re-used, possibly for practical or analysis purposes. Finally, our publication has been frequently mentioned as prior work to related research systems (e.g., Strobelt et al., 2018; Liu et al., 2018; Cho et al., 2019; Lee et al., 2019a).

3.2.6 Conclusion

We presented an extensible service architecture, allowing researchers to analyze their attention-based models qualitatively by interactive exploration. All our system components are modular, giving researchers the possibility to add additional functionality easily. For example, our answer retrieval component can accommodate new datasets, and our experimental framework can be complemented with new models, evaluation procedures, training methods, or training sets without requiring changes in other parts of the application.

As with all our work presented through Chapters 3–7, our system and framework is publicly available and open-source (Apache 2.0 license):
`https://github.com/UKPLab/acl2017-non-factoid-qa`.

3.3 Chapter Summary

In this chapter, we introduced LSTM-based importance weighting (LW), a self-attentive approach to non-factoid answer selection. We demonstrated on two versions of InsuranceQA that our approach is effective. It considerably outperforms stacked models, which use the same neural network components, but compose them differently. We have shown that LW performs on-par or better than many cross-attention mechanisms that introduce a dependency between the question and answer encoders to determine the importance of text segments. Furthermore, we have found

that our approach is not specific to non-factoid answer selection; it also generalizes well to the short factoid answers of WikiQA.

Our self-attentive representation learning approach LW has two critical advantages in our setup. (1) It independently encodes questions and answers, which allows pre-computing all answer representations. Therefore, we need to encode only one representation—the question representation—for answer selection during run-time. (2) Self-attentive approaches do not determine the importance of answer text segments based on question information, thereby avoiding potentially misleading importance weights in some settings.

To thoroughly understand the advantages of different attention mechanisms, we need to explore and compare them interactively. Therefore, we have presented an end-to-end question answering system and a service architecture that enables analyzing attention-based models in depth. Its components are fully configurable, extensible, and modular, allowing researchers to add new functionality easily.

We conclude this chapter by noting that with the recent revolution caused by large pre-trained transformer models (Vaswani et al., 2017) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), self-attention has become ubiquitous in modern NLP. Current approaches to representation learning for sentences (Reimers and Gurevych, 2019) and question-answer pairs (Yang et al., 2020) all leverage multi-head self-attention, proving that related techniques are indeed very effective.

Chapter 4

Cross-Lingual Transfer of Representation Learning Models

In the previous chapter, we studied attention-based models for learning dense vector representations of questions and answers. Besides focusing entirely on the task of answer selection, we trained and evaluated our models with English data only. This has been (and still is) common practice across a wide range of NLP tasks due to a broad availability of English corpora and datasets of sufficient sizes, both labeled or unlabeled, in the world's most commonly spoken language.

Training and evaluating models for one task and only for the English language, however, has limitations. Most importantly, the resulting models are not universal in that they do not transfer well to other tasks and languages, limiting their scope. In this chapter, we tackle this challenge and address RQ2 of this thesis:

> **RQ2: How can we obtain approaches to representation learning that transfer well to different languages?**

More accurately, we address RQ2 from two angles and answer the following questions in this chapter:

1. How can we obtain text representations that are universal across classification tasks such as question type classification and sentiment analysis? How can we transfer these representations across different languages?
2. How can we transfer monolingual question similarity models cross-lingually? Which challenges arise in an expert domain that uses a specialized vocabulary?

In Section 4.1, we investigate universal cross-lingual *sentence embeddings* and propose an efficient training-free technique that combines different *power means* over pre-trained word embeddings. We show that our approach generalizes well across languages with a smaller cross-lingual performance decreases compared to more complex techniques.

Section 4.2 then investigates the impact of neural machine translation on the cross-lingual transfer of monolingual question similarity models. We find that in technical domains with programming questions, translation errors' negative impact has a

subtle effect on task performances. We show that data augmentation with back-translation yields better models that achieve considerable improvements.

4.1 Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations

Parts of this section have been previously published as listed below. Verbatim quotes from this publication are included in this section.

Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych: 'Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations', published on the *arXiv pre-print* server (arXiv 2018; arXiv:1803.01400v2), March 2018.

My contributions: Monolingual- and cross-lingual implementation of concatenated power mean word embeddings. Conducting the monolingual and cross-lingual experiments. Extension of the experimental frameworks to the cross-lingual case. Cross-lingual (re-)mapping of word embeddings. Analyzing different power means and the cross-lingual performance gap. Establishing the cross-lingual adaptations of InferSent.

Sentence embeddings are dense vectors that encode different properties of a sentence, e.g., its meaning and its sentiment. They thereby extend the concept of word embeddings (Mikolov et al., 2013a; Pennington et al., 2014) to the sentence-level.

Universal sentence embeddings have gained considerable attention over the past years due to their wide range of applications (Conneau et al., 2017). They differ in several aspects from the task-specific representations that we learned in Chapter 3:

- Universal sentence embeddings are task-agnostic.
- They encode individual sentences instead of text passages or documents.
- They can efficiently be transferred to small data scenarios, e.g., when there is little training data for a downstream task (Subramanian et al., 2018a).

Hence, universal sentence embeddings are also broadly applicable to cQA. For instance, we could use them to determine the type of a question, the sentiment of an answer, or whether an answer contains argumentative sentences.

To a certain degree, the history of sentence embeddings parallels that of word embeddings, but on a faster scale. Early word embeddings were complex and often required months of training (Bengio et al., 2003; Collobert and Weston, 2008; Turian et al., 2010) before Mikolov et al. (2013a) introduced a much simpler method. This made it possible to leverage more data, leading to better results. Sentence embeddings, likewise, originated from the rather resource-intensive Skip-thought encoder-decoder model of (Kiros et al., 2015), before more efficient models (Hill et al., 2016; Kenter et al., 2016; Arora et al., 2017) were proposed. One of the most popular sentence embeddings models is InferSent (Conneau et al., 2017), which learns sentence embeddings with a neural network architecture in a single day (on a GPU from around that time), but on high quality Natural Language Inference data (Bowman et al., 2015). Following previous work (e.g., Kiros et al., 2015), InferSent has also set the

standards in measuring the usefulness of sentence embeddings by requiring the embeddings to be *universal* in the sense that they should perform well across a wide variety of transfer tasks.

We follow both of these trends and posit that sentence embeddings should be *simple*, on the one hand, *and universal*, on the other hand. Moreover, we extend the property of universality to the cross-lingual case. Universal sentence embeddings should perform well across multiple tasks and **across natural languages**. These properties ensure the wide reusability of sentence embeddings in both the computational (low resource consumption) and conceptual sense (widely applicable).

One of the simplest sentence embedding techniques is to average individual word embeddings. This is the starting point of our experiments. We observe that average word embeddings have suffered from disadvantages in previous work such as (Conneau et al., 2017) because the newly proposed methods yield sentence embeddings of large sizes (e.g., with a dimensionality of $d = 4096$) while they have been compared to much smaller average word embeddings (e.g., $d = 300$). Increasing the size of individual—and therefore also average—word embeddings likely improves the quality of average word embeddings, but with an inherent limitation: there is practically only a finite number of words, so that the additional dimensions beyond a certain threshold will not be used to encode new information.

To remedy this, we propose two extensions that leverage simple and, thus, computationally efficient operations:

1. We concatenate diverse word embeddings that encode *different kinds of information*, such as syntactic, semantic, or sentiment information. Concatenation of word embeddings is a simple and effective technique that yields improvements in different setups (Zhang et al., 2016a).

2. We argue that *mean* or average has been defined too narrowly in previous work. Standard average word embeddings stack the word embeddings of a sentence and compute per-dimension *arithmetic* means. Here, we instead focus on the *power mean* (Hardy et al., 1952), which naturally generalizes the arithmetic mean and thereby captures more information from the sequence of word embeddings.

Finally, we combine the concatenation of word embeddings with different power means and show that our sentence embeddings satisfy our extended notion of universality. They outperform many popular sentence embedding models across a number of classification tasks monolingually and substantially outperform other approaches cross-lingually.

4.1.1 Background: Word and Sentence Embeddings

We provide a brief overview over common monolingual and cross-lingual sentence embeddings that have been introduced prior to our work. We also include an overview over popular word embedding techniques to illustrate their broad variety.

Monolingual word embeddings are typically learned to predict context words within fixed windows (Mikolov et al., 2013a; Pennington et al., 2014). Extensions predict contexts given by dependency trees (Levy and Goldberg, 2014) or combinations of windows and dependency context (Komninos and Manandhar, 2016), leading to more syntactically-oriented word embeddings. FastText (Bojanowski et al., 2017) represents words as the sum of their n-gram representations trained with a skip-gram model. Retro-fitting approaches extend this by introducing constraints from lexical resources to fine-tune word embeddings with linguistic information (Faruqui et al., 2015; Nguyen et al., 2016a; Rothe and Schütze, 2015; Vulić et al., 2018; Glavaš and Vulić, 2018). For instance, Attract-repel (Mrkšić et al., 2017) use synonymy and antonymy constraints and (Vulić et al., 2017) leverage language-specific rules so that derivational antonyms ("expensive" vs. "inexpensive") move further away in a vector space.

Cross-lingual word embeddings originate from the idea that monolingually and cross-lingually similar words should be close in a vector space. A common practice is to learn a mapping between two monolingual word embedding spaces (Faruqui and Dyer, 2014; Artetxe et al., 2016). Other approaches predict monolingual and cross-lingual context using word alignment information as an extension to the standard skip-gram model (Luong et al., 2015b) or inject cross-lingual synonymy and antonymy constraints similar to the monolingual setting (Mrkšić et al., 2017). As with monolingual embeddings, there are many different approaches, but they have been reported to perform similarly in applications (Upadhyay et al., 2016).

In this section, we experiment with embeddings that were trained with one of the simplest cross-lingual approaches: BIVCD (Vulić and Moens, 2015). This approach creates bilingual word embeddings from aligned bilingual documents by concatenating parallel document pairs and shuffling the words in them before running a standard word embedding technique.

Monolingual sentence embeddings usually build on top of existing word embeddings, and different approaches focus on computing sentence embeddings by composition of word embeddings. Wieting et al. (2015) learn paraphrastic sentence embeddings by fine-tuning skip-gram word vectors while using additive composition to obtain representations for short phrases. SIF (Arora et al., 2017) computes sentence embeddings by taking weighted averages of word embeddings and then modifying them via SVD. Sent2vec (Pagliardini et al., 2018) learns n-gram embeddings and averages them. Siamese-CBOW (Kenter et al., 2016) trains word embeddings that, when averaged, should yield good representations of sentences. However, it has been shown that even non-optimized average word embeddings encode valuable information about the sentence, e.g., its length or which words appear in it (Adi et al., 2017).

Other approaches consider sentences as additional tokens whose embeddings are learned jointly with words (Le and Mikolov, 2014), use auto-encoders (Hill et al., 2016), or mimick the skip-gram model (Mikolov et al., 2013a) by predicting surrounding sentences (Kiros et al., 2015).

One of the most popular approaches is InferSent (Conneau et al., 2017), which achieved state-of-the-art results across a wide range of different transfer tasks. This model uses bidirectional LSTMs and is trained on the SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) corpora. This contrasts previous work, which, likewise, used LSTMs to learn sentence embeddings but trained models on other tasks (e.g., identifying paraphrase pairs) and usually did not achieve significant improvements compared to simple word averaging models (Wieting et al., 2016).

Cross-lingual sentence embeddings have received comparatively less attention. Hermann and Blunsom (2014) learn cross-lingual word embeddings and infer document-level representations with simple composition of unigrams or bigrams, finding that summed word embeddings perform on par with the more complex bigram model. Several authors proposed to extend ParagraphVec (Le and Mikolov, 2014) to the cross-lingual case: (Pham et al., 2015) add a bilingual constraint to learn cross-lingual representations using aligned sentences; Mogadala and Rettinger (2016) add a general cross-lingual regularization term to ParagraphVec; Zhou et al. (2016b) train task-specific representations for sentiment analysis based on ParagraphVec by minimizing the distance between paragraph embeddings of translations. Finally, Chandar et al. (2013) train a cross-lingual auto-encoder to learn representations that allow reconstructing sentences and documents in different languages, and Schwenk and Douze (2017) use representations learned by an NMT model for translation retrieval.

To our best knowledge, all of these prior cross-lingual works evaluate on few individual datasets, and none focuses on *universal* cross-lingual sentence embeddings that perform well across a wide range of different tasks.

4.1.2 Concatenated Power Mean Word Embeddings

4.1.2.1 The Power Mean

Average word embeddings summarize a sequence of embeddings $\mathbf{w}_1, ..., \mathbf{w}_n \in \mathbb{R}^d$ with the component-wise arithmetic mean:

$$\forall i = 1, \ldots, d: \quad \frac{w_{1i} + \ldots + w_{ni}}{n} \tag{4.1}$$

This is arguably one of the simplest sentence embedding techniques, both conceptually and computationally. The arithmetic mean, however, only captures a small amount of information about the word sequence $(w_{1i}, \ldots, w_{ni})$. In order to capture more information, we might also compute other statistics such as the standard deviation, the skewness, etc. For simplicity and to focus on only one type of extension, we consider the *power mean* (Hardy et al., 1952), defined as:

$$\left(\frac{x_1^p + \ldots + x_n^p}{n}\right)^{1/p}; \quad p \in \mathbb{R} \cup \{\pm\infty\} \tag{4.2}$$

where $(x_1, \ldots, x_n)$ is a sequence of numbers. The power mean generalizes many well-known means such as the arithmetic mean ($p = 1$), the geometric mean ($p = 0$),

and the harmonic mean ($p = -1$). In the extreme cases the power mean specializes to the minimum ($\lim_{p\to-\infty}$) and maximum ($\lim_{p\to\infty}$) of the sequence.

We refer to such different instantiations of the power mean (with given p-values) as *a power mean* or *power means*.

4.1.2.2 Concatenation

For vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$, concisely written as the matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_n] \in \mathbb{R}^{n\times d}$, we let $H_p(\mathbf{W})$ stand for the vector in $\mathbb{R}^d$ whose d components are the results of a power mean applied to the sequences $(w_{1i}, \ldots, w_{ni})$, for all $i = 1, \ldots, d$.

Given a sentence $s = w_1, \ldots, w_n$ we first retrieve the embeddings $\mathbf{W}^{(i)} = [\mathbf{w}_1^{(i)}, \ldots, \mathbf{w}_n^{(i)}] \in \mathbb{R}^{n\times d_i}$ of its words from some word embedding space $\mathbb{E}^i$. We then apply K different power means on s and then concatenate the results:

$$\mathbf{s}^{(i)} = H_{p_1}(\mathbf{W}^{(i)}) \oplus \ldots \oplus H_{p_K}(\mathbf{W}^{(i)}) \tag{4.3}$$

where $\oplus$ stands for the concatenation operation and $p_1, \ldots, p_K$ are the K different p-values of the power means. Our resulting sentence representation, denoted as $\mathbf{s}^{(i)} = \mathbf{s}^{(i)}(p_1, \ldots, p_k)$, lies in $\mathbb{R}^{d_i \cdot K}$.

To leverage further representational power from multiple word embedding spaces, we concatenate different $\mathbf{s}^{(i)}(p_1, \ldots, p_k)$ obtained from different embedding spaces $\mathbb{E}^i$:

$$\bigoplus_i \mathbf{s}^{(i)} \tag{4.4}$$

The dimensionality of this representation is $K \sum_i d_i$. When all embedding spaces have the same dimensionality d—which is commonly the case—this becomes $K \cdot L \cdot d$, where L is the number of spaces considered.

In summary, the concatenation of different power means and word embedding spaces *captures more information* about the sequence compared to average word embeddings, and at the same time *includes richer information*. Our approach—which we refer to as **concatenated power mean (word) embeddings**—is computationally efficient, training-free, and seamlessly extensible with new power means or embedding spaces. This allows us to expand our approach to the cross-lingual case by incorporating cross-lingual word embeddings (see §4.1.4).

4.1.3 Monolingual Experiments

We first study the effectiveness of our sentence embeddings in monolingual scenarios in comparison to average word embeddings (a common baseline) and the previous state of the art. In §4.1.4, we then investigate the cross-lingual transfer.

4.1.3.1 Monolingual Setup

Tasks. We replicate the setup of Conneau et al. (2017) and evaluate on their six transfer tasks for sentence classification. Since their selection of tasks is slightly

Task	Type	Size	C	Translation, Example
AM	Argumentation	7k	4	HT(German), MT(French). Example (label="claim"): "*Viele der technologischen Fortschritte helfen der Umwelt sehr*"
AC	Argumentation	450	2	HT(German), MT(French). Example (label="none"): "*Too many promises have not been kept*"
CLS	Product reviews	6k	2	HT. Example (label="pos"): "*En tout cas on ne s'ennuie pas à la lecture de cet ouvrage!*"
MR	Sentiment	11k	2	MT. Example (label="pos"): "*Dunkel und verstörend, aber auch überraschend witzig*"
CR	Product reviews	4k	2	MT. Example (label="neg"): "*This camera has a major design flaw*"
SUBJ	Subjectivity	10k	2	MT. Example (label="obj"): "*On leur raconte l'histoire de la chambre des secrets*"
MPQA	Opinion polarity	11k	2	MT. Example (label="neg"): "*nicht zu unterstützen*"
TREC	Question types	6k	6	MT. Example (label="desc"): "*What's the Olympic Motto?*"
SST	Sentiment	70k	2	MT. Example (label="pos"): "*Holm... incarne le personnage avec un charisme regal sans effort*"

Table 4.1: Evaluation tasks with examples from our transfer languages. The first three tasks include human-generated cross-lingual data (HT), the last 6 tasks contain machine translated sentences (MT). C is the number of classes.

biased towards sentiment analysis, we add three additional tasks. (1) AM, an argumentation mining task based on (Stab and Gurevych, 2017) where sentences are classified into the categories major claim, claim, premise, and non-argumentative. (2) AC, an argumentation mining task with very few data points based on (Peldszus and Stede, 2015) in which the goal is to classify sentences as to whether they contain a claim or not. (3) CLS, a task based on (Prettenhofer and Stein, 2010) to identify *individual sentences* as being part of a positive or negative book review.[1]

Table 4.1 contains example sentences and shows statistics for the tasks.

Word embeddings. We use four diverse, potentially complementary types of word embeddings as the basis for our sentence representation techniques: GloVe embeddings (GV) (Pennington et al., 2014) trained on Common Crawl; Word2Vec (Mikolov et al., 2013b) trained on GoogleNews (W2V); Attract-Repel (AR) (Mrkšić et al., 2017) and MorphSpecialized (MS) (Vulić et al., 2017).

We use pre-trained word embeddings except for Attract-Repel where we use embeddings obtained with the retrofitting code of (Komninos and Manandhar, 2016).

[1] The original CLS was built for *document* classification.

Evaluated approaches. For each type of word embedding, we evaluate the standard average ($p = 1$) as sentence embedding as well as different power mean concatenations. We also evaluate concatenations of embeddings $\mathbf{s}^{(i)}(1, \pm\infty)$, where i ranges over the word embeddings mentioned above.[2] We motivate this choice later in our analysis.

We compare against the following four approaches: SIF (Arora et al., 2017), applied to GloVe vectors; average Siamese-CBOW embeddings (Kenter et al., 2016) based on the Toronto Book Corpus; Sent2Vec (Pagliardini et al., 2018), and InferSent (Conneau et al., 2017).

While SIF ($d = 300$), average Siamese-CBOW ($d = 300$), and Sent2Vec ($d = 700$) embeddings are relatively low-dimensional, InferSent embeddings are high-dimensional ($d = 4096$). In all our experiments, the maximum dimensionality of our concatenated power mean sentence embeddings does not exceed $d = 4 \cdot 3 \cdot 300 = 3600$.

Evaluation procedure We train a logistic regression classifier on top of sentence embeddings for our added tasks with random subsample validation (50 runs) to mitigate the effects of different random initializations. We use SGD with Adam and tune the learning rate on the validation set. To allow for a direct comparison against previously published results, we use the SentEval tool (Conneau and Kiela, 2018) for MR, CR, SUBJ, MPQA, TREC, and SST. For most tasks, this tool likewise uses logistic regression with cross-validation.

We report macro F1 performance for AM, AC, and CLS to account for imbalanced classes, and accuracy for all tasks evaluated using SentEval.

4.1.3.2 Monolingual Results

Table 4.2 compares all embedding techniques across all transfer tasks. The results show that we can substantially improve average word embeddings when concatenating multiple word embedding types. All four embedding types concatenated achieve 2pp (percentage points) improvement over the best individual embeddings (GV). Incorporating different power means also substantially improve performances. GV improves by 0.6pp on average, W2V by 1.9pp, MS by 2.1pp and AR by 3.7pp when concatenating $p = \pm\infty$ to the standard value $p = 1$ (dimensionality increases from 300 to 900). The combination of concatenation of embedding types and power means yields an average improvement of 3pp over the individually best embedding type.

However, there is one caveat with concatenated power mean embeddings: both the concatenated embeddings and the different power means live in their own "coordinate system", i.e., they may have different ranges and are potentially scaled differently. Thus, we subtract the column-wise mean of the embedding matrix and divide by the standard deviation, which is equal to the z-norm operation proposed in (LeCun

[2] Monolingually, we limit our experiments to the three named power mean operations to not exceed the dimensionality of InferSent.

Model	Σ	AM	AC	CLS	MR	CR	SUBJ	MPQA	SST	TREC
Arithmetic Mean Embeddings										
GloVe (GV)	77.2	50.0	70.3	76.6	77.1	78.3	91.3	87.9	80.2	83.4
Word2Vec (W2V)	76.1	50.6	69.4	75.2	76.3	74.6	89.7	88.2	79.9	81.0
Morph Specialized (MS)	73.5	47.1	64.6	74.1	73.0	73.1	86.9	88.8	78.3	76.0
Attract-Repel (AR)	74.1	50.3	63.8	75.3	73.7	72.4	88.0	**89.1**	78.3	76.0
GV⊕W2V⊕MS⊕AR	**79.1**	**53.9**	**71.1**	**77.2**	**78.2**	**79.8**	**91.8**	**89.1**	**82.8**	**87.6**
Power Mean Embeddings	**p-values** $= [-\infty, 1, \infty]$									
GV	77.9	54.4	69.5	76.4	76.9	78.6	92.1	87.4	80.3	85.6
W2V	77.9	55.6	71.4	75.8	76.4	78.0	90.4	88.4	80.0	85.2
MS	75.8	52.1	66.6	73.9	73.1	75.8	89.7	87.1	79.1	84.8
AR	77.6	55.6	68.2	75.1	74.7	77.5	89.5	88.2	80.3	89.6
GV⊕W2V⊕MS⊕AR	80.1	58.4	71.5	77.0	78.4	80.4	**93.1**	88.9	83.0	90.6
→ with z-norm†	**81.1**	**60.5**	**75.5**	**77.3**	**78.9**	**80.8**	93.0	**89.5**	**83.6**	**91.0**
Other Sentence Embeddings										
GloVe + SIF	76.1	45.6	72.2	75.4	77.3	78.6	90.5	87.0	80.7	78.0
Siamese-CBOW	60.7	42.6	45.1	66.4	61.8	63.8	75.8	71.7	61.9	56.8
Sent2Vec	78.0	52.4	72.7	75.9	76.3	80.3	91.1	86.6	77.7	**88.8**
InferSent	**81.7**	60.9	72.4	**78.0**	81.2	**86.7**	**92.6**	90.6	**85.0**	88.2
→ with z-norm†	81.6	**61.1**	**73.0**	**78.0**	**81.3**	85.6	92.5	**90.7**	84.1	87.8

Table 4.2: Monolingual results. †=we normalized the embeddings of our full model with the z-norm as proposed by LeCun et al. (1998).

et al., 1998). The normalization of embeddings improves the performance by 1.0pp.[3] In total, we considerably close the gap to InferSent from 4.5pp (for GV) to 0.6pp (or 86%) while having a lower dimensionality (d=3600 vs. d=4096).

We consistently outperform the lower-dimensional SIF, Siamese-CBOW, and Sent2Vec embeddings. We conjecture that these methods underperform as universal sentence embeddings[4] because they each discard essential information. For instance, SIF assigns low weight to common words such as discourse markers, while Siamese-CBOW similarly tends to assign low vector norm to function words (Kenter et al., 2016). However, depending on the task, function words may have critical signaling value. For instance, in AM, words like "thus" can indicate argumentativeness.

While the representations learned by Siamese-CBOW and SIF are lower-dimensional than both our own representations and even more so compared to those of InferSent, we find it surprising that they both perform below the GV baseline on average. Sent2Vec (d=700) outperforms GV, but performs below the concatenation of GV and W2V (d=600). This challenges their status as hard-to-beat baselines across many different transfer tasks. We further note that our concatenated power mean word embeddings outperform much more resource-intensive approaches such as Skip-thought in 4 out of 6 common tasks reported in Conneau et al. (2017) and the neural MT (en-fr) system reported there in 5 of 5 common tasks.

[3] For InferSent, the same normalization operation decreases the performance scores by 0.1pp on average.

[4] SIF and others were initially only evaluated in textual similarity tasks.

The performance of our best combination is very close to that of InferSent while being lower-dimensional and considerably cheaper to compute at run-time. Furthermore, we do not rely on high-quality natural language inference data at training time, which may be unavailable for many languages. This demonstrates our method's advantages and adds crucial new perspectives to previous work: concatenated power mean word embeddings are surprisingly difficult to beat.

4.1.3.3 Dimensionality in Relation to Performance

Figure 4.1 investigates the relationship of dimensionality and performance based on our previous experiments. We see that larger embedding sizes result in higher average performance scores. More precisely, we observe a sub-linear growth in average performance as we increase the embedding size by concatenating diverse word embeddings. This holds for both the standard concatenation of average ($p = 1$) embeddings and the power mean concatenation with $p = 1, \pm\infty$. Furthermore, we observe that the concatenation of diverse average ($p = 1$) word embeddings typically outperforms the concatenation of different power means with the same dimensionality ($p = 1, \pm\infty$). For example, concatenating arithmetic averages of GV, W2V, and MS embeddings ($d = 900$) outperforms the power mean concatenations ($p = 1, \pm\infty$) of GV. Similarly, GV$\oplus$W2V$\oplus$MS$\oplus$AR ($d = 1200$) outperforms the even higher-dimensional GV$\oplus$W2V$_{[1,\pm\infty]}$ ($d = 1800$). This suggests that there is a trade-off for the considered power mean concatenations: while they typically improve performance, the increase is accompanied by an increase in embedding size, which makes alternatives (e.g., concatenation of arithmetic average embeddings) competitive. However, when a further concatenation of more embedding types is not possible—e.g., because no more are available—or when re-training of a given embedding type with higher dimensionality is unfeasible—e.g., because the original resources are not available or because training times are prohibitive— concatenation of power mean embeddings offers a strong performance increase by capturing a better summary of the present information.

We believe that these findings illustrate that sentence embeddings research should consider our embeddings as a fair baseline to higher-dimensional sentence embeddings instead of merely reporting the performance of average GloVe embeddings. We further analyze the importance of comparing embeddings of similar sizes and the effect of normalization in (Eger et al., 2019a).

4.1.4 Cross-Lingual Experiments

In our monolingual experiments, we have established that concatenated power mean embeddings are universal across a wide range of classification tasks and represent a genuinely hard-to-beat baseline to more complex methods. We now extend the notion of universality to the cross-lingual case, by transferring sentence embeddings across languages.

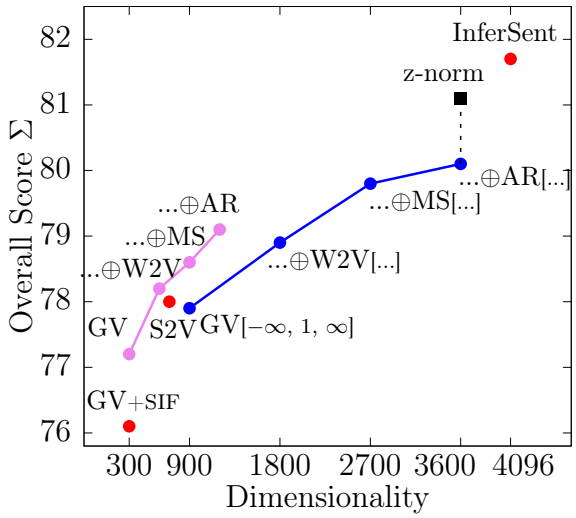

Figure 4.1: The average monolingual performance for the different sentence embeddings in relation to their dimensionality. We visually group related embeddings (i.e., average and power mean embeddings). S2V is Sent2Vec.

4.1.4.1 Cross-Lingual Setup

Tasks We use German (de) and French (fr) translations of all sentences in our 9 transfer tasks. Sentences in AC are already parallel (en, de), having been (semi-)professionally translated by humans from the original English text. For AM, we use student translations from the original English text into German (Eger et al., 2018a). CLS (en, de, fr) is also available bilingually. For the remaining datasets, we leverage machine translated versions obtained with Google Translate for the directions en-de and en-fr.

Word embeddings Since our monolingual embeddings are not all available cross-lingually, we use alternatives:

- We use en-de and en-fr BIVCD (BV) embeddings, that were trained on aligned sentences from the Europarl (Koehn, 2005) and the UN corpus (Ziemski et al., 2016), respectively, using word2vec;
- Attract-Repel (AR) (Mrkšić et al., 2017) provides pre-trained cross-lingual word embeddings for en-de and en-fr;
- Monolingual FastText (FT) word embeddings (Bojanowski et al., 2017) of multiple languages trained on Wikipedia, which we map into a shared vector space with a non-linear projection method similar to the ones proposed in (Wieting et al., 2015), but with necessary modifications to account for the cross-lingual setting. We provide the technical details of our cross-lingual mapping procedure in Appendix C and note that there exist a wide range of other methods (e.g., Joulin et al., 2018; Artetxe et al., 2018; Patra et al., 2019; Glavaš and Vulić, 2020). A comparative evaluation of such methods in the context of cross-lingual sentence embeddings is outside the scope of this section.

We also re-map the BV and AR embeddings using our technique. Even though

BV performances were not affected by this projection, AR embeddings were greatly improved by it. All our cross-lingual word embeddings have $d = 300$.

Evaluated approaches Similar to the monolingual case, we evaluate standard averages ($p = 1$) for all embedding types, as well as different concatenations of word embedding types and power means. Since we have only three rather than four types of word embeddings here, we additionally report results for $p = 3$. Again, we motivate our choice below.

We also evaluate bilingual SIF embeddings, i.e., SIF applied to bilingual word embeddings, CVM-add of Hermann and Blunsom (2014) with dimensionality $d = 1000$ trained on sentences from Europarl and the UN corpus,[5] and three novel cross-lingual variants of InferSent:

1. InferSent MT: We translated all 569k sentences in the SNLI corpus (Bowman et al., 2015) to German and French using Google Translate.[6] To train, e.g., en-de InferSent, we consider all 4 possible language combinations over each sentence pair in the SNLI corpus. Therefore, our new SNLI corpus is four times as large as the original.

2. InferSent TD: We train the InferSent model on a different task where it has to differentiate between translations and unrelated sentences (translation detection), i.e., the model has two output classes but has otherwise the same architecture. To obtain translations, we use sentence translation pairs from Europarl (en-de) and the UN corpus (en-fr); unrelated sentences were randomly sampled from the respective corpora. We limited the number of training samples to the size of the SNLI corpus to keep the training time reasonable.[7]

3. InferSent MT+TD: This is a combination of the two previous approaches where we merge translation detection data with cross-lingual SNLI. The two label sets are combined, resulting in 5 different classes.

We trained all InferSent adaptations using the cross-lingual AR word embeddings, which yielded the best results in initial experiments.

Evaluation procedure We replicate the monolingual evaluation procedure and train the classifiers on English sentence embeddings. However, we then measure the transfer performance on German and French sentences (en→de, en→fr).

4.1.4.2 Cross-Lingual Results

We report average results over en→de and en→fr in Table 4.3. Detailed per-language scores are included in Appendix B.1.

[5] We observed that $d = 1000$ performs slightly better than higher-dimensional CVM-add embeddings of $d = 1500$ and much better than the standard configuration with $d = 128$. This is in line with our assumption that single-type embeddings become better with higher dimensionality, but will not incorporate additional information beyond a certain threshold.

[6] At the time of publication, there existed no cross-lingual variant of NLI. Today, it would be possible to use XNLI (Conneau et al., 2018), which contains NLI data for 15 languages.

[7] Also, adding more data did not improve performances.

	Σ	AM	AC	CLS	MR	CR	SUBJ	MPQA	SST	TREC
Arithmetic Mean										
BIVCD (BV)	67.3	40.5	67.6	66.3	64.4	71.7	81.1	81.6	65.7	67.0
Attract-Repel (AR)	69.2	38.6	68.8	68.9	68.2	73.9	82.8	84.4	72.5	64.5
FastText (FT)	68.3	38.4	63.4	70.0	69.1	73.1	85.1	81.5	69.3	65.1
BV⊕AR⊕FT	**71.2**	40.0	67.7	**71.6**	**70.3**	**76.8**	**86.2**	84.7	73.3	70.5
Power Mean [p-values]										
BV $[1, \pm\infty]$	68.7	48.0	68.8	65.8	63.7	72.2	82.5	81.3	66.9	69.5
AR $[1, \pm\infty]$	71.1	44.2	67.8	68.7	68.8	75.5	84.3	84.4	73.0	73.5
FT $[1, \pm\infty]$	69.4	43.9	64.2	69.4	67.6	73.4	85.8	81.4	73.2	65.5
BV⊕AR⊕FT $[1, \pm\infty]$	73.2	50.2	**69.3**	**71.5**	70.4	**76.7**	86.7	84.5	75.2	74.3
BV⊕AR⊕FT $[1, 3, \pm\infty]$	**73.6**	**52.5**	69.1	71.1	**70.6**	**76.7**	**87.5**	**84.9**	**75.5**	**74.8**
Other Sentence Embeddings										
AR + SIF	68.1	38.4	67.7	**69.1**	67.7	73.8	81.6	**81.7**	70.0	63.2
CVM-add	67.4	47.8	68.9	64.2	63.4	70.3	79.5	79.3	70.2	67.8
InferSent MT	71.0	49.3	69.8	67.9	69.2	**76.3**	**84.6**	76.4	**73.4**	72.3
InferSent TD	71.0	**51.1**	**72.0**	67.9	68.9	74.7	84.3	76.8	72.7	71.0
InferSent MT+TD	**71.3**	50.2	71.3	67.7	**69.6**	76.2	84.4	77.0	72.1	**73.2**

Table 4.3: The cross-lingual results, averaged over en→de and en→fr. Per-language scores are included in Appendix B.1.

As in the monolingual case, we observe substantial improvements when concatenating different types of word embeddings of ∼2pp on average. Conversely, incorporating different power means is more effective than in the monolingual case, considerably improving performance scores compared to arithmetic mean word embeddings. On average, the concatenation of word embedding types plus different power means outperforms the best individual word embeddings by 4.4pp cross-lingually, from 69.2% for AR to 73.6%.

We considerably outperform the InferSent adaptations by more than 2pp on average cross-lingually, with consistent improvements in 8 out of 9 individual transfer tasks. Further, we perform on-par with InferSent already with dimensionality $d = 900$, either using the concatenation of our three cross-lingual word embeddings or using AR with three power means ($p = 1, \pm\infty$).

One reason for this corresponds to one of our method's most important advantages: because of its training-free composition, we can make use of several pre-existing high-quality word embeddings. In contrast, techniques such as InferSent require high-quality annotated training data, which has not been broadly available cross-lingually at the time we conducted our experiments. Automatic translation of this NLI data can yield noisy training pairs, which may explain (in parts) the less effective sentence representations. Future work could explore this in more detail, e.g., by training InferSent with machine translations of varying qualities to quantify how much this effect contributes to the observed cross-lingual performance decrease. The simplicity of our method makes it straightforward to extend concatenated power mean embeddings to other languages because good cross-lingual word embeddings can be obtained with little bilingual data (Zhang et al., 2016b; Artetxe et al., 2017;

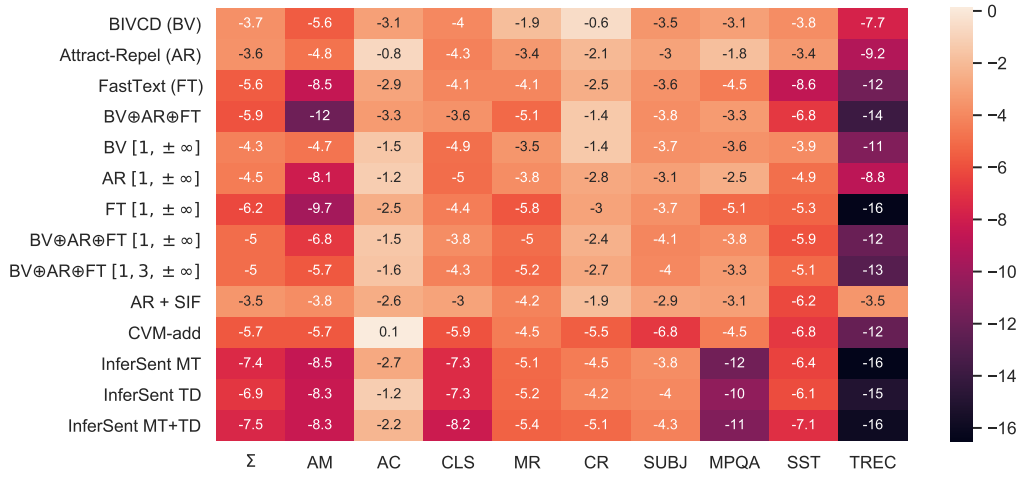

Figure 4.2: The cross-lingual performance degradation, i.e., the cross-language performance minus the in-language performance.

Vulić et al., 2019; Glavaš et al., 2019).

4.1.5 Analysis

Machine translations. To test the validity of our evaluations that are based on machine translations, we compared performances when evaluating on machine (MT) and human translations (HT) of our two parallel AM and AC datasets.

We re-evaluated the same 14 methods as in Table 4.3 using MT target data. We observe a Spearman correlation of $\rho = 96.5\%$ and a Pearson correlation of $\tau = 98.4\%$ between HT and MT for AM. For AC, we find a ρ value of 83.7% and a τ value of 89.9%. While the latter correlations are lower, we note that the AC scores are relatively close in the direction en→de, so small changes (which may also be due to chance, given the dataset's small size; see Table 4.1) can lead to rank differences. Overall, this indicates that our MT experiments yield reliable rankings, which strongly correlate to performance values measured on HT.

Cross-lingual performance decrease. For all models and tasks, we observed decreased performances in the cross-lingual evaluation compared to the in-language evaluation, which is shown in Figure 4.2. For instance, we see a substantial difference between the performance decreases of our best model (-5pp on average) and our best cross-lingual InferSent adaptation (-7.5pp). Two reasons may explain these observations:

1. InferSent is a complex approach based on a bidirectional LSTM. In the same vein as (Wieting et al., 2016), we hypothesize that embeddings learned from complex approaches transfer less well across domains compared to embeddings derived from simpler methods such as power mean embeddings. In our case, we transfer across languages, which can be considered a particularly strong form of domain shift.

Power mean-values	Σ X-Ling	Σ In-Language
$p = 1, \pm\infty$	73.2	78.2
$p = 1, \pm\infty, -1$	59.9	61.6
$p = 1, \pm\infty, 0.5$	73.0	78.6
$p = 1, \pm\infty, 2$	73.4	78.5
$p = 1, \pm\infty, 3$	73.6	78.6
$p = 1, \pm\infty, 2, 3$	**73.7**	78.7
$p = 1, \pm\infty, 0.5, 2, 3$	73.6	**78.9**

Table 4.4: Average scores (en→de and en→fr) for additional power means (based on BV⊕AR⊕FT).

2. InferSent requires a large amount of high-quality training data. In the cross-lingual case we rely on translated sentences for training. Even though we found that these translations are of high quality, they can still introduce noise because some aspects of meaning in languages can only be approximately captured by translations. This effect could increase with more distance between languages. In particular, we observe a higher cross-language drop for the language transfer en→fr than for en→de (see Figure B.2 and Figure B.1 in the Appendix). This difference is less pronounced for our power mean embeddings than it is for InferSent, further supporting our assumption.

Different power means. We performed additional cross-lingual experiments based on the concatenation of BV⊕AR⊕FT with additional power means. In particular, we test (1) if some power means are more effective than others, and (2) if using more power means, and thus increasing the dimensionality of the embeddings, further improves performances.

We chose several intuitive p-values in addition to the ones we used before, namely $p = -1$ (harmonic mean), $p = 0.5$, $p = 2$ (quadratic mean), and $p = 3$ (cubic mean). Table 4.4 reports the average performances over all tasks. We notice that $p = 3$ is the most effective power mean here and $p = -1$ is (by far) least effective. We discuss below why $p = -1$ may hurt the performances in this case. For all cases with $p > 0$, the concatenation of additional power means tend to improve the results further, with decreasing marginal returns. This also shows that improvements are not merely due to additional dimensions but due to the addition of complementary information.

4.1.6 Discussion

Why is it useful to concatenate different power means? The average of word embeddings discards a lot of information because similar averages can represent different sentences. The concatenation of different power means can yield a more accurate summary because it reduces the uncertainty about the sentence's semantic variation. For example, suppose a hypothetical word embedding space that reserves one dimension for representing the sentiment that is typically associated with a word. Knowing the minimum and the maximum of this value across all words in a sentence

can be useful to better determine the sentiment expressed in that sentence.

Not all power means are equally promising. Large values for $|p|$ quickly converge to the minimum ($p = -\infty$) and the maximum ($p = \infty$) of the sequence. Therefore, besides the minimum and maximum, further promising p-values are typically small numbers, e.g., $|p| < 10$. If they are discrete, then odd numbers may be preferable over even ones because even p-values will result in the loss of sign information. More importantly, positive p-values are preferable over negative ones (see our results in Table 4.4) because negative ones will result in discontinuous power means for negative input numbers.

How can we leverage sentence embeddings in applications? Fruitful application scenarios of sentence embeddings are settings in which learning adequate neural network models is infeasible, e.g., due to insufficient labeled training data. In such cases, sentence-level approaches typically outperform task-specific sentence representations induced from word-level models (Subramanian et al., 2018a). This opens a wide range of potential use-cases: when labeled data is nonexistent for our task, we can hand-label a small number of instances—e.g., questions in cQA with question-types (similar to the "TREC" task)—and train a simple classifier on top of sentence embeddings to obtain a suitable model.

Others use sentence embeddings for computational efficiency reasons: the INCEpTION annotation platform (Klie et al., 2018) provides a module for interactive annotation recommendation that leverages sentence embeddings for fast and accurate model training. Annotators label sentences of a text corpus to construct a novel dataset, and during annotation, a classifier is frequently re-trained on top of the sentence embeddings to adapt the recommendations. Recent work builds upon similar approaches for computationally efficient learning-to-rank in the context of interactive entity linking recommendation (Klie et al., 2020).

Application scenarios of **cross-lingual sentence embeddings** are cases in which we do not have access to labeled target language training data. Even though we could, in theory, machine translate sentences from the target language into English and apply a monolingual classifier, state-of-the-art MT systems like Google Translate currently only cover a small fraction of the world's ∼7000 languages. However, using cross-lingual sentence embeddings, we can train a classifier on English and then directly apply it to low-resource target language sentences. This so-called direct transfer approach (Zhang et al., 2016a) on the sentence-level can be beneficial when training data is scarce.

4.1.7 Conclusion

We proposed concatenated power mean word embeddings, a conceptually and computationally simple method for inducing sentence embeddings that combines two important ingredients:

1. The concatenation of diverse word embeddings, which store different kinds of information. This allows us to inject complementary information in the resulting representations.

2. The use of power means to extract different types of summary statistics from the sequence of (concatenated) word embeddings. This allows us to capture more information from the sequence.

Our proposed method is **training-free** and narrows the monolingual gap to state-of-the-art supervised methods. We have extended the property of universality from the monolingual to the **cross-lingual** case and substantially outperformed several cross-lingual adaptations of InferSent.

We have established that our embeddings are a challenging and truly hard-to-beat baseline across a large array of classification tasks and across languages. This has been widely recognized in recent work (e.g., Kayal and Tsatsaronis, 2019; Yang et al., 2019b; Schmidt et al., 2019; Indurthi et al., 2019; Gao et al., 2020; Wang et al., 2020a). Researchers have also applied and adapted our approach to other tasks. Böhm et al. (2019) leveraged our monolingual embeddings in the context of reinforcement learning based document summarization. They showed that when used within learned reward functions, concatenated power mean embeddings can induce better representations than CNNs (Kim, 2014), but are outperformed by BERT (Devlin et al., 2019). Furthermore, Cao et al. (2020) have used our technique to augment a FastText baseline for the cross-lingual word retrieval task, reporting partly better performances than for some word-aligned BERT variants.

4.2 Improving Cross-Lingual Question Similarity with Back-Translation

Parts of this section have been previously published as listed below. Verbatim quotes from this publication are included in this section.

Andreas Rücklé, Krishnkant Swarnkar, and Iryna Gurevych: 'Improved Cross-Lingual Question Retrieval for Community Question Answering', in: *Proceedings of the 2019 World Wide Web Conference* (WWW 2019), pp. 3179–3186, May 2019.

My contributions: Training and evaluating neural machine translation models, adapting transformer with back-translation (synthetic parallel data), analyses of experimental results, error analysis, concept and design of the experimental setup, overall project roadmap. The publication was authored by me.

Krishnkant Swarnkar contributed to this publication as an undergraduate student during his internship at UKP Lab, supervised by me. He helped me with the data collection (retrieving data from Google Translate, processing StackOverflow dumps for dataset construction) and helped to fine-tune RCNN models according to my task description. All co-authors contributed to the models, evaluations, and analyses during discussions.

We studied universal sentence embeddings for sentence classification in the previous section, focusing on the cross-lingual case. Our embeddings transfer well across different tasks and languages, and they have a wide array of possible applications, e.g., determining the types of questions posed in different languages. However, when enough labeled data is available for a task of interest, word-level models often outperform sentence embeddings, e.g., see the experimental results of Conneau et al. (2017). For the cross-lingual case, this means that when enough labeled data is

available in English, we can train a suitable word-level model monolingually and machine translate the target language text to English for cross-lingual transfer.

A particularly appealing use-case for this approach is in cQA question similarity, a task which we have described in detail in Section 2.3. Even though most previous work in question similarity has only considered the monolingual case (e.g., Cao et al., 2012; Dos Santos et al., 2015; Lei et al., 2016b; Nakov et al., 2017; Shah et al., 2018), cross-lingual question similarity can have a broad impact. By comparing questions posed in different languages, we can make the data from the English cQA forums available to more users. This is particularly promising when dealing with **programming and operating systems cQA forums**, where the English forums such as StackOverflow and AskUbuntu are predominant (both are from the StackExchange network, see §2.3.1).

To the best of our knowledge, research prior to our work has only explored the language pair English/Arabic and only one relatively general domain—i.e., questions from the Qatar Living forum (see Table 2.3). Joty et al. (2017) use adversarial training with feed-forward networks to learn language-invariant feature representations for question pairs. Adversarial training requires in-domain questions from both languages, which are typically not available for specialized domains (e.g., there is no Arabic StackOverflow). Furthermore, Da San Martino et al. (2017) apply a mixture of a cross-lingual tree kernel with a dictionary and machine translation (MT) with monolingual features. Importantly, they observed relatively small effects from using MT in their setting. However, applying MT to specialized domains is typically more error-prone because large parallel corpora—which are used to train MT models—often only contain sentences from more general domains (Luong and Manning, 2015; Farajian et al., 2017). Thus, cross-lingual question similarity for **specialized domains**, such as programming cQA forums, presents us with different challenges because it would be challenging to obtain data from the target domain in the target language.

Due to the considerable potential outlined before, this section explores cross-lingual question similarity for programming and operating systems cQA forums. Similar to Da San Martino et al. (2017), we first outline an approach that is common in cross-lingual information retrieval, i.e., to machine translate the query to our target language and then continue with a monolingual model (Hartrumpf et al., 2009; Lin and Kuo, 2010). Unlike previous work, machine translation is more error-prone in our specialized domains, leading to a higher performance decrease for cross-lingual question similarity.

To remedy this, we propose adapting the transformer (Vaswani et al., 2017), a state-of-the-art neural machine translation (NMT) approach, to the idiosyncrasies of our specialized domains by extending the training corpus with synthetic parallel in-domain sentences from cQA forums. The method we use is known as back-translation in MT literature and has proven to be effective in other settings before (Lambert et al., 2011; Sennrich et al., 2016; Poncelas et al., 2018; Edunov et al., 2018)

We investigate the impact of back-translation on cross-lingual question similarity for

the language pair English/German and on two datasets from different cQA forums—the Askubuntu benchmark (Dos Santos et al., 2015; Lei et al., 2016b) and a dataset crawled from StackOverflow. Our results show a large cross-lingual performance decrease compared to the monolingual setup. In our analysis, we observe that this is mostly due to translation errors in our specialized domains. We find that our proposed adaptation with back-translation substantially improves the cross-lingual question similarity performance, narrowing the gap to an approach that leverages an external commercial MT service (Google Translate) by up to 56%.

4.2.1 Cross-Lingual Question Similarity

Given a query question q and a set of n candidate questions $Q' = \{q'_1, q'_2, \ldots, q'_n\}$ from a cQA forum that are retrieved with a search engine, the goal is to re-rank the questions in this set according to each candidate's similarity in regard to q.

In our cross-lingual setup, the query is given in a source language L1 (German) and the candidates are in a target language L2 (English). We indicate languages with a superscript, e.g., q^{L1} stands for the query in the source language and q^{L2} stands for the query in the target language. To perform the above process cross-lingually, given q^{L1} and Q'^{L2}, we first translate q^{L1} to L2 using neural machine translation (NMT). We then continue with a monolingual L2 question similarity model to re-rank Q'^{L2} according to the translated q^{L2}.

This is a standard technique in cross-lingual information retrieval (Hartrumpf et al., 2009; Lin and Kuo, 2010) and it has been applied to cross-lingual question similarity in a general travel-related domain (Da San Martino et al., 2017). We go beyond this by exploring the effects of NMT to the performance of neural question similarity in specialized domains, i.e., programming and operating systems cQA forums. One limitation of this approach is that it can only be applied for language pairs that offer sufficient parallel data for NMT training. If we cannot obtain such data, other approaches may be better suited, e.g., unsupervised retrieval with cross-lingual word embeddings (Litschko et al., 2019; Glavaš and Vulićtajner, 2019) or cross-lingual sentence embeddings as in our previous chapter.

Most importantly, our domains contain a specialized vocabulary and domain-specific idiosyncrasies, which makes NMT—and therefore also our standard cross-lingual approach—more prone to errors. This could be even more pronounced for statistical machine translation (Chu and Wang, 2018), which has been used in previous question similarity work (Da San Martino et al., 2017). Thus, in §4.2.2 we adapt the NMT model to our specialized target domains.

In the following, we present the NMT model and the neural approach to question similarity. We visualize the process in Figure 4.3.

4.2.1.1 Neural Machine Translation (NMT)

NMT is a popular research area in NLP that has seen rapid progress over the past years. The most effective approaches use the encoder-decoder architecture

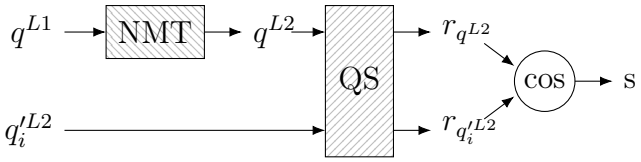

Figure 4.3: The question similarity (QS) model learns (monolingual) question representations, which are then compared with cosine similarity to determine a ranking score s. L2 is always English in our setup.

(Sutskever et al., 2014) and integrate various forms of attention (Vaswani et al., 2017; Gehring et al., 2017; Bahdanau et al., 2015).

In this section, we use transformer (Vaswani et al., 2017), which is a state-of-the-art NMT approach based on multi-head self-attention layers. Because it does not make use of recurrent units, transformer can be efficiently trained on modern hardware while achieving better translation results compared to computationally more expensive models. Edunov et al. (2018), for example, used this approach and achieved state-of-the-art results on the WMT'14 en-de benchmark (Bojar et al., 2014) at the time we conducted the experiments presented in this section.

For our cross-lingual question similarity approach we train two models, one for each direction en→de and de→en using the public WMT'13 (Bojar et al., 2013) and WMT'18 (Bojar et al., 2018) training corpora with more than 4.5M parallel sentences. This is the standard approach of training transformer models and the WMT datasets are openly available. Even though a small number of sentences are on technical topics, there is no specific data that is related to our domains.

We compare transformer with translations from an external commercial translation service, namely Google Translate.[8] Even though we expect Google Translate to produce higher quality translations because their model was likely trained on a large amount of proprietary parallel data, it is not suitable for many practical cQA scenarios due to (1) potential technical restrictions (requiring internet access); (2) legal reasons (confidential information might be transmitted); or (3) associated costs (commercial translation services are not free). The same restrictions do not apply to transformer models because they can be made available on-premise.

In the rest of this work, we indicate the use of transformer with `TR` and the use of Google Translate with `GT`.

4.2.1.2 Question Similarity (QS)

`RCNN` (Lei et al., 2016b) is a question similarity model that combines recurrent units and convolutional neural networks, and it outperforms several other neural models on the AskUbuntu question similarity benchmark (Lei et al., 2016b; Dos Santos et al., 2015). For an input question q, RCNN learns a fixed-size dense vector representation r_q. This representation is obtained by applying the recurrent CNN sequentially on each token in q where the last output state is considered as the

[8] Translations were obtained in Q2/2018. It is unclear which model exactly Google Translate used and how much it has evolved since then.

NMT: Synthetic Parallel Data (§4.2.2.1)		
↷	orig	Unable to change the launcher icon size
↳	en→de	Nicht in der Lage, die Starter-Symbolgröße zu ändern
QR: Data Augmentation (§4.2.2.2)		
↷	orig	How to convert a map to list in java?
↳	en→de	Wie konvertiert man eine Map in Java?
↳	de→en	How to convert a map in Java?

Table 4.5: Examples for our two data-driven adaptations.

question representation. Model details are given in (Lei et al., 2016b). This is a monolingual model, which means that we can only learn representations for texts that are in L2.

To determine the similarity of a candidate question $q_i'^{L2}$ in regard to the query question q^{L2}, RCNN compares the learned representations with cosine similarity to determine a similarity score s:

$$s = \cos\left(r_{q_i'^{L2}}\ ,\ r_{q^{L2}}\right)$$

RCNN uses question duplicates for training, which we denote $(q^{L2},\ q_+^{L2})$. It also obtains one negative sample q_-^{L2} by randomly selecting N questions from the whole corpus and choosing the one with the highest ranking score according to the currently trained model. The training can then be performed with the same procedure as used previously in §3.1.3 using the max-margin hinge loss.

RCNN generally performs better when it is pre-trained in an encoder-decoder setup (Lei et al., 2016b). The encoder RCNN learns a representation of, e.g., the question body, and the decoder tries to reconstruct the title from it. In general, RCNN can use both the question title and the question body during scoring by applying the element-wise average over the learned representations of both texts. The results shown in this section were obtained using only question titles to simplify error analysis and better identify typical error cases. In addition, the performances of models that utilize information from the question body is only marginally better compared to the title-only variant when pre-trained as mentioned above.

4.2.2 Cross-Lingual Adaptations with Back-Translation

We first outline our proposed adaptation using back-translation, which yields synthetic parallel sentences (§4.2.2.1) for training an in-domain NMT model. We then briefly outline a second approach that generates monolingual parallel data for improving the monolingual question similarity model (§4.2.2.2).

4.2.2.1 In-Domain NMT with Synthetic Parallel Sentences

NMT training procedures commonly leverage the aforementioned public WMT corpora, which contain parallel text from the European Parliament Proceedings[9] and

[9] https://www.statmt.org/europarl/; last accessed 29 Dec. 2020.

do not include data from our specialized domains. However, applying NMT models out-of-domain may yield sub-par translation performances (Luong and Manning, 2015; Farajian et al., 2017). Thus, we adapt the transformer model to our specialized domains.

There are several options for domain adaptation in NMT, ranging from data-centric methods that extend the training corpus with in-domain text to model-centric methods that change the loss function, training objective, and neural network architecture (see Chu and Wang (2018) for a survey). We choose a data-centric approach based on back-translation, which has been proven to be effective in both statistical MT (Bertoldi and Federico, 2009; Lambert et al., 2011) and NMT (Sennrich et al., 2016; Poncelas et al., 2018; Edunov et al., 2018) before. Here we train a transformer model on synthetic parallel sentences which we generate from monolingual in-domain texts. To generate these parallel sentences, we apply another transformer model, which we originally trained on the WMT corpora, on sentences from technical cQA forums. We describe our source data in §4.2.3.1.

Because we translate queries from L1 to L2 (see Figure 4.3), we use monolingual L2 data to generate L1 sentences, i.e., the synthetic data is generated by translating in-domain sentences from the target to the source language. This is suitable in our setup because L2 is always English, which means that large-scale in-domain data exists for L2. Also, Lambert et al. (2011) show that using translations that were obtained from the target to the source side results in a better domain adaptation compared to using translations from the source to the target side.

An example of our synthetic data is shown in Table 4.5. We introduce the source dataset later in §4.2.3.1. Based on the synthetic data we then train the in-domain transformer model, which we denote `TR-cQA`, by back-translating the generated sentences from L1 to L2. During training, we also include the WMT corpora.

4.2.2.2 QS Data Augmentation

In addition to adapting the NMT model to our specialized domains, we may also improve the monolingual question similarity model's robustness to common translation mistakes. We consider a procedure that augments the training data with back-translated texts. The titles of the training questions are translated from L2 to L1 and back to L2, which are then added as additional training instances. Back-translating is sometimes used to automatically generate paraphrases, e.g., to train paraphrastic sentence embeddings (Wieting et al., 2017) or to obtain more diverse sentences for other monolingual tasks (Dong et al., 2017; Yu et al., 2018; Iyyer et al., 2018). Due to the close relation to monolingual paraphrasing, we consider this technique as a baseline for assessing the effectiveness of our NMT adaptation.

An example for this data augmentation technique is given in Table 4.5 (bottom), where the aspect of converting something *to a list* is lost during back-translation. This uses GT to obtain back-translations because in initial experiments we found that TR back-translations were too noisy and in many cases they completely altered the meaning of a sentence. The reason is that in the direction en→de we cannot obtain a good TR-cQA model because there exists no large German/L1 technical

Dataset	Number of Examples			Pool Size	Duplicates
	Train	Valid	Test		Valid and Test
AskUbuntu	12 584	189	186	20	6.0
StackOverflow	23 558	1770	2779	20	1.3

Table 4.6: Dataset statistics. "Pool Size" denotes the number of candidate questions that the question similarity model re-ranks. "Duplicates" refers to the average number of duplicates for each query question.

cQA platform from which we could generate synthetic parallel sentences.

We can leverage back-translated texts in both the pre-training and in the regular model training of RCNN. In both phases, a new training example is added for each data point, replacing the original question title with its back-translated text.

We denote the augmented question similarity model `RCNN-A`.

4.2.3 Experiments

The data collection and the fine-tuning of RCNN models was conducted in collaboration with Krishnkant Swarnkar, under the close supervision of the thesis author and following the thesis author's task description (Andreas Rücklé furthermore defined the overall project roadmap, trained and evaluated the neural machine translation models, adapted the transformer with back-translation, performed analyses of experimental results, conducted error analyses).

4.2.3.1 Experimental Setup

Question similarity data. The experiments were conducted on two datasets from different cQA forums. The dataset statistics are given in Table 4.6.

AskUbuntu is a publicly available question similarity benchmark (Dos Santos et al., 2015) that has been extended with additional manual relevance annotations in (Lei et al., 2016b). For the cross-lingual setup, we manually translate the queries of the validation and test splits to German.

StackOverflow has been introduced as part of our publication (Rücklé et al., 2019c), which includes questions that were labeled with "Java" or "Python" tags by the community. Each query question includes 20 potentially similar candidate questions in the validation and test splits, which were retrieved using BM25 (Robertson and Zaragoza, 2009). Duplicates of the query questions are considered as similar. Leveraging such community-labeled instances from StackExchange is a common technique that achieves a high precision but potentially a low recall (Hoogeveen et al., 2016). Due to the dataset's larger size, German query questions were obtained with Google Translate.

Question similarity models and baselines. We report the results of the cross-lingual approaches as described in §4.2.1 (RCNN TR) with all combinations of extensions, i.e., RCNN with in-domain NMT (`RCNN TR-cQA`), with data augmentation

(RCNN-A TR) and with both extensions (RCNN-A TR-cQA). We also include results for the same approach when evaluated using Google Translate instead of transformer to provide a comparison to a commercial state-of-the-art MT system (RCNN GT and RCNN-A GT). RCNN models have been trained with the implementation of Lei et al. (2016a).

TF*IDF, RCNN, and RCNN-A represent the monolingual baselines and an upper-bound.

Question similarity evaluation. We report mean average precision (MAP), mean reciprocal rank (MRR), and precision@1 (P@1; accuracy). To mitigate the effects from random initialization of neural network weights, we report averaged scores over five runs.

Transformer training. We train the TR models with the official transformer implementation and recommended hyperparameters.[10] To obtain synthetic in-domain training data for TR-cQA we use TR to translate the question titles from the training split of both AskUbuntu and StackOverflow datasets to German. This comprises of roughly 36k sentences. We note that better results could potentially be achieved by leveraging more data from AskUbuntu and StackOverflow forums and by applying filtering techniques to exclude back-translations of low quality. We leave the exploration of such extensions for future work.

To train our in-domain model TR-cQA, we implemented a new translation "problem" in the tensor2tensor library, which merges the WMT corpora (with 4.5M instances) with our synthetic in-domain data at training time. We repeat the in-domain data 12 times (totaling 430k instances) to better balance the noisy in-domain and the clean out-domain data.

4.2.3.2 Experimental Results

Table 4.7 and Table 4.8 show the results for StackOverflow and AskUbuntu, respectively. There exist considerable differences between the performances of the monolingual and the cross-lingual question similarity approaches. For instance, compared to the monolingual RCNN, we see that RCNN TR decreases the performance by 6.6 MAP on StackOverflow and by 3.1 MAP on AskUbuntu (test). This contrasts previous work where much smaller effects were observed using phrase-based MT for cross-lingual question similarity (Da San Martino et al., 2017). We conjecture that this is due to the fact that our questions address domain experts and are therefore more difficult to translate automatically.

The translation quality has a strong effect on the cross-lingual question similarity performance—i.e., GT achieves better results compared to TR (3.1 MAP on average over all datasets and data splits). Most cross-lingual approaches perform better compared to the monolingual TF*IDF, which is generally considered as a hard-to-beat baseline in question similarity setups (e.g., Dos Santos et al., 2015).

[10] (Vaswani et al., 2018); using the 'transformer_base_single_gpu' hyperparameter set of https://github.com/tensorflow/tensor2tensor; last accessed 29 Dec. 2020.

	Valid			Test		
	MAP	MRR	P@1	MAP	MRR	P@1
Monolingual						
TF*IDF	58.9	61.0	47.7	54.6	56.7	43.0
RCNN	65.3	67.3	54.5	60.2	62.3	47.9
RCNN-A	**65.9**	**67.9**	**54.9**	**60.8**	**62.9**	**48.7**
Cross-lingual						
RCNN GT	62.3	64.2	50.8	57.7	59.7	45.3
RCNN-A GT	**62.9**	**64.8**	**51.2**	**58.5**	**60.5**	**46.3**
RCNN TR	57.2	59.0	44.5	53.6	55.4	40.4
RCNN TR-cQA	60.2	62.0	48.1	55.4	57.4	42.5
RCNN-A TR	58.1	59.9	45.3	54.4	56.2	41.4
RCNN-A TR-cQA	**61.2**	**63.1**	**49.3**	**56.0**	**58.0**	**43.1**

Table 4.7: Monolingual and cross-lingual performance scores for **StackOverflow**.

	Valid			Test		
	MAP	MRR	P@1	MAP	MRR	P@1
Monolingual						
TF*IDF	53.0	67.5	54.5	52.0	64.5	50.5
RCNN	58.3	72.1	60.1	60.3	73.9	60.6
RCNN-A	**58.4**	**72.8**	**60.5**	**60.6**	**75.7**	**64.2**
Cross-lingual						
RCNN GT	57.3	70.1	56.1	59.0	73.7	60.3
RCNN-A GT	**57.8**	**71.1**	**57.8**	**59.2**	**74.9**	**62.7**
RCNN TR	55.9	68.3	54.3	57.2	70.8	56.9
RCNN TR-cQA	57.0	69.8	55.9	58.3	71.7	58.0
RCNN-A TR	56.1	68.4	54.4	58.3	**73.2**	**60.4**
RCNN-A TR-cQA	**57.6**	**70.5**	**57.4**	**58.4**	72.8	60.3

Table 4.8: Monolingual and cross-lingual performance scores for **AskUbuntu**.

Most importantly, we find that adapting TR to our technical cQA domains with synthetic parallel data can yield large improvements. RCNN TR-cQA consistently outperforms RCNN TR, and it substantially closes the performance gap to RCNN GT. Augmenting RCNN with monolingual data (i.e., paraphrases; RCNN-A) also achieves consistently better results, however, we observe similar trends monolingually. This suggests that RCNN-A TR's improvements may be independent of the cross-lingual setting. In the large majority of cases, RCNN TR-cQA is more effective cross-lingually than RCNN-A TR.

In the context of practical applications, it can be particularly important to close the performance gap to GT. We visualize this in Figure 4.4, which shows the performance improvements of the extensions on top of TR (averaged over dev and test results across both datasets). Adapting TR to our technical cQA domains closes

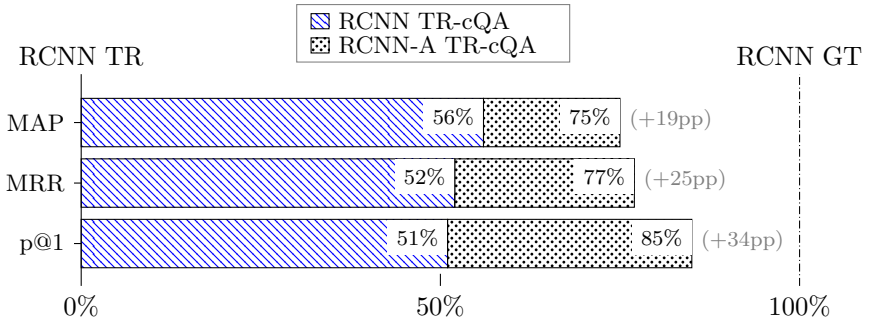

Figure 4.4: A visualization that shows how much the two adaptations close the performance gap between RCNN TR and RCNN GT. Results are averaged over dev and test splits of AskUbuntu and StackExchange datasets.

the performance gap by up to 56%. Combining this with the augmented RCNN model (RCNN-A TR-cQA) can further increase the performance, reducing the gap to RCNN GT by 75–85% (depending on the scoring measure).

All in all, these results show that adapting the NMT model with synthetic parallel data has a strong impact on the cross-lingual question similarity performance. Its independence from the question similarity model makes this approach very versatile and widely applicable.

4.2.4 Analysis

In-domain NMT performance To verify that our observed improvements with TR-cQA over TR are due to improved translations, we evaluate both models with our human translations from the AskUbuntu evaluation splits and measure their translation performance for de→en. TR-cQA achieves 54.96 BLEU whereas TR has a substantially lower score of 41.70 BLEU. Even though the number of parallel sentences is small (375), the large difference suggests that the improvements of RCNN TR-cQA are indeed due to more accurate translations. Moreover, TR-cQA only decreases marginally on the WMT'14 en/de benchmark (28.04 BLEU vs. 28.30 BLEU).[11]

Error analysis We inspect all instances for which RCNN-A TR-cQA ranks a relevant question first and RCNN TR ranks an unrelated question first. We observe that RCNN-A TR-cQA is more robust if the translations slightly differ from the English original query, e.g., when translating "throwing an exception" to "triggering an exception" (which does not carry equal meaning). In most cases, however, the differences are likely due to the improved translations.

Figure 4.9 shows a case where TR discards information ("discrete graphics"). Here, TR-cQA generates a translation that is closer to the original English question. During analysis, we found that this is a common error of TR, which is greatly improved

[11] The BLEU scores in AskUbuntu may be higher because the question titles are considerably shorter compared to WMT'14 sentences.

AskUbuntu

Query: wie kann man diskrete grafik in ubuntu 14.04 deaktivieren
Original text (en): how to disable discrete graphic in ubuntu 14.04

RCNN TR-cQA	RCNN TR
Translation: how to disable discrete graphics in ubuntu 14.04	**Translation:** how to disable grafik in ubuntu 14.04
Top 3	**Top 3**
✓ 14.04 how to disable discrete graphics card	✗ overheating laptop dual ati gpu and discrete
✓ how can i disable ati discrete graphic gpu at startup in ubuntu 14.04 without bios	✓ disabling discrete gpu at startup without system crash
✗ disabling discrete gpu at startup without system crash	✓ how can i disable ati discrete graphic gpu at startup in ubuntu 14.04 without bios

Table 4.9: A comparison of RCNN TR-cQA and RCNN TR on an example from AskUbuntu, showing that TR discards information during translation.

by TR-cQA. Another common issue of TR is that it translates words without considering their domain-specific semantics, e.g., Figure 4.10 shows an example where TR translates "null" to "zero".[12] Such errors are often avoided by TR-cQA.

We also compared RCNN GT, RCNN-A TR-cQA, and the monolingual RCNN to understand how we can further improve the cross-lingual question similarity performances. Most importantly, both RCNN GT and RCNN-A TR-cQA suffer from the same type of problems as described before, but on a smaller scale. This suggests that further improvements can likely be achieved with even better MT domain adaptation.

4.2.5 Why Train Your Own NMT Model?

Commercial MT services often achieve better results compared to publicly available models because they have access to large (proprietary) parallel corpora. The experimental results shown above exhibit a similar trend where RCNN GT still performed better than TR with both our cross-lingual adaptations. Moreover, data augmentation in RCNN resulted in improvements for all models, including the model with GT translations. *Then why can't we just use Google Translate?*

There exist several disadvantages when using external MT services in practical scenarios. For example, they can be costly and the data is transferred to third parties. Consider, for instance, a German company that wants to implement a service for their software developers that helps them to find related English questions within the StackOverflow data dump (if nothing is found, the input question is sent to colleagues via an internal cQA forum). Because questions of software developers

[12] "Null" in German typically refers to the number zero. There exist many similar cases, e.g., TR translates (http) 'post" to "mail".

StackOverflow

Query: try catch exception gibt immer null zurück
Original text (en): try catch exception always returns null

RCNN TR-cQA	RCNN TR
Translation: try catch exception always returning null	**Translation:** try catch exception always returns zero
Top 3	Top 3
✓ catching null exception	✗ catch same exception multiple times
✗ exception handling with multiple catch block	✗ exception handling with multiple catch block
✗ catch same exception multiple times	✗ is it expensive to use try - catch blocks even if an exception is never thrown ?

Table 4.10: A comparison of RCNN TR-cQA and RCNN TR on an example from StackOverflow, illustrating in-domain vs. out-of-domain translation (see null vs. zero in the translations).

can contain critical and confidential information—e.g., information about the software architecture and stack traces—they may not be allowed to send this data to an external service for translation. In such cases, MT needs to be performed with a model that is available on-premise to avoid exposing internal information to third parties.

This is not merely a theoretic scenario. It represents the setting of our project partner DATEV eG, with whom we have collaborated between 2018–2020 as part of our SoftwareCampus project.

4.2.6 Conclusion

We investigated the cross-lingual question similarity performance in programming and operating systems cQA by machine translating query questions from German to English and continuing with a monolingual model. We observed a considerable performance difference in the cross-lingual case compared to the monolingual setup due to translation mistakes within our specialized domains.

To remedy this, we have proposed adapting a state-of-the-art NMT model to our target domains by adding synthetic in-domain parallel sentences to the training corpus. Our adaptation greatly improved the cross-lingual question similarity performance due to better in-domain translations. It considerably narrowed the performance gap by up to 56% to a model with access to an external state-of-the-art commercial MT system, namely, Google Translate—which is arguably not available in many practical scenarios.

4.3 Chapter Summary

In this chapter, we studied the cross-lingual transfer of representation learning models from two complementary perspectives.

First, we proposed concatenated power mean word embeddings as universal **cross-lingual sentence embeddings**, which has fruitful application scenarios where task-specific training data is scarce. Our training-free approach combines the concatenation of complementary word embeddings with different power means to capture more information from the word sequence. We demonstrated that our sentence embeddings outperform several more complex embedding models and are, thus, hard-to-beat monolingually—despite our approach's conceptual and computational simplicity. Moreover, we achieved the best results cross-lingually compared to three different InferSent adaptations, demonstrating our embeddings' universality across tasks and languages. Subsequent work has widely acknowledged our sentence embeddings as a strong baseline to more advanced techniques and our embeddings have been re-used in different applications.

Secondly, we studied the cross-lingual **transfer of monolingual question similarity models** using machine translation. This enables us to make the data from large English cQA forums, which contain enough labeled duplicate questions for model training, available to a broader audience. We have demonstrated the importance and the difficulty of machine translation in English programming and operating systems cQA forums, where the cross-lingual question similarity performance suffers considerably from translation mistakes. We adapted a machine translation model to the idiosyncrasies of these specialized domains by back-translation, thereby narrowing the cross-lingual performance gap to an approach using a commercial state-of-the-art MT system.

We would like to close this chapter by highlighting that we are just beginning to witness an advent for more universally applicable approaches. Large pre-trained language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have not only established monolingual models that achieve state-of-the-art results on most NLP tasks. Their multilingual counterparts mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) also allow for effective zero-shot cross-lingual transfer across similar languages, and they allow for effective few-shot transfer across distant languages (Lauscher et al., 2020b). Moreover, models such as T5 (Raffel et al., 2020) and GPT-3 (Brown et al., 2020) unify several recent innovations in one single model suitable for natural language generation, translation, and classification. Nevertheless, important challenges remain, such as how to obtain simpler and more computationally efficient models[13], and how to close the monolingual and cross-lingual performance gap—areas that we have also touched upon and contributed to in this chapter.

[13] For instance, the largest GPT-3 model contains 175 billion parameters (Brown et al., 2020), which means that it needs to run on specialized hardware.

Chapter 5

Long Answer Selection with Small Training Data

Parts of this chapter have been previously published as listed below. Verbatim quotes from this publication are included in this chapter.

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych: 'COALA: A Neural Coverage-Based Approach for Long Answer Selection with Small Data', in: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence* (AAAI 2019), pp. 6932–6939, January 2019.

My contributions: Neural network design and implementation (COALA and COALA p-means), dataset creation, experimentation (full datasets and few-shot learning), analyses.

In the previous two chapters 3 and 4, we have studied approaches for learning dense vector representations of questions, answers, and, more generally, natural language sentences. We focused on efficient approaches—e.g., by proposing a self-attentive model instead of using cross-attention and with our training-free sentence embeddings. Furthermore, we contributed to making these techniques much more broadly accessible, e.g., by investigating cross-lingual scenarios. Starting with this chapter, in the remainder of the thesis, we continue to emphasize accessibility and efficiency and study a different perspective, namely, how to better deal with **limited labeled training data**.

In this chapter, we address two important limitations of non-factoid answer selection. First, because common models are based on complex deep neural network architectures, they require large amounts of training data. Notably, such data is not available in many cQA forums, let alone non-English data (we review this problem more closely in §5.1.2). We thus need approaches that can effectively and efficiently learn from small training data to handle these scenarios. Second, dealing with long answers is a crucial property in non-factoid QA as opposed to factoid QA. Typically, state-of-the-art answer selection approaches fall short in these cases (Cohen et al., 2018). A likely reason for this is that learning suitable representations for long and complex texts requires more training data.

In this chapter, we tackle these two limitations and address RQ3 of this thesis:

RQ3: How can we train cQA models in settings with limited labeled training data?

We propose a simple and effective task-specific model—compare-aggregate for long answers (COALA)—based on the compare-aggregate framework (He and Lin, 2016; Parikh et al., 2016; Wang and Jiang, 2017). In contrast to previous compare-aggregate models for answer selection—which we outline in §5.1.1—in its simplest form, COALA only contains one trainable layer. This layer learns representations of *text segments* of questions and answers, which we then compare and score with unsupervised techniques. This greatly enhances our model's capabilities in small data scenarios. Because of its task-specific network architecture, COALA is also well-suited to handle long answers.

We also propose an extension to our model, demonstrating that COALA can be further enhanced with more powerful aggregation functions. To achieve this, we use several learned power means (Hardy et al., 1952) to extract more information from the aspect comparisons for aggregation. We have used power means before in Section 4.1, and here, we go beyond that by *learning* suitable values for p.

We study the effectiveness of COALA on seven non-factoid answer selection datasets regarding its performance (a) on the full datasets, (b) in small data scenarios, and (c) in relation to the length of answers. We show that COALA outperforms more complex models such as our LW-BiLSTM of Section 3.1 and the much deeper compare-aggregate model of Wang and Jiang (2017) that previously achieved state-of-the-art results in different text matching tasks including answer selection. More importantly, we show that COALA can handle long answers substantially better than other approaches; it achieves 21pp improvement over the previous state-of-the-art approach when answers are longer than 250 words. Further, due to its simple network architecture, COALA already outperforms a strong unsupervised baseline by more than 3pp and Wang and Jiang (2017)'s model by 32pp when both have access to only 25 training questions.

Therefore, this chapter contributes to obtaining models that are more widely applicable to many small data scenarios in non-factoid answer selection with long answers.

5.1 Background: Compare-Aggregate and Small Data

5.1.1 The Compare-Aggregate Framework

A popular approach for answer selection is the relevance matching model by Wang and Jiang (2017). In contrast to learning representations of questions and answers and comparing them with cosine similarity (see Chapter 3), their model is based on the so-called *compare-aggregate* framework (He and Lin, 2016; Parikh et al., 2016; Wang and Jiang, 2017). Compare-aggregate first compares question and answer words and then aggregates this information. The network finally infers a score based on the aggregated comparisons. Wang and Jiang (2017)'s approach is rather complex and consists of several deep layers; however, it generalizes well to different

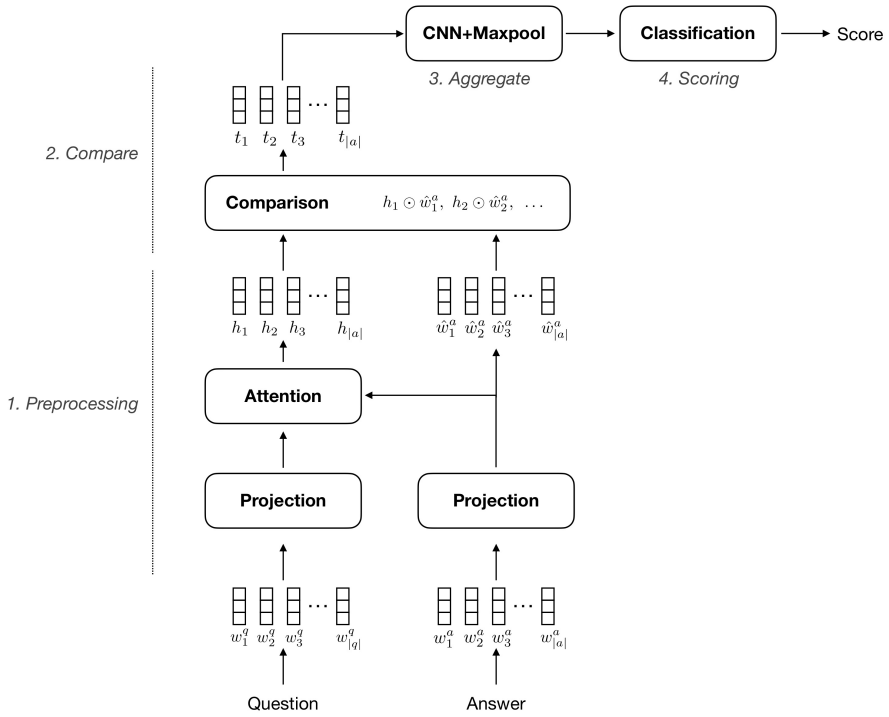

Figure 5.1: An abstract visualization of CA-MTS. $|q|$ and $|a|$ denote the length of the question and answer, respectively. w_i^q and w_i^a are the ith word embedding of the question or answer texts. All layers except for "Comparison" (with the element-wise product $\odot$) contain learned parameters.

tasks, including answer selection and natural language inference (Bowman et al., 2015). Others have proposed similar approaches for related tasks with slight variations in neural network architectures (e.g., Shen et al., 2015; Wang et al., 2016b). Most importantly, Wang and Jiang (2017)'s approach outperforms the previous representation learning models in non-factoid answer selection. In the following, we denote this model `CA-MTS` ("compare aggregate for matching text sequences").

Motivated by CA-MTS's success on different text matching tasks, we use the same compare-aggregate framework for our model COALA. Compare-aggregate is a loosely defined sequence of operations, which yield different outputs such as matrices or vectors (depending on the model). We identify four key components:

1. **Preprocessing**: Question and answer word embeddings are transformed either by context-independent projections or with more complex context-sensitive approaches such as LSTMs. Moreover, attention mechanisms can be applied to capture the interactions between the two texts.

2. **Compare**: The preprocessed embeddings of the question and answer are individually compared with a measure of similarity.

3. **Aggregate**: The comparisons are aggregated, which yields either a vector that contains a compressed view of the comparisons (a summary), or a scalar value (the final score).

4. **Scoring**: If the aggregation result is a vector or a matrix, we learn a trans-

formation to infer the final prediction.

As we can see, questions and answers are encoded jointly, and the models leverage **fine-grained interactions** between the two texts for prediction. This yields better results than independent encoders such as our representation learning approaches in Chapter 3. However, the representations cannot be pre-computed for the whole answer corpus, limiting their computational efficiency.

The compare-aggregate model CA-MTS, visualized in Figure 5.1, consists of several layers with learned parameters, corresponding to our four components:

1. *Preprocessing* learns a projection for each word of the question and answer using a gated importance-weighted representation of words. Further, an attention mechanism learns an alignment between question and answer words using a standard attention mechanism. The jth output vector h_j represents the parts of the question that best match the jth answer word.
2. For *comparison*, CA-MTS combines the results of the attention output and answer words and captures their interactions with element-wise multiplication.[1]
3. A CNN with different window sizes of 1–5 (with max-pooling) then learns an *aggregation* of the interactions.
4. Finally, a two-layer perceptron for *scoring* infers a prediction score based on the CNN output vector.

All components except for the comparison have trainable parameters, which includes the attention mechanism, the CNN with different window sizes, and multiple non-linear projections. Thus, the relatively deep CA-MTS model likely requires training sets of sufficiently large sizes. Furthermore, their comparison and aggregation components determine how well each word in the answer is related to one or more words in the question. Thus, the aggregation depends on the answer length and might not be best-suited to deal with long answers, which are common in cQA and non-factoid answer selection. Our model COALA addresses both limitations.

5.1.2 Small Training Data

Small training data is ubiquitous in cQA due to the large number of specialized forums that do not contain many community-labeled questions and answers. Most research in cQA, however, has been conducted on datasets of sufficiently large sizes. These common datasets contain large quantities of labeled instances, such as in InsuranceQA (Feng et al., 2015) or different versions of Yahoo! Answers (Tay et al., 2017). Some datasets were manually labeled to allow training of a wide range of data-hungry models, e.g., the SemEval cQA challenge datasets for cQA answer selection (Nakov et al., 2017) or the ANTIQUE benchmark (Hashemi et al., 2020). We provided a comprehensive overview of datasets earlier in Section 2.4.

Manual labeling of datasets is expensive and time-consuming, and assuming the availability of large quantities of community labeled instances (e.g., labeled accepted

[1] A key contribution in (Wang and Jiang, 2017) is that they study different measures of similarity. Element-wise multiplications worked best for non-factoid answer selection.

answers) is not realistic for a large number of cQA forums. For instance, as of August 2020 there were 24 forums in the StackExchange network with less than 2000 questions in total—which includes questions without answers or questions with unlabeled answers.[2] Besides, there exist cQA forums such as QatarLiving[3] that do not contain community labels at all.

A task in which research more frequently deals with small training sets is determining question similarity. One reason is that labeled duplicate questions used for model training are even more scarce than labeled answers. We address this scenario later in Chapter 6. In this chapter, we focus on non-factoid answer selection with small training data and study our models' performances on larger datasets to better compare to the previous state of the art.

5.2 COALA: Compare Aggregate for Long Answers

While our general neural network architecture is motivated by CA-MTS's success on different text matching tasks, our proposed approach differs considerably in two key aspects. First, COALA is specific to non-factoid answer selection. We do not address other text-matching tasks such as natural language inference or textual entailment. Thus, we can design our network structure specifically to deal with short questions and long answers. Second, COALA is considerably simpler: instead of attention-based alignment and neural aggregation, it uses max-pooling and averaging techniques, respectively. Therefore, our approach does not require many training instances and is applicable to small data scenarios.

We formalize question-answer matching as follows:

$$f(q, a) = \Omega\left(\Phi(q), \Phi(a)\right) \tag{5.1}$$

where Φ is a function that identifies aspects in the question q and the answer a (e.g., n-grams or syntactic structures), and Ω is a function that scores a based on aspect interactions. In the compare-aggregate framework, Φ refers to the **preprocessing** step, and Ω refers to **compare, aggregate, and scoring**. We emphasize the preprocessing here as it contains the only trainable parameters in COALA without extensions.

In the following, we present our choices of Φ and Ω in more detail. Figure 5.2 shows a visualization of our network architecture.

5.2.1 Preprocessing: Aspect Identification

The core idea of COALA is that both questions and answers address various aspects, which determine the information need (question) and the information content (answer). Correct answers should address as many question aspects as possible. The definition of "aspect" is not well-defined. There can be different approaches of identifying aspects in a text: we could model aspects as individual words, word n-grams

[2] https://stackexchange.com/sites; last accessed: 6 Aug. 2020.

[3] https://www.qatarliving.com/. SemEval cQA shared tasks (Nakov et al., 2016, 2017) used data from this forum, which they manually annotated.

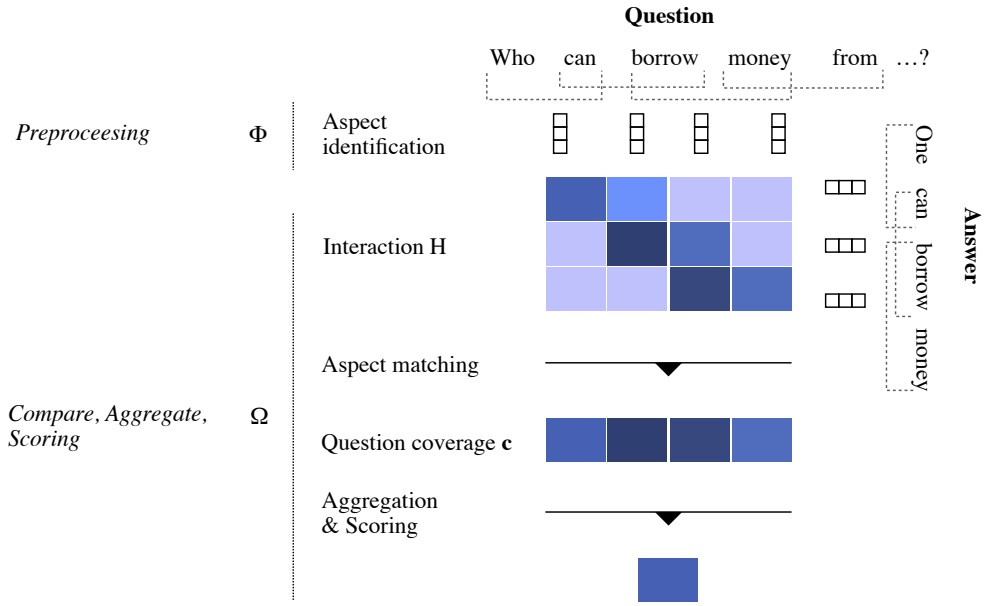

Figure 5.2: A simplified visualization of COALA. Aspect identification learns representations of bigrams with a CNN (no pooling). The interaction layer compares each aspect representation of question and answer with the dot product. Dark colors represent high interactions, i.e., larger values. The aggregation calculates the maximum over rows, which is the coverage c of a question aspect by the answer. Scoring then determines a final similarity score based on c.

(and representations thereof), or more linguistically aware units of the text, which take the syntax or semantic structure of the sentence into account—e.g., syntactic n-grams (Sidorov et al., 2012) or predicate-argument structures.

Here we define the aspects of q and a as vector representations of all observed n-grams[4] and extract them with a convolutional operation. We apply a CNN on $\mathbf{W}^q \in \mathbb{R}^{|q| \times e}$ and $\mathbf{W}^a \in \mathbb{R}^{|a| \times e}$ which represent the sequence of word embeddings of the question q and the answer a, respectively (with dimensionality e):

$$\Phi(q) = \text{CNN}(\mathbf{W}^q) \tag{5.2}$$
$$\Phi(a) = \text{CNN}(\mathbf{W}^a) \tag{5.3}$$

where $\Phi(q) \in \mathbb{R}^{|q| \times d}$ and $\Phi(a) \in \mathbb{R}^{|a| \times d}$ are learned aspect representations (with d CNN filters). We share the parameters between the CNNs and use tanh activation to learn representations with positive and negative components.

Due to the independent processing of different n-grams, our approach likely requires little training data to learn suitable aspect representations (which would be required for learning representations that depend on the entire text of a long answer).

[4] We use $n = 2$, i.e., bigrams. Preliminary experiments showed that bigrams usually achieve the best results with a slight improvement over trigrams and a significant improvement over unigrams.

The extraction of aspects based on word sequences is generally advantageous because it does not require any preprocessing beyond tokenization. Extracting aspects from linguistic structures, however, could result in more informed comparisons between aspects. Thus, later, we propose a linguistically motivated extension of this approach.

5.2.2 Compare, Aggregate, and Scoring

We determine the matching of the question by the answer with three steps: (1) modeling the interaction by comparing question and answer aspects, (2) determining how well the individual question aspects are matched by the answer, and (3) inferring the final score by aggregation.

Interaction. We compute the dot product to capture the interactions between all aspects of q and a in the interaction matrix $\mathbf{H}$. This operation does not introduce additional parameters to the network and has proven successful in other domains before (Cui et al., 2017).

$$\mathbf{H} = \Phi(q)\Phi(a)^{\intercal} \tag{5.4}$$

Here, the value of the ith row and the jth column in $\mathbf{H}$ indicates the similarity of aspect i of q to aspect j of a.

Aspect matching. We now determine how well the ith aspect of Q is covered by all aspects of A by selecting the maximum of each row in $\mathbf{H}$:

$$\mathbf{c}_i = \max_j \left(\mathbf{H}_{i,j}\right) \tag{5.5}$$

It is worth mentioning that—unlike in (Wang and Jiang, 2017)—our aggregation function is now fully independent of the answer's complexity (and length) as we only consider the *best* match of a question aspect by all answer aspects. Furthermore, the aggregation determines how well all *question aspects* are covered (in contrast to aggregating how well the aspects of the answer are related to the question). We thereby explicitly determine how well the answer addresses the question. This also differentiates our approach from others that likewise compute interaction matrices but instead feed them into more complex classifiers (Shen et al., 2015; Wang et al., 2016b; Feng et al., 2017). Our approach is more closely related to the question similarity model of (Zhang and Wu, 2018), who pre-train a transformer encoder and perform interaction and matching as above, but combined with additional features. We instead use CNN, do not require pre-training, and can leverage learned aggregation functions.

Aggregation and scoring. We finally infer a score with an aggregation function g that summarizes the sequence $\mathbf{c}$:

$$\Omega = g\left(\mathbf{c}\right) \tag{5.6}$$

To not introduce additional network parameters, we summarize the values in $\mathbf{c}$ with the arithmetic mean:

$$g(\mathbf{c}) = \frac{1}{|\mathbf{c}|} \sum_{i=1...|\mathbf{c}|} \mathbf{c}_i \tag{5.7}$$

With these operations COALA, contains only a small number of parameters and has a shallow network structure. Both can be advantageous for long answer selection and low-resource scenarios.

5.2.3 Power Mean Aggregation

To extract more descriptive statistics from the sequence $\mathbf{c}$ that determines how well each question aspect is matched by the answer (Equation 5.5), we use the *power mean* (Hardy et al., 1952):

$$power\text{-}mean(\mathbf{x}, p) = \left(\frac{\mathbf{x}_1^p + \cdots + \mathbf{x}_n^p}{n} \right)^{1/p} \tag{5.8}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $p \in \mathbb{R} \cup \{\pm\infty\}$. The power mean naturally generalizes the arithmetic mean (with $p = 1$; see §4.1.2.1 for more details).

In our extended approach `COALA p-means`, we replace the arithmetic mean with m different power means, where we *learn* the values for p as part of the network (initialized with 1.0, i.e., the arithmetic mean). To infer a final score from the summaries, we use a two-layer feedforward network.[5]

Learning the values for p has the unique advantage that we do not need to pre-define different power means. Instead, the network learns the optimal operations.

5.3 Experimental Setup

5.3.1 Data

We study the effectiveness of COALA with and without extensions on several different datasets that cover a broad spectrum of domains for cQA answer selection. We provide an overview in Table 5.1.

InsuranceQA is a well-known answer selection benchmark and was introduced in (Feng et al., 2015). We use the most recent version v2, in which candidate answers are retrieved with a search engine (using the question as a query). WikiPassageQA (Cohen et al., 2018) is a benchmark for passage retrieval where queries are non-factoid questions, and relevant passages are paragraphs from a Wikipedia article. Even though this dataset does not contain forum data, it models a related and realistic scenario.

[5] We use relu as activation for the first layer and sigmoid function for the output layer (to ensure that it is in $[0, 1]$). The number of hidden units is equal to m (number of power means).

Dataset	Number of Questions			Answer Length
	Train	Valid	Test	
InsuranceQA	12 889	1592	1625	112±69
WikiPassageQA	3332	417	416	153±48
Long Answer Selection (LAS) — StackExchange data				
LAS-Travel	3572	765	766	214±174
LAS-Cooking	3692	791	792	189±162
LAS-Academia	2856	612	612	229±168
LAS-Apple	5831	1249	1250	114±110
LAS-Aviation	3035	650	652	281±203

Table 5.1: Statistics of the datasets we use for evaluation. The answer length is the average number of tokens in an answer. Each question has multiple candidate answers of which often only one is correct (e.g., InsuranceQA has 500 candidate answers for each question, and questions in LAS datasets have 100 candidate answers).

For a more thorough evaluation, we also obtain data from travel, cooking, academia, apple (computer), and aviation communities of StackExchange and create datasets that reflect real-life cQA scenarios. We refer to these as "Long Answer Selection" (LAS) datasets. For a given question, we retrieve similar questions from the forum and use their accepted answers as candidate answers to the initial question.[6] The accepted answer to the initial question (and the accepted answers of the question's duplicates) are labeled as correct.

In Table 5.1, we see that one of the distinguishing differences is the length of the answers in the different domains. However, all datasets contain long multi-sentence answer texts (e.g., explanations, descriptions, advice). Notably, most of our datasets contain only a small number of labeled question-answer pairs for training, e.g., between 2.8k–5.8k for LAS datasets. Thus, they represent small-data scenarios compared to InsuranceQA, which contains more than 12k question-answer pairs for training.

5.3.2 Models and Baselines

We compare COALA to several baselines and recent neural models:

(1) *IR baselines*: TF*IDF and BM25 are often considered strong baselines in both cQA (Lei et al., 2016b) and passage re-ranking (Cohen et al., 2018). We use the gensim implementation of BM25 and the sklearn implementation of TF*IDF. We use NLTK's Porter stemmer to preprocess the texts.

(2) *Semantic similarity*: these models compare learned semantic representations of questions and answers with cosine similarity. In addition to the models used in Chapter 3, we also implement Attentive-BiLSTM (Tan et al., 2016).[7]

[6] We use the title of the question and discard the detailed description in the question body. We use ElasticSearch with BM25 to retrieve 100 similar questions.

[7] On InsuranceQA, we report the results from Chapter 3 with 100d word embeddings (vs. 300d used here) due to the large computational effort required to run experiments on this dataset, and

Apart from supervised semantic similarity approaches, we also evaluate *universal sentence embeddings*. To score a candidate answer, we embed the question sentence and all answer sentences and compute the maximum cosine similarity between the question embedding and all answer sentence embeddings.[8] We test two recent models: InferSent (Conneau et al., 2017) and our training-free concatenated power mean word embeddings (Section 4.1, with p-values $[-\infty, 1, \infty]$). Both were not trained for determining the similarity of questions and answers. Nevertheless, they are natural choices in small data scenarios due to their good transfer capabilities to a wide range of downstream tasks.

(3) *Relevance matching methods*: We implement CA-MTS in our framework, which is the compare-aggregate architecture proposed by Wang and Jiang (2017). Since we use bigrams for extracting aspects, we also include a simple bigram model among our baselines, which counts the number of co-occurring bigrams.

5.3.3 Training Procedure

We train the semantic similarity approaches as in Section 3.1. Here we use triples of *(question, answer, incorrect candidate)* and train models with the max-margin hinge loss. During training, we obtain triples by randomly sampling 50 (incorrect) candidate answers and choosing the one with the highest similarity to the question according to the currently trained model.

We train all other approaches with a pointwise approach, i.e., using labeled pairs (question, candidate answer, label), where the label is a binary class *(correct/incorrect answer)*. For each question-answer pair, we sample one corresponding pair of question/incorrect answer during the training with the same method described above. Here we minimize the cross-entropy loss.

We train all models using the Adam optimizer (Kingma and Ba, 2015).

5.3.4 Neural Network Setup

We performed a random search for the hyperparameters of all models (CNN, LSTM, COALA, COALA p-means, CA-MTS). This included, e.g., the number of CNN filters, learning rate, batch size, and dropout rate. Random search for a model and dataset was stopped after 48 hours. We evaluate models with the hyperparameters that achieved the best validation score.[9]

We train models with 300d pre-trained GloVe embeddings (Pennington et al., 2014).

for reasons of consistency. The newly implemented Attentive-BiLSTM uses 300d GloVe word embeddings, which gives this model more representational capacity compared to LW.

[8] In preliminary experiments, we found that this technique outperformed comparisons between the question embedding and the average over all answer sentence embeddings.

[9] The resulting hyperparameters are given in our source code. See https://github.com/UKPLab/aaai2019-coala-cqa-answer-selection.

	∑	IQA	Tr	Co	Ac	Ap	Av	WPQA	
	Accuracy							MAP	MRR
IR Baselines									
BM25	30.3	24.9	38.1	30.9	29.2	21.8	37.0	53.00	61.71
TF*IDF	32.4	18.7	39.9	35.1	32.2	26.7	41.9	39.92	46.38
Semantic Similarity									
InferSent	23.0	14.8	27.0	21.3	22.5	22.8	29.3	43.62	50.53
PMeans	25.7	17.0	32.1	29.3	24.3	19.6	31.7	42.82	50.44
CNN	25.9	24.4	36.9	25.9	22.5	20.2	25.3	27.33	31.48
BiLSTM	34.8	32.4	45.3	35.2	31.5	27.2	37.3	46.16	52.89
Att.-BiLSTM	34.5	37.9	43.0	36.2	31.2	24.7	33.9	47.04	54.36
AP-BiLSTM	31.3	31.9	38.8	32.2	27.3	22.9	34.5	46.98	55.20
LW-BiLSTM	34.1	36.9	43.2	32.3	30.2	23.4	38.5	47.56	54.33
Relevance Matching									
Bigrams	18.3	19.4	19.3	16.7	19.8	13.0	21.5	39.84	47.55
CA-MTS	39.1	37.0	46.5	39.4	36.1	29.2	46.5	48.71	56.11
COALA	43.6	38.0	**53.8**	**47.3**	42.2	32.0	48.4	**60.58**	**69.40**
COALA p-means	**45.2**	**39.9**	53.4	46.5	**44.2**	**34.5**	**52.9**	59.29	68.48

Table 5.2: Experimental results for models trained on the full datasets. Σ shows the average performance over the cQA benchmarks. The following six columns are InsuranceQA, LAS-Travel, Cooking, Apple, Academia, and Aviation.

5.4 Experiments

5.4.1 Results

We report the results of all models across InsuranceQA, WikiPassageQA, and the five StackExchange datasets in Table 5.2. For the cQA answer selection datasets, we measure the accuracy, which is the ratio of correctly selected answers. For the non-factoid answer selection dataset WikiPassageQA, following (Cohen et al., 2018), we report MAP (mean average precision) and MRR (mean reciprocal rank).

The results show that COALA outperforms all other relevance matching and semantic similarity approaches on our seven datasets. For instance, on the cQA answer selection datasets, COALA improves by 4.5pp (accuracy) over CA-MTS and by 8.8pp over the best semantic similarity method, on average. Our extended approach COALA p-means yields an additional gain of 1.6pp. However, the improvements with power mean aggregation are not consistent across all datasets: we observe better performances on four datasets and small decreases in three cases.[10]

The reasons for the better performance of COALA compared to CA-MTS may be two-fold. (1) The training data is likely not large enough for CA-MTS, and indeed, follow-up work of (Deng et al., 2020a) showed that CA-MTS and COALA perform

[10] We also tested feeding **c** into a standard MLP without p-mean aggregation. Here we observed decreased performances in most datasets. Therefore, the improvement of COALA p-means vs. COALA is not just due to the added capacity.

on-par with roughly 76k training questions; (2) There is a mismatch between the negative training instances (sampled semi-randomly, see §5.3) and negative instances at test time (retrieved with a search engine; lexically similar). Due to its strong inductive bias, COALA may be more robust to such variations.

InferSent and concatenated power mean word embeddings do not perform well for cQA answer selection. This indicates that our task requires more information beyond general sentence-level semantic similarity, e.g., domain-specific representations. Supervised semantic similarity models achieve better results on average, but in some cases still perform below IR baselines—especially in our small LAS datasets with very long answers. This suggests that these models require larger training sets to learn suitable representations. Notably, we also find no improvements of attention-based representation learning models over BiLSTM on average for LAS, which further supports this assumption.

COALA also performs the best on WikiPassageQA. Moreover, it achieves an improvement of 3.21 MAP and 1.48 MRR compared to the best results in (Cohen et al., 2018), which they obtained with a complex model, namely, Memory-LSTM-CNN-TF. This demonstrates that COALA can also serve as a strong baseline for related tasks, i.e., our approach is not limited to only cQA data.

Finally, we note that COALA has been extended with structured information, i.e., enhanced dependency parse trees (Schuster and Manning, 2016). Rücklé et al. (2019a) learn syntactic embeddings of the word's dependency relations and concatenate them with word embeddings before aspect identification.[11] This extension yields slight improvements over COALA but performs below COALA p-means (on average). Nevertheless, this suggests that further extensions of our approach are possible, e.g., to obtain better aspect representations.

5.4.2 Few-Shot Learning

While COALA achieves promising results on several datasets with at least two thousand training instances, it is also considerably simpler than other models. All trainable parameters are within the CNN for aspect identification, and its network structure is shallow. This property can be beneficial when we have access to even fewer training instances, which is a realistic scenario because there are many specialized, and thus smaller, cQA platforms.

We study COALA's effectiveness in such few-shot scenarios, and train both COALA and CA-MTS on the cQA datasets (including IQA) from Table 5.2 with a reduced number of 25, 50, 100, ..., 3.2k question-answer pairs.[12]

Figure 5.3 shows the averaged accuracies over all six cQA datasets—all exhibit the same trend with overall smaller differences for "full" on IQA (see Table 5.2). We find that COALA already performs better than the unsupervised IR baseline with

[11] Syntax-based extensions have been explored by Nafise Sadat Moosavi in conjunction with the thesis author.

[12] For experiments with less than 200 question-answer pairs, we average over five runs with different random network initialization, and for the rest, we average over three runs.

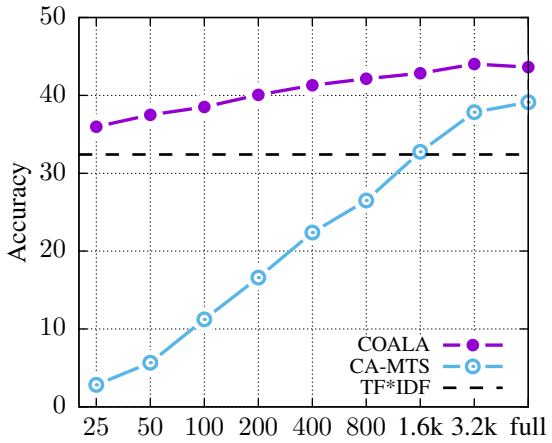

Figure 5.3: Model accuracies (averaged over the six cQA datasets) as a function of available training questions.

25 training questions. This result is remarkable, given that CA-MTS needs at least 1.6k training questions to achieve a comparable performance. At the same time, CA-MTS has a much steeper learning curve, which is the expected behavior for a deep network and partly due to its lower initial performance. When we train both approaches on the full datasets (of which some have less than 3.2k training instances), the learning curve finally flattens.

These results demonstrate that COALA can be applied to various realistic scenarios, e.g., for answer selection in small-scale cQA platforms where only a few questions exist. This is often the case for highly specialized cQA platforms and even more so for non-English platforms. Even if there are no labeled question-answer pairs, it is still possible to use our approach because manual annotation of 25 examples would suffice to train a good model.

5.5 Analysis

5.5.1 Answer Length

In Figure 5.4, we report the average accuracy over all cQA datasets for COALA, CA-MTS, and TF*IDF as a function of the length of correct answers.

We observe that COALA performs better, especially for very long answers. Our approach achieves an average accuracy of 57% for questions with correct answers that are longer than 250 words, which is substantially higher than the accuracy of 36% for CA-MTS. More importantly, we observe a steady increase in COALA's performance as the answer length increases. One potential reason is that longer answers are more likely to cover more aspects of the question, which are then retrieved by COALA. CA-MTS, on the other hand, needs to process the full answer text throughout the whole network, which is more difficult for long answers.

The better results for long answers are due to COALA's strong inductive bias; its coverage mechanism is especially designed for these cases. However, this also means

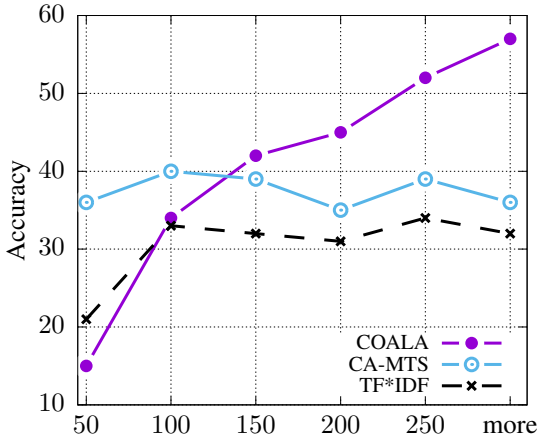

Figure 5.4: Model accuracies as a function of the length of the correct answers (averaged over the six cQA datasets). We group the results into six buckets: answers of length 0–50 (shown as "50"), 50–100 ("100"), ..., and more than 250 ("more").

that our approach is tied to the task of long answer selection, i.e., it is unlikely to perform well for other sequence pair tasks such as NLI.

For answers that are shorter than 100 words, CA-MTS is more effective than COALA. This indicates that processing of the full answer text through a deep network is beneficial (and possible) in these cases. To further study this, we evaluated COALA on WikiQA (Yang et al., 2015), which is a well-known benchmark for factoid *short* answer selection (with an average answer length of 25 words). The results of COALA (69.7 MRR) are on the same level as strong semantic similarity methods such as AP-LSTM (Dos Santos et al., 2016) but considerably below CA-MTS (75.5 MRR) as reported in (Wang and Jiang, 2017). This highlights that different tasks with different answer lengths, particularly cQA answer selection and factoid answer selection, require fundamentally different approaches to obtain optimal results.

5.5.2 Error Analysis

Following the same procedure as in §3.1.5.2, we manually analyze failure cases of COALA on InsuranceQA. In most cases, when COALA selects an incorrect candidate answer, the text either covers all aspects of the question or covers more aspects than the correct answer. The aspects of the question then typically appear individually in the answer but with a different composed meaning. The following question gives an example:

> Does car insurance improve credit?

Here, COALA selects an incorrect candidate answer that covers all essential aspects, i.e., car insurance and improving a credit score:

> Bad credit can have an impact on the premium rates you are asked to pay for car insurance when first applying. Many auto insurers utilize credit scores to make underwriting decisions on new applications. If you have bad credit take steps to improve your score. Shop around for

> auto insurance from companies that use more traditional data in pricing policies. Or stick with your current carrier and drive safely. Of course you want to drive safely no matter what!

However, the individual aspects are composed such that the answer carries a different meaning, which does not match the information need of the question. Hence, the answer should not be selected. Such errors occur because COALA does not leverage information from different aspects simultaneously when processing an answer.

Overall, our observations suggest that COALA is not affected by one of the most critical shortcomings that we identified earlier in §3.1.5.2—i.e., that important aspects of the question are not addressed by the selected answer. However, we believe that it could be beneficial to combine COALA with other approaches that leverage more information from the answer text. For instance, we could achieve this with a two-step ranking process in which COALA selects several candidate answers, and a different model then chooses the final answer that composes the covered aspects such that they match the information need of the question. For instance, such a model could verify whether the covered aspects appear in the same order in both the question and the answer.

5.6 Chapter Summary

In this chapter, we studied cQA answer selection under small data conditions. We investigated compare-aggregate models, which are often deep, complex, and not specialized to a particular task. One of the most effective model architectures by Wang and Jiang (2017) requires large amounts of training data and does not scale well to long answers. To remedy this, we proposed COALA, a much simpler task-specific model architecture based on the same compare-aggregate framework with three important properties:

1. COALA contains only one layer with learned parameters. We can train it with as little as 25 question-answer pairs, which shows that our approach is suitable for few-shot learning. We can apply COALA within cQA forums that do not contain community labels by manually annotating a small number of question-answer pairs.

2. Our approach scales well to long answers, which are common in cQA, and outperforms more complex models on six cQA datasets from different domains.

3. It is possible to extend our model by using learned power means during aggregation, which leads to improvements in our experiments.

We achieve this with a task-specific network architecture that avoids complex learned operations. A downside is then, of course, that COALA cannot be applied to other tasks such as natural language inference. Interestingly, we find that our simplified compare-aggregate model somewhat resembles a shallow variant of the models previously proposed for neural ad-hoc retrieval. These compare query terms to all document terms and infer a score based on kernel pooling and/or multiple convolutional operations with multi-layer perceptrons for scoring (e.g., Xiong et al., 2017;

Hu et al., 2014). Therefore, we believe that adapting ideas from this research area to our scenarios with small data could lead to further interesting extensions. Indeed, our work was later extended with some related concepts by Han et al. (2019), who propose matching n-grams of different lengths and including an additional term weighting component.

Our work has also influenced various other research projects. For instance, Deng et al. (2020b) use COALA as a strong baseline for the task of selecting sentences for extractive answer summarization. Deng et al. (2020a) adapt our setup, using our datasets and models, and propose a two-stage approach consisting of joint learning of answer summarization and answer selection to better deal with long answers. They also studied the effect of pre-training different models on a large data set before fine-tuning them on ours, and found that AP-BiLSTM can benefit greatly from this technique. This shows that pre-training models can be a suitable option for dealing with small data settings.

Our work has also been included in different research projects at UKP. For instance, Moosavi et al. (2019) improve models to natural language inference with a coverage module inspired by COALA's network architecture, and Simpson et al. (2020) use our model as a basis for demonstrating the advantages of their novel interactive preference-based ranking approach.

Finally, we would like to highlight that few-shot learning is currently on the rise in NLP, but is approached with different techniques than outlined in this chapter. For example, pre-training strategies that are compatible with a given target task can yield considerably improved results when having access to only a few target dataset instances (Ram et al., 2021). Converting classification tasks into cloze tasks allows researchers to better leverage the knowledge of massively pre-trained language models, requiring only a small number of instances for model fine-tuning (Schick and Schütze, 2020; Shin et al., 2020). Few-shot language adaptation can be performed by fine-tuning models with a few target language instances after pre-training in a source language (Lauscher et al., 2020b).

In combination with light-weight models, similar to what we have proposed in this chapter, these advances could make state-of-the-art NLP approaches more accessible to a wider range of researchers and practitioners. They can also be applied more easily than notoriously deep models that require large amounts of training data, and they can contribute to decelerating the ever-growing computational demand for training state-of-the-art models; a problem that has been widely recognized recently.[13]

[13] See, e.g., Strubell et al. (2019). In addition, there are dedicated workshops for related topics such as the "First Workshop on Simple and Efficient Natural Language Processing" (Moosavi et al., 2020) and dedicated conference tracks, e.g., "Green and Sustainable NLP" at EACL 2021.

Chapter 6

Training cQA Models Without Labeled Data

Parts of this chapter have been previously published as listed below. Verbatim quotes from this publication are included in this chapter.

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych: 'Neural Duplicate Question Detection without Labeled Training Data, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2019), pp. 1607–1617, November 2019.

My contributions: Proposing duplicate question generation, training models with different training methods, model evaluation, studying performance in relation to increased data sizes, cross-domain question generation, answer selection experiments, BERT experiments, analyses.

Previously, we have studied small data scenarios with a few labeled training instances. However, it is often the case in cQA that **no labeled data** is available at all, particularly for the task of question similarity. As we have seen in Section 2.3, question similarity typically leverages annotated duplicate questions for supervised in-domain training, which are uncommon outside of the StackExchange network and expensive to obtain by manual annotation. For instance, even very popular forums such as GuteFrage.net[1] do not contain any labeled duplicates at all (as of August 2020), illustrating the great need for alternative training methods. In a similar vein, Shah et al. (2018) argue that even many of the larger StackExchange forums do not offer enough duplicates for supervised training of in-domain models.

Three popular approaches tackle this problem in the recent literature: (1) weak supervision with question-answer pairs where we train an answer selection model and use it for determining question similarity (Qiu and Huang, 2015; Wang et al., 2017a); (2) semi-supervised training where we first train a simple non-neural model and use it to generate large amounts of noisy labeled instances (Uva et al., 2018); (3) adversarial domain transfer where we train a model on a source domain and adversarially adapt it to a target domain (Shah et al., 2018).

A significant limitation of these approaches is that they nevertheless rely on substan-

[1] https://gutefrage.net; last accessed 15 Jan. 2021.

tial amounts of labeled data—either thousands of duplicate questions (e.g., from a similar source domain in the case of domain transfer) or large quantities of question-answer pairs. There also exist unsupervised methods based on the encoder-decoder architecture that do not suffer from this shortcoming; however, they impose other crucial limitations. We can only use them with neural network architectures that independently encode the texts, and they usually fall short of the performances that can be achieved with supervised training (Lei et al., 2016b). To train effective question similarity models for cQA forums without labeled data, we need other methods that do not require annotations while optimally performing on-par with supervised in-domain training.

In this chapter we focus on training strategies that leverage **unlabeled questions** from cQA forums. This complements our previous chapter by addressing RQ3 from a different perspective, namely, settings with only unlabeled texts.

RQ3: How can we train cQA models in settings with limited labeled training data?

In particular, we investigate the effectiveness two methods that leverage only unlabeled questions: (1) we propose automatic duplicate question generation (**DQG**); (2) we adapt self-supervised training with title-body pairs (**SSTB**, first studied in a Master thesis of Wiedmeier, 2017, supervised by Andreas Rücklé). Because a question body typically provides additional information not included in the title (Wu et al., 2018), we conjecture that titles and bodies have similar properties as duplicate questions. For instance, question titles and bodies are only partially redundant but fundamentally describe the same information need (Figure 6.1 gives an example). Under this assumption, we can use those texts to train question similarity models.

In DQG, we propose using question generation models to generate a new question title from the question's body. We then consider the generated title as a duplicate to the question's original title. In contrast to most previous work in question generation (QG), we train the QG model itself without annotated data such as question-answer pairs (e.g., Du et al., 2017; Duan et al., 2017; Subramanian et al., 2018a; Zhao et al., 2018; Du and Cardie, 2018).

SSTB directly trains models on title-body pairs—i.e., predicting whether both texts belong to the same question or are from unrelated questions. A master and a bachelor thesis—both supervised and carried out in accordance to detailed task descriptions and project roadmaps authored by Andreas Rücklé—have already provided some indications for the effectiveness of this method (Wiedmeier, 2017; Vatter, 2019). In this chapter, we go considerably beyond this by studying it on a much broader scale with different kinds of models, more tasks, and larger amounts of data.

The advantage of these two methods is that we can make use of a large number of unlabeled questions (titles and bodies) in cQA forums, which can be an order of magnitude more data than it is available for supervised training. Question titles and bodies are common in all StackExchange sites, popular platforms in other languages (e.g., GuteFrage.net), and platforms such as Reddit. A counterexample is Quora,

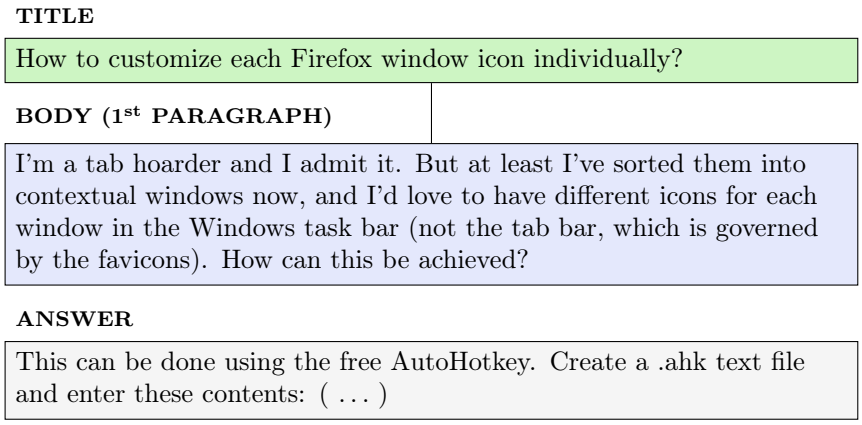

Figure 6.1: An example question, the first paragraph of its body, and the first answer. The example is from SuperUser.com: https://superuser.com/questions/1393090/how-to-customize-each-firefox-window-icon-individually

which only contains question titles. However, there exists a large annotated corpus of question pairs for this platform.

In particular, **our contributions** are to show that:

1. With larger quantities of unlabeled questions as compared to labeled instances, both DQG and SSTB outperform the adversarial domain transfer in the setup of Shah et al. (2018) by more than 5.6pp on average. Because the number of labeled question duplicates is often limited, DQG and SSTB can in some cases achieve better performances than the in-domain supervised training.

2. DQG generalizes well to unseen target domains with minimal impact on the performances. This demonstrates that our method can be broadly applied to train question similarity models, even if we cannot obtain in-domain question generation models.

3. SSTB is also very effective for fine-tuning the more recent BERT model (Devlin et al., 2019), which achieves considerably improved performance scores.

4. SSTB is a suitable choice for training cQA answer selection models without direct answer supervision. Leveraging large quantities of unlabeled questions can yield better answer representations compared to training models with limited numbers of question-answer pairs.

6.1 Background: Question Generation and Title-Body Information

6.1.1 Question Generation

There is a broad body of work for question generation (QG) in the context of reading comprehension. In this setting, the goal is to generate a factoid question based on a given answer sentence or a passage in combination with an answer span.

Neural approaches to QG often use sequence-to-sequence architectures with attention (Du et al., 2017), copy mechanisms (Zhou et al., 2017), and policy gradient techniques (Yuan et al., 2017). Some works leverage paragraph-level information by applying a sentence-level tagger to identify question-worthy sentences (Du and Cardie, 2017), by extracting answers with neural keyphrase detection and rule-based methods (Subramanian et al., 2018b; Yang et al., 2017), by using maxout pointer mechanism (Zhao et al., 2018), or by including co-reference information (Du and Cardie, 2018). Others treat question generation and question answering as dual tasks that are jointly optimized (Song et al., 2017b; Tang et al., 2017; Yang et al., 2017; Wang et al., 2017b).

Question generation with data from cQA forums has been explored by Duan et al. (2017), who propose mining common question patterns from Yahoo! Answers with question clustering techniques, e.g., "Who is _" or "What is _". They use a complex pipeline to obtain common patterns, followed by a component for topic identification. They show that answer sentence selection can be improved with generated questions by optimizing question generation and question answering jointly. Further, Rao and Daumé III (2018b) propose the novel task of ranking *clarification* questions in cQA, i.e., finding existing user comments that ask for additional information regarding user-generated questions.

In contrast to them, and to the best of our knowledge, we were the first to generate *duplicate questions* in cQA and explore the cross-domain capabilities of QG models in the context of question similarity tasks.

6.1.2 Training with Title-Body Information

A variant of self-supervised training with title-body information has been explored in preliminary work in the context of cQA question similarity tasks before. The above mentioned master thesis (Wiedmeier, 2017) has found on one dataset that the training of representation learning models with title-body pairs can, in some limited cases, surpass supervised training. The above mentioned bachelor thesis (Vatter, 2019) then compared different training strategies, including the techniques mentioned above, on more datasets. Namely, the question similarity datasets also studied in this chapter. A particular focus lied on low-resource scenarios, i.e., when there are only a few labeled and unlabeled questions available. This work suggests that *pre-training* with title-body pairs can avoid catastrophic performance decreases in such settings.

Both works provide some preliminary indications that training models with title-body information can be effective. However, they are considerably limited in several important aspects relevant to this chapter. We highlight differences in §6.2.2.

6.2 Training Methods

We now formalize the different training methods that we study in this chapter. We describe them from the perspective of question similarity tasks, however, later in §6.4.2 we also study cQA answer selection.

Method	Duplicates	Answers	Bodies
Supervised in-domain training	✕	-	(✕)
Weak supervision with question-answer pairs (WS-QA)	-	✕	(✕)
Domain Transfer	✕*	-	(✕)
Duplicate question generation (DQG)	-	-	✕
Self-supervised training with title-body (SSTB)	-	-	✕

Table 6.1: The different training methods and the data they use. Models typically also use text from the bodies during training and evaluation, which we indicate with (✕). ✕* = duplicates from a sufficiently similar source domain.

To train models, we obtain a set of examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ in which each $x_n \in \mathcal{X}$ is an instance (i.e., two questions) and $y_n \in \{-1, +1\}$ is its corresponding binary label (duplicate or no-duplicate). Obtaining instances with positive labels $\mathcal{X}^+ = \{x_n^+ \in \mathcal{X} | y_n = 1\}$ is generally more difficult than obtaining $\mathcal{X}^-$ because instances with negative labels can be automatically constructed—e.g., by randomly sampling unrelated questions.

In §6.2.1, we outline three training methods that use different kinds of labeled instances. In §6.2.2, we present the methods that do not require annotations to obtain $\mathcal{X}^+$: self-supervised training with title-body pairs, and our proposed approach to duplicate question generation. Table 6.1 provides an overview of the different training methods in regard to the kind of data they use.

6.2.1 Training Methods with Labeled Data

6.2.1.1 Supervised In-Domain Training

Supervised in-domain training is the most common method to train question similarity models. It requires labeled question duplicates:

$$x_n^+ = (q_n, \tilde{q}_n) \tag{6.1}$$

Unrelated questions can be randomly sampled during training or prior to training. With this data, we can train representation learning models (e.g., Lei et al., 2016b) or pairwise classifiers (e.g., Uva et al., 2018). Most models combine the titles and bodies of the questions during training and evaluation, e.g., by concatenating them, which can improve performances (Lei et al., 2016b; Wu et al., 2018).

Duplicates are sometimes annotated by users of the cQA forum (e.g., in StackExchange), or they are obtained by manual annotation. Duplicate questions are therefore only available in large quantities for the most popular cQA forums (e.g., AskUbuntu.com) or when significant resources have been invested in manual annotation. If we assume perfect labels without noise this would reflect an optimal scenario.[2]

[2] Unfortunately, we cannot assume perfect labels when relying on community data, see (Hoogeveen et al., 2016).

6.2.1.2 Weak Supervision With Question-Answer Pairs (WS-QA)

WS-QA is an alternative to supervised training for larger platforms without duplicate annotations (Qiu and Huang, 2015; Wang et al., 2017a). Instead of question duplicates and randomly sampled unrelated questions, WS-QA trains models with questions q_n and their answers a_n, and therefore:

$$x_n^+ = (q_n, a_n) \tag{6.2}$$

In this chapter, we only consider annotated pairs of questions and "accepted answers"—i.e., good answers that were approved by the community. It may also be possible to consider other signals that indicate answer quality, e.g., up-votes, user authority, etc., thereby increasing the training data size. We leave this for future work.

Instances in X^- can be obtained by randomly sampling unrelated answers. An advantage of this method is that there typically exist more labeled question-answer pairs than duplicate questions. For instance, Yahoo! answers has accepted answers but it does not contain labeled duplicates. The underlying idea is that accepted answers are strongly related to their question, thereby allowing the model to learn appropriate question representations.

6.2.1.3 Domain Transfer

Domain transfer performs supervised training in a source domain and applies the trained model to a different target domain in which no labeled duplicate questions exist. Shah et al. (2018) combine this method with *adversarial training* to learn domain-invariant question representations prior to transfer. They show that adversarial training can considerably improve upon direct transfer, but their method requires sufficiently similar source and target domains. For instance, they were not able to successfully transfer their BiLSTM models between technical and other non-technical domains.

6.2.2 Training Methods with Unlabeled Data

As can be seen in Table 6.1, a central shortcoming of the previous methods is that they require labeled question duplicates, question-answer pairs, or similar source and target domains for transfer. We could alternatively use unsupervised training with an encoder-decoder framework, but this imposes important limitations on the network architecture, e.g., questions can only be encoded independently (no cross-attention).

DQG and SSTB do not suffer from these drawbacks. They do not require labeled data and they do not impose architectural limitations.

6.2.2.1 Duplicate Question Generation (DQG)

We propose generating a new question title from a question body, which we then consider as a duplicate to the question's original title. The idea is that both title and body describe the same question but are only partly redundant—e.g., in Figure 6.1 we see that the body contains additional information. If we generate a new title

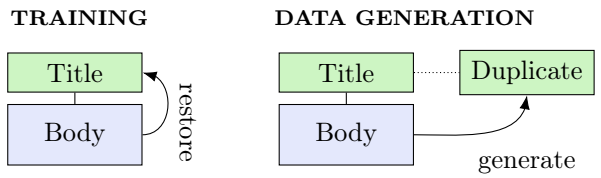

Figure 6.2: Our proposed approach to duplicate question generation. During training of our QG model, we restore the original question title from its body. During data generation we consider the generated title as a duplicate to the question's original title.

from the body, it is likely to be only partially redundant with the original title—thus having similar attributes to duplicate questions that rarely overlap entirely.

Our overall approach is depicted in Figure 6.2. We first train a question generation model QG to maximize $P(\text{title}(q_n)|\text{body}(q_n))$. This is similar to news headline generation and abstractive summarization (Rush et al., 2015; Chopra et al., 2016) because QG learns to identify the most relevant parts of the text in the body that best characterize the question. Importantly, in our case, restoring the exact title from a question body is usually not possible because titles and bodies often contain complementary information—i.e., the title is not a summary of the body. However, not being able to fully restore the original title is a desired property in our case as we can then consider $\text{QG}(\text{body}(q_n))$ a duplicate to $\text{title}(q_n)$. A similar approach has been used by (Lei et al., 2016a) to *pre-train* their question encoder, however, not to generate new training examples.

We obtain positive labeled instances:

$$x_n^+ = (\text{title}(q_n), \text{QG}(\text{body}(q_n))) \tag{6.3}$$

Negative instances in X^- are titles of randomly sampled unrelated questions.

Because DQG requires **no labeled data**, we can apply this method to all cQA forums that offer a reasonable number of unlabeled title-body pairs (the smallest number of questions we tried for training QG models is 23k, see Section 6.4). Later in Section 6.5, we further study whether we can transfer QG across distant domains, which makes our method applicable to more extreme cases. An important advantage of DQG is that we can make use of *all questions* (after some basic filtering that we describe in §6.3.1.3), which is often an order of magnitude more training data than annotated duplicates. We can then use any standard sequence-to-sequence model for QG.

6.2.2.2 Self-Supervised Training With Title-Body Pairs (SSTB)

SSTB trains models to predict whether a given title and body are related, i.e., whether they belong to the same question. Therefore:

$$x^+ = (\text{title}(q_n), \text{body}(q_n)) \tag{6.4}$$

Instances in X^- are bodies of randomly sampled unrelated questions.

This method considerably simplifies the dataset creation because it requires no separate QG model. However, the question similarity model needs to be able to handle texts of considerably different lengths during training. For instance, bodies in SuperUser.com have an average length of 125 words, which might not be suitable for text matching models that were designed to compare two sentences. We do not apply filtering strategies beyond ensuring that the questions are not down-voted and contain more than three words in the title and body each.

Notably, similar techniques have been explored for other tasks. For the task of ad-hoc re-ranking, MacAvaney et al. (2019) use headline-passage pairs of news articles and title/passage pairs from Wikipedia for model training. Furthermore, the inverse cloze task (Chang et al., 2020; Lee et al., 2019b) can be used to train retrieval models by predicting whether a sentence-passage pair belong together or not—i.e., whether the sentence was taken out of this passage or comes from another passage.

As described in §6.1.2, a variant of SSTB was studied in a Master thesis (Wiedmeier, 2017) and later in a Bachelor thesis (Vatter, 2019), both under close supervision by Andreas Rücklé. They provide preliminary indications that training models with title-body information can be effective. Our work goes considerably beyond those in that (a) our negative instances contain question bodies instead of titles, allowing for a better comparison against WS-QA, which also deals with negative instances that are long texts. (b) we study the impact of adding more unlabeled data instead of reducing the labeled and unlabeled data, which is arguably the most realistic setting in cQA; (c) we compare SSTB to DQG that uses the same data but reduces the text lengths with QG; (d) we investigate whether SSTB is also suited to train answer selection models; and (e) we also study the effectiveness of SSTB in combination with other models such as COALA (Chapter 5.2) and BERT (Devlin et al., 2019).

6.3 Question Similarity Experiments

In this section, we investigate the effectiveness of the different training methods for *question similarity tasks*. Later in Section 6.4, we then extend this to other setups, including answer selection.

6.3.1 Experimental Setup

We use the models, data, and evaluation procedures of previous literature to obtain comparable results, and we rely on their official implementations and hyperparameters. We provide an overview of the datasets in Table 6.2. This also illustrates that the datasets considerably differ in the amount of data that is available for the different training methods.

The evaluation setup is the same for all datasets: Given a query question $q \in Q$, a set of n candidate questions $Q' = \{q'_0, q'_1, \ldots, q'_n\} \subset Q$, and a function $rel : Q, Q \rightarrow \{0, 1\}$ indicating whether two questions are similar or not. Our goal is to find a model that produces an optimal ranking of the candidates in Q' with respect to rel.

Even though not all training methods use question bodies during training—e.g.,

Dataset	Train	Dev	Test	|Q|	|A|
AskUbuntu-Lei	12 584	189	186	288k	84k
AskUbuntu	9106	1000	1000	288k	84k
SuperUser	9106	1000	1000	377k	142k
Apple	-	1000	1000	89k	29k
Android	-	1000	1000	47k	14k

Table 6.2: The dataset statistics for the question similarity datasets. Numbers for Train/Dev/Test refer to the number of questions with duplicates. |Q| denotes the number of unlabeled questions, and |A| denotes the number of questions with accepted answers. AskUbuntu-Lei ist the dataset proposed in (Lei et al., 2016b), and the remaining four datasets are the StackExchange datasets in (Shah et al., 2018)

DQG only uses titles—we evaluate all methods with question titles and bodies. The reason is that it has been shown previously that including bodies in the experimental setup can lead to improved performances (Lei et al., 2016b). In our initial experiments, we found that the performance scores are indeed positively impacted by having access to bodies during evaluation (and less so when having access to bodies during training).

Next, we describe the two setups that we use in our experiments, and we detail the construction of SSTB and DQG training sets.

6.3.1.1 Datasets and Models

AskUbuntu-Lei. We replicate the setup of Lei et al. (2016b) using RCNN to learn dense vector representations of questions, which we compare with cosine similarity for scoring. We previously used RCNN in Section 4.2, which also includes more details on the neural network architecture. Besides supervised training, we can train RCNN with unsupervised training using an encoder-decoder architecture.[3] We report precision@5 (P@5), i.e., how many of the top-5 ranked questions are actual duplicates. We find that this is more stable than P@1 (accuracy) or MRR/MAP because the development and test splits only contain a small number of questions, which have been extended with manual annotations (a query question has an average of 5.7 similar questions, thus being suited for measuring P@5). Following (Lei et al., 2016a), we report the average performance over five runs.

Android, Apple, AskUbuntu, and SuperUser. We replicate the setup of Shah et al. (2018), using BiLSTM to learn question representations. The results are obtained for a single run with fixed random seed. This setup also includes adversarial domain adaptation. The data is from the AskUbuntu, SuperUser, Android, and Apple sites of StackExchange, and different to AskUbuntu-Lei, each query question has only one similar question (a community-labeled duplicate). We measure AUC(0.05), which is the area under curve with a threshold for false positives. Shah

[3] In contrast to (Wiedmeier, 2017), we do not limit the number of questions for this method to 2000.

et al. (2018) argue that this is more stable when there are many unrelated questions. In contrast to AskUbuntu-Lei, they construct the evaluation sets by randomly sampling unrelated questions for each duplicate pair. We note that this does not reflect a realistic scenario as we would typically use a search engine to collect potentially similar questions (e.g., using BM25; see our cQA system design in Section 2.2). We include this setup in our experiments, because it (a) covers multiple domains, and (b) allows us to directly compare against the adversarial domain adaptation.

6.3.1.2 SSTB and WS-QA Setup

To train models with SSTB and WS-QA, we use questions and answers from publicly available data dumps[4] of the StackExchange platforms. We construct new training sets as described in §6.2.2. For SSTB, we replace every annotated duplicate $(q_n, \tilde{q}_n)$ from the original training split with $(\text{title}(q_n), \text{body}(q_n))$. We then randomly sample bodies of unrelated questions as negative instances. For WS-QA, we follow a similar procedure, however, some questions of the training split do not contain accepted answers. We thus add additional (randomly selected) question-answer pairs to the training splits until the number of training instances matches that of the supervised training with annotated duplicates. We randomly sample unrelated answers as negative instances.

Notably, the number of questions and answers is much larger than the number of annotated duplicate questions in the training sets, see Table 6.2. We can therefore add more instances to the training splits with these methods. However, if not otherwise noted, we use the same number of training instances as in the original training splits with duplicates to obtain performance scores that are directly comparable to supervised training. We always ensure to not add instances from the development or test splits to our extended datasets.

Our setup consists of the following important changes against the preliminary study of (Vatter, 2019). (1) We sample unrelated bodies instead of unrelated titles. This allows for a better comparison against WS-QA (both then sample unrelated long texts). (2) We re-use the query questions from the training splits of supervised training for WS-QA instead of choosing random new questions. Thus, we do not introduce any new information to the models, which provides us with a more fair setup. (3) We obtain SSTB data directly from the StackExchange data dumps instead of extracting it from the pre-existing training sets. This means that when we add more questions that are not in the original training sets, all use the same pre-processing.

However, we also mention that both projects were partly built upon the same source code, i.e., an initial version of our question generation work presented in this chapter—including StackExchange data processing and an adaptation of Shah et al. (2018)'s experimental framework. Our experiments re-used minor parts from Vatter (2019)'s adaptation of this source code (developed according to the detailed task description provided by the thesis author) related to *only data preparation*, but did not directly use any of their data, trained models, results, etc. (we implement a

[4] https://archive.org/download/stackexchange; last accessed 15 Jan. 2021.

different setup, see above).

6.3.1.3 DQG Setup

We use the same StackExchange data as in SSTB to train our question generation models. We filter the questions to ensure that the bodies contain multiple sentences. Furthermore, if a body contains multiple paragraphs, we only keep the one with the highest similarity to the title. Technical details of the filtering approach are given in Appendix D. We discarded less than 10% of the questions on average.

We train MQAN (Multi-task Question Answering Network), which is a very general network architecture to solve a wide variety of tasks, and has been proposed as part of the Natural Language Decathlon (McCann et al., 2018). We adapt its official implementation and add the QG task.[5] MQAN first encodes the input with LSTMs and applies different attention mechanisms, including multi-head self-attention. MQAN also includes pointer-generator networks (See et al., 2017), which allow it to copy tokens from the input text depending on the attention distribution of an earlier layer.

We perform the same experiments with a transformer sequence-to-sequence model (Vaswani et al., 2017), which has been originally proposed in the context of machine translation. One of the most important differences to MQAN is that transformer does not include a copy mechanism, i.e., it generates questions by sampling tokens from its output vocabulary as opposed to also sampling from the input text. We use the official implementation from the Tensor2Tensor library (Vaswani et al., 2018) and use the same encoder-decoder approach as in machine translation. Instead of translating an input sentence to a target language, we generate a question from a paragraph of the body. We use the "transformer_small" hyperparameter configuration because of the smaller quantity of training examples as compared to typical machine translation setups.

We use all available questions (after filtering) of a cQA forum to train the question generation model. We perform early stopping using BLEU scores to avoid overfitting (5k title-body pairs are reserved for validation). To generate duplicate questions, we then apply the trained model on all questions from the same cQA forum. We do not use a separate heldout set for the generation of duplicate questions because this would considerably limit both the question generation training data and the number of generated duplicates. We did not observe negative effects from using this procedure and our QG models tend to not overfit on the training data (i.e., they do not learn to memorize the exact question titles).

6.3.2 Experimental Results

We present the experimental results in Table 6.3. For domain transfer, we include the best scores reported in Shah et al. (2018). This reflects an *optimal transfer setup* from a very similar source domain with sufficient numbers of annotated duplicate

[5] https://github.com/salesforce/decaNLP; last accessed 12 Dec. 2020.

	AU-Lei	Android	Apple	AU	SU	Σ
	P@5 (dev / test) for RCNN	AUC(0.05) for BiLSTM				
Trained on 1x data (all methods use the same number of instances as in supervised training)						
Supervised (in-domain)	**48.0** / 45.0	-	-	**0.848**	**0.944**	-
◇Unsupervised	42.6 / 42.0	-	-	-	-	-
Direct Transfer	-	0.770	0.828	0.730	0.908	0.809
Adversarial Transfer	-	0.790	0.861	0.796	0.911	0.840
◇WS-QA	47.2 / 45.3	0.780	**0.894**	0.790	0.919	0.846
DQGTransformer	47.2 / 44.9	0.723	0.809	0.799	0.917	0.812
DQGMQAN	46.4 / 44.8	0.793	0.870	0.801	0.921	0.846
◇SSTB	46.4 / **45.4**	**0.811**	0.866	0.804	0.913	**0.849**
Trained on all available data						
◇Unsupervised	43.0 / 41.8	-	-	-	-	-
◇WS-QA	47.3 / 44.2	0.814	0.901	0.828	0.951	0.874
DQGTransformer	46.4 / 44.7	0.783	0.876	0.836	0.942	0.859
DQGMQAN	**47.4** / 44.3	0.833	0.911	0.855	0.944	0.886
◇SSTB	47.3 / **45.3**	0.852	0.910	**0.871**	**0.952**	0.896
DQGMQAN + SSTB	46.4 / 44.0	**0.863**	**0.916**	0.866	0.946	**0.898**

Table 6.3: Results of the models with different training strategies. AU refers to AskUbuntu, and SU refers to SuperUser. Direct transfer and Adversarial transfer performance scores are the best results reported in (Shah et al., 2018)—i.e., an optimal transfer scenario from a very similar domain. Android and Apple datasets do not contain labeled duplicates for supervised in-domain training. Σ denotes the average over performance scores. ◇ denotes methods that we newly evaluate, but that have been studied in related settings on these datasets before with important differences as described in §6.2.2.2 and §6.3.1.2 (different negative instances, less data, other data collection procedures, compared to different approaches etc.)

questions. For instance, the best scores for AskUbuntu were achieved by transferring from SuperUser.

Supervised training. As we expect, supervised in-domain training with labeled duplicates achieves better scores compared to other training methods when all use the same number of training instances. An exception is on AskUbuntu-Lei where DQGMQAN, SSTB, and WS-QA can achieve results that are on the same level on the test split or marginally worse on the development split. All three methods outperform direct transfer from a similar source domain as well as the encoder-decoder approach on AskUbuntu-Lei. This is in line with the preliminary results of (Wiedmeier, 2017; Vatter, 2019) obtained for similar methods on these datasets.

MQAN vs. Transformer for DQG. For our newly proposed approach to DQG, we observe that the QG model can have a large impact on the model performance. Most notably, transformer performs considerably below MQAN on the smaller cQA forums (Android and Apple). One reason for the better performance scores of MQAN lies in its copy mechanism, which allows the model to sample output tokens directly from the question body. For instance, if there are only a few questions

in the QG training set, it can be difficult to adequately learn the idiosyncrasies of our domains (including the domain-specific vocabulary such as version numbers in programming forums). However, models with a copy mechanism can learn to extract certain parts of the text that seem relevant, without having observed the specific vocabulary before. We provide further insights on the performance of MQAN compared to transformer in §6.4.1.

DQGmqan, SSTB, and WS-QA. We do not observe large differences between DQGmqan, SSTB, and WS-QA, which shows that (1) the models successfully learn from different text lengths (title-body and question-answer in comparison to title-generated duplicate); (2) the information we extract in DQGmqan is suitable for model training (we show examples later in Section 6.5). The good results of SSTB might suggest that question generation as separate step is not required. However, we argue that it can be important in a number of scenarios, e.g., for training sentence-level models that would otherwise not be able to handle long texts or for efficiency reasons (shorter sequence length).

Using all available data. One of the most important advantages of the aforementioned methods is that they can use larger quantities of training data. This greatly improves the model performances for BiLSTM, where we observe average improvements of up to 4.7pp (for SSTB). In many cases these methods now perform better than supervised training. We observe smaller improvements for WS-QA (2.8pp on average), likely because it has access to fewer training instances.[6] The performance scores of RCNN on AskUbuntu-Lei are mostly unchanged with minor improvements on the development splits. One potential reason can be that the performances were already close to those of supervised training with the same number of training instances.

Figure 6.3 shows the performance scores of BiLSTM for the four StackExchange datasets in relation to the available training data with SSTB. We see that the model performance consistently improves with increased training data sizes (we observe similar trends for DQG and WS-QA).

We also explore a combination of SSTB and DQGmqan by concatenating their respective training sets. We find that this improves the results for smaller cQA platforms with fewer questions, e.g., the performances on Android and Apple improve by 0.6–1.1pp compared to SSTB. Even though the combination of SSTB and DQGmqan does not introduce new information because both use the same unlabeled data, complementing SSTB with generated questions can provide additional variation during training.

In summary, our results show that even with access to sufficient numbers of labeled duplicates, the best method is not always supervised training. When we use larger

[6] One potential method for further improving WS-QA could be to increase the number of question-answer pairs in the training set. For instance, one could include answers that have not been accepted by the forum community, but are of sufficient quality. Designing appropriate heuristics is outside the scope of this chapter.

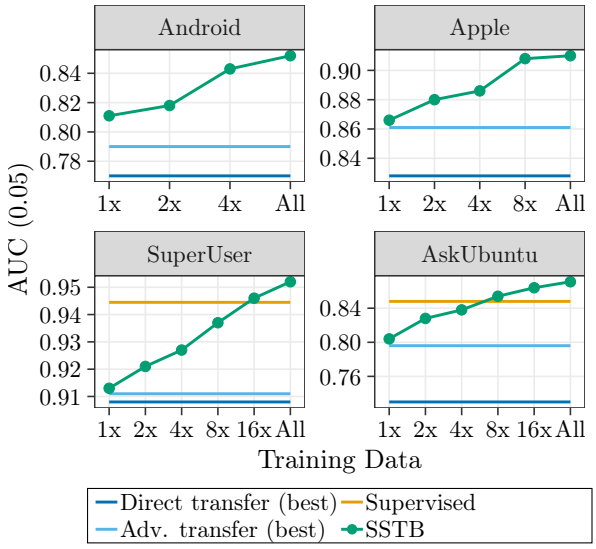

Figure 6.3: Performances of BiLSTM as a function of the available training data. *2x* means that there are twice as many unlabeled questions available to SSTB than there are annotated duplicate questions in the original training set (1x = 9106).

numbers of title-body pairs, both DQG (with a good QG model) and SSTB can achieve better performances.

6.4 Further Application Scenarios

We study three additional scenarios with significant practical relevance: (1) We investigate whether QG models generalize well to unseen domains. (2) We train answer selection models without labeled question-answer pairs. (3) We examine how well BERT models perform when fine-tuned with SSTB.

6.4.1 Cross-Domain QG

In our previous experiments, we assumed that there exist enough unlabeled questions to train our question generation models (at least 47k; see Table 6.2). We now investigate whether we can apply DQG to even more challenging scenarios where it might be impossible to train suitable QG models. We achieve this by transferring QG models across domains.

We replicate the domain transfer setup of Shah et al. (2018), in which they originally transfer question similarity models from a source domain to a target domain. For DQG we instead train the QG model on the source domain and generate duplicates for the target domain. We then use the generated duplicates to train our question similarity model. To provide a direct comparison against adversarial domain transfer, we always train question similarity models with the same number of 9106 duplicates (either from the source domain, or generated with our QG model). Better performance scores can likely be achieved by further increasing the number of training examples.

Table 6.4 shows the results for the transfer from SuperUser and AskUbuntu to the other forums. Our QG models generalize well to similar forums, where they achieve performances comparable to those obtained using in-domain QG. We observe positive as well as negative differences (Δ) for MQAN and mostly positive differences for transformer. The improvements of transformer are likely due to the larger quantities of title-body pairs for the source domains. As we have seen previously in Section 5.4, DQG with transformer performs generally better on larger cQA forums. Most importantly, question similarity models trained with DQG$_{\text{MQAN}}$ still achieve better performances than adversarial domain transfer.

To study an even more extreme case, we also transfer from StackExchange Travel and StackExchange Academia forums, which only contain 30k and 23k unlabeled questions, respectively. Both forums do not contain enough annotated duplicates to train models for direct transfer or adversarial transfer. Importantly, both source domains are much more distant to the target domains as compared to the previous setup. Dealing with such distant domains could, for instance, be more realistic for other languages where fewer cQA forums exist. In Table 6.4, we see that the performances of DQG decrease in all cases. We observe the largest decreases for DQG$_{\text{Transformer}}$ (-0.061 to -0.233) but, most notable, only mild decreases (-0.004 to -0.041) for DQG$_{\text{MQAN}}$. On the Apple dataset, our model still performs better than adversarial domain transfer from a similar source domain.

This clearly demonstrates the effectiveness of DQG$_{\text{MQAN}}$ and its applicability to very challenging scenarios. The good performances are likely due to MQAN's copy mechanism, which is less affected by domain shift (it can copy directly from the input). We provide examples later in Section 6.5.

We note that more recent work has successfully transferred transformer-based QG models across distant domains. Ma et al. (2020) train a transformer model on 2 million question-answer pairs from Yahoo! Answers and different StackExchange forums and apply this model to generate synthetic queries for specialized domains such as BioASQ (Tsatsaronis et al., 2015). Liang et al. (2020) fine-tune a BART model (Lewis et al., 2020) on MSMARCO (Nguyen et al., 2016b) for question generation and generate synthetic queries for different retrieval tasks such as ReQA (Ahmad et al., 2019). Both works train transformer-based QG models with large amounts of mixed-domain data and the latter leverages a pre-trained model. This likely diminishes possible domain transfer losses, i.e., both works successfully train good retrieval models with these techniques. Future work could explore whether this also holds for duplicate question generation.

6.4.2 Answer Selection

Training models without labeled data is also highly desirable for the task of answer selection, i.e., being able to train models for cQA forums that do not contain annotated question-answer pairs. We train a siamese BiLSTM and our COALA model of Chapter 5 with SSTB on the five StackExchange datasets LAS-Apple, Aviation, Academia, Cooking, and Travel (see Section 5.3). We evaluate both by how well they re-rank a list of candidate answers in regard to a question.

Domains		Domain Transfer		Duplicate Question Generation			
Source	**Target**	**Direct**	**Adv.**	**Transformer**	**Δ**	**MQAN**	**Δ**
AskUbuntu	Android	0.692	0.790	0.762	+0.039	**0.797**	+0.004
	Apple	0.828	0.855	0.821	+0.012	**0.861**	−0.009
	SuperUser	0.908	0.911	0.913	−0.004	**0.916**	−0.005
SuperUser	Android	0.770	0.790	0.755	+0.032	**0.794**	+0.001
	Apple	0.828	**0.861**	0.833	+0.024	**0.861**	−0.009
	AskUbuntu	0.730	0.796	0.797	−0.002	**0.809**	+0.008
Distant Domains							
Travel	Android	-	-	0.550	−0.173	0.789	−0.004
	Apple	-	-	0.624	−0.185	0.864	−0.006
	SuperUser	-	-	0.856	−0.061	0.914	−0.007
	AskUbuntu	-	-	0.664	−0.135	0.787	−0.014
Academia	Android	-	-	0.530	−0.193	0.776	−0.017
	Apple	-	-	0.576	−0.233	0.854	−0.016
	SuperUser	-	-	0.840	−0.077	0.912	−0.009
	AskUbuntu	-	-	0.672	−0.127	0.760	−0.041

Table 6.4: The domain transfer performances measured in AUC(0.05). Δ denotes the difference to the performances that are achieved with the in-domain DQG (same number of generated duplicate questions, i.e., 1x training data as in Section 6.3). Bold values indicate the best results in a specific transfer scenario.

	Supervised	**SSTB** (1x)	**SSTB** (all)
BiLSTM	35.3	37.5	42.5
COALA	44.7	45.2	44.5

Table 6.5: Answer selection accuracies averaged over our five StackExchange datasets presented in Section 5.3.1

.

The results are given in Table 6.5 where we report the accuracy, averaged over the five datasets. Surprisingly, we do not observe large differences between supervised training and SSTB for both models when they use the same number of positive training instances (ranging from 2.8k to 5.8k). Thus, using title-body information instead of question-answer pairs to train models without direct answer supervision is feasible and effective. Further, when we use all available title-body pairs, the BiLSTM model substantially improves by 5pp, which is only slightly below the performance of COALA. This further illustrates that BiLSTM requires more training data to learn suitable representations for the long texts.

Our results demonstrate that SSTB can be more broadly applied to other cQA tasks beyond question similarity. Indeed, we build upon this important finding later in Chapter 7.

	AskUbuntu-Lei	Android	Apple	AskUbuntu	SuperUser	AS
	P@5; dev / test	AUC(0.05)				Accuracy
Supervised	**54.0** / **52.3**	-	-	0.862	0.954	56.8
SSTB (1x)	47.8 / 47.2	0.857	0.908	0.841	0.932	55.5
SSTB (8x)	50.4 / 49.6	**0.896**	**0.933**	**0.897**	**0.971**	**59.7**

Table 6.6: Results of fine-tuned BERT models with different training strategies. *AS* refers to the answer selection accuracies, averaged over our five StackExchange datasets that we have presented in Section 5.3.1.

6.4.3 BERT Fine-Tuning

Large pre-trained transformers have recently achieved considerable improvements across a wide range of NLP tasks. We investigate whether we can successfully use SSTB with such models and fine-tune BERT (Devlin et al., 2019) in the same way as in our previous experiments. We use the HuggingFace framework (Wolf et al., 2020) for our BERT experiments.

For the question similarity setup of (Shah et al., 2018) (Android, Apple, AskUbuntu, SuperUser), BERT replaces the BiLSTM encoder. We learn question representations by averaging over the BERT output representations of all tokens in a question. The rest of the implementation is the same as for BiLSTM (e.g., loss calculation). We train the models until they do not improve for at least 20 epochs, and we restore the weights of the epoch that obtained the best development score.

In the answer selection experiments, BERT replaces COALA (as a pointwise ranking model). We add the AskUbuntu-Lei dataset to our experimental framework from Chapter 5 because the source code of RCNN and AskUbuntu-Lei relies on Theano—a discontinued project that does not support the seamless integration of BERT models (as of May 2019). The output prediction is then used as a ranking score. We train the models for 10 epochs and restore the weights of the epoch that obtained the best development score. We fine-tune BERT-base (uncased) with supervised training, SSTB (1x), and SSTB (8x). We do not include SSTB (all) for computational reasons.

In Table 6.6, we observe similar trends as before but with overall better results. When increasing the number of training examples, the model performances consistently improve in all settings. This underlines, even more, the broad applicability and the wide range of use-cases for SSTB. Future work could achieve further improvements by combining SSTB and supervised training for BERT models, either through data concatenation or intermediate pre-training (Phang et al., 2018).

6.5 Analysis

6.5.1 Lexical Similarity

We investigate the differences between the training methods by measuring the lexical similarity of texts from positive training instances (e.g., title-body, question-

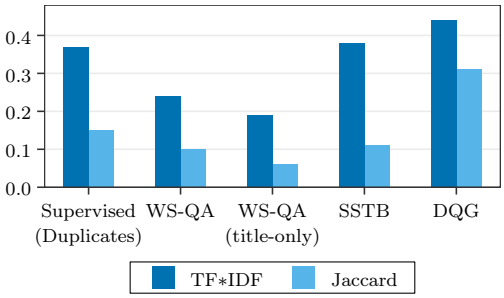

Figure 6.4: Average overlap of positive training instances (all words were stemmed and lowercased).

Title: 14.10, 15.04 - HDMI audio not working on Dell Vostro 3750 - nVidia card not detected by aplay -l
DQG: ALSA not detected in nVidia

Title: Installing ubuntu 12.04.02 in uefi mode
DQG: Ubuntu 16.04 LTS boot loader not working

Title: Grub2 not updating
DQG: How to fix Grub2 error

Figure 6.5: Three random examples of question titles and DQGMQAN output.

answer).[7] Figure 6.4 shows the Jaccard coefficient and the TF∗IDF score averaged over all instances in the four StackExchange datasets of Section 5.4.

We observe that the similarity of positive instances in SSTB is very close to the similarity of annotated duplicate questions in supervised training. The lexical similarity of positive instances in DQG is higher because the generated titles are shorter compared to the bodies and they only contain relevant content (e.g., no conversational phrases). We also calculated the BLEU scores for MQAN and found that they are not very high—between 13.3–18.9 BLEU depending on the dataset—which indicates that the generated texts are still sufficiently different from the question's original title. The similarities show that both SSTB and DQG use training data with sufficiently similar, but not fully redundant texts.

Interestingly, the lexical similarity for question-answer pairs is lower, especially when considering title-answer pairs—as it is the case in the answer selection experiments. This may contribute to the better performance scores that we achieve with SSTB for answer selection when using BiLSTM. Because the overlap of title-body pairs is higher, the siamese network receives a stronger training signal for the overlapping (important) text segments.

6.5.2 Qualitative Analysis

To better understand the results of DQGMQAN, we manually inspected a random sample of 200 generated questions and title-body pairs from multiple datasets. Three titles and generated duplicates from AskUbuntu are shown in Figure 6.5 and five additional examples, including a comparison of MQAN and transformer together with the source bodies are given in Figure 6.6 and Figure 6.7.

We found that most of the generated duplicates are sensible, but around 20% of the texts suffer from one of the following problems:

1. Some generated questions are off-topic because they contain information that was generated from a body with minimal overlap to the title (see example 4 in Figure 6.7).
2. Some generated questions include wrong version numbers or wrong names (see example 5 in Figure 6.7, or the second example in Figure 6.5), which is more often the case with transformer and less often with MQAN because it copies these rare words directly from the body.

We found that many of the generated titles contain complementary information to the original title, as can be seen in Figure 6.5 (e.g., "ALSA", and "boot loader").

We also inspected the training data of the most difficult DQG domain transfer case (from StackExchange Academia) to investigate the reasons for the surprisingly good performances when transferring the MQAN question generation model. Most notably, we find that the model often falls back to copying important phrases from the body and sometimes generates additional words from the source domain. We note that this is not the case with the transformer, which often generates unrelated text from the source domain (see the transformer examples in Figure 6.7).

[7] Prior work of Vatter (2019) measured the n-gram overlap or n-gram coverage for a subset of these training methods, but they report contradictory results in their analysis and appendix. We, thus, provide no further comparison here.

Example 1

TITLE

how to get beep working?

BODY (PARAGRAPH)

I have a laptop, i installed the "beep" package. I turned every sound to full, and i: but i can't hear any "beeping" sound. What am I missing? I just need to run the "beep" when a script is finished. Thank you for any links/howtos!

IN-DOMAIN QG MODELS

MQAN: How to fix beep package?

Transformer: How to remove " beep " from my laptop?

Example 3. This contains a particularly short description with not much overlap to the question. Such examples could be removed with additional preprocessing / filtering.

TITLE

How to echo contents of file in a DOS/Windows command prompt?

BODY (PARAGRAPH)

Like in Unix' cat filename ... Can this be done?

IN-DOMAIN QG MODELS

MQAN: How to make cat filename in Unix' cat filename ?

Transformer: What is the difference between 'cat' and 'cat' in Unix?

Example 2

TITLE

13" MacBook Pro with Win 7 and External VGA gets 640x480

BODY (PARAGRAPH)

I have a brand new 13" MacBook Pro - 2.26 GHz and the NVIDIA 9400M Video card. I installed Windows 7 (final) in boot camp and booted up to Windows 7. Installed all the drivers from the Apple disk and it was working great. Then I attached the external VGA adapter (from apple) to connect to a projector and it dropped down at 640x480 resolution. No matter what I did it would n't let me change to a higher resolution if the external VGA was connected. Once it disconnects then it goes back to the normal resolution. If I am booted into Snow Leopard it works fine. I tried updating the NVIDIA drivers and it behaved exactly the same. Ultimately I want to get 1024x768 or better resolution when connected to an external display. If it isn't fixable then I am curious if anyone else has seen this, if it is a known issue, and who to contact for support (Apple, Microsoft or NVIDIA?) Update: Just attaching the Mini-DVI to VGA adapter kicks it into 640x480, no projector is required. I tried forcing the display driver from Generic PnP Monitor to one that supported 1024x768 and that didn't work either.

IN-DOMAIN QG MODELS

MQAN: 1024x768 or better resolution when connected to an external display

Transformer: How can I get an external display to work?

Figure 6.6: Examples of question titles, question bodies (which are the source texts for question generation) and the generated questions. The questions generated with MQAN more closely retain the meaning of the body paragraph, but transformer questions also contain the relevant keywords (except for the transfer cases). Original questions: https://superuser.com/questions/232672 (left, top); https://superuser.com/questions/256651 (left, bottom); https://superuser.com/questions/51004 (right).

Example 4. The duplicates were generated based on a body that does not have much overlap with the question.

TITLE

How can I remove an autostart service that is not listed in gnome-session-properties?

BODY (PARAGRAPH)

Today I upgraded from raring to saucy.
This brought the Ubuntu One icon back to my indicator applet, which I had disabled.
So I removed the line NoDiplay=true from /etc/xdg/autostart/ubuntuone-launch.desktop.
But still Ubuntu One doesn't show up in gnome-session-properties and I can't disable it. What gives?

IN-DOMAIN QG MODELS

MQAN: Ubuntu One doesn't show up in gnome-session-properties

Transformer: Why doesn't Ubuntu One sync with an indicator?

Example 5. The question generated by the (in-domain) transformer model is suitable, but it does not contain the correct product name of the printer ("0b" instead of "LBP2900b"). However, even the MQAN model that was trained on StackExchange Travel is able to correctly copy all necessary information from the input. The transformer trained on StackExchange Travel fails with generic (and grammatical) text from the travel domain.

TITLE

How to install Canon LBP2900b drivers?

BODY (PARAGRAPH)

I am trying very hard to install Canon LBP2900b Printer in Ubuntu 13.10. I have searched and googled a lot for the solution over fortnight but none of the site / link gave me the simple solution for me. How can accomplish my goal?

IN-DOMAIN QG MODELS

MQAN: How to install Canon LBP2900b Printer in Ubuntu 13.10?

Transformer: How to Install Canon 0b Printer on Ubuntu 13.10?

DOMAIN TRANSFER QG MODELS
(from SE Travel)

MQAN: How to install to install Canon LBP2900b in Ubuntu 13.10?

Transformer: How can I find my boat in Hokkaido?

Figure 6.7: Further examples of question titles, question bodies (which are the source texts for question generation) and the generated questions, reflecting common mistakes. Original questions: https://askubuntu.com/questions/381563 (left); https://askubuntu.com/questions/383695 (right).

6.6 Chapter Summary

In this chapter, we studied how to train cQA models without labeled data. The two methods we have investigated, duplicate question generation (DQG) and self-supervised training with title-body pairs (SSTB), require only title-body information of unlabeled questions and can thus leverage more data than supervised training. We can apply them to any cQA forum where questions consist of titles and bodies.

Our experiments revealed several interesting findings:

1. SSTB achieves slightly better results than DQG in our experiments, but it trains models on texts of different lengths. If a model is designed to compare individual sentences, DQG might be a better choice. Like SSTB, DQG outperform adversarial domain transfer with the same number of training instances.
2. Using all available unlabeled data leads to considerably better performances on most datasets.
3. The question generation model can have a considerable impact on the downstream task performance. For instance, we observe considerably lower performances with transformer compared to MQAN on smaller StackExchange forums.
4. We can use SSTB to train models that are suitable for cQA answer selection without direct question-answer supervision. This indicates that both question similarity and answer selection tasks are inherently similar, both relying mostly on domain-specific textual similarity.
5. We can fine-tune BERT with SSTB on all our datasets, thus demonstrating that this training method is not specific to any particular model.
6. Surprisingly, we find that QG models transfer well even across distant domains, yielding suitable generated duplicates for model training. This is likely due to MQAN's copy mechanism, which is more robust against domain shift. This opens up a wide range of further use-cases. For instance, we could use this model to generate questions for answers, or potentially for text passages of other sources (e.g., Wikipedia).

We believe that our work can be widely extended and lays the foundation for several promising future directions. For instance, we could use QG models trained on different source domains to generate multiple different training instances for each question in the target domain. Furthermore, we could combine the unlabeled data from a large number of cQA forums to learn one model that transfers well to various target datasets. Indeed, we investigate this in our next chapter.

Chapter 7

Zero-Shot Transfer of cQA Text Matching Models

Parts of this chapter have been previously published as listed below. Verbatim quotes from this publication are included in this chapter.

Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych: 'MultiCQA: Zero-Shot Transfer of Self-Supervised Text Matching Models on a Massive Scale', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2020), pp. 2471–2486, November 2020.

My contributions: Model training and evaluation (including MultiCQA models), implementation of multi-task experiments, analysis with regard to domain similarity and data size, error analysis.

In the final chapter of this dissertation, we now go beyond the previous chapters by studying the zero-shot transfer capabilities of text matching models in the context of cQA. In contrast to our small data settings (Chapter 5) and the settings in which we only have access to unlabeled target domain data (Chapter 6), we now assume no prior knowledge about the target domain. Thereby, we will provide important insights on how to obtain universal and thus re-usable models, which is particularly crucial in cQA as there exist a large number of domains (i.e., forums).

In Chapter 4, we already studied a different perspective of universality, namely, how to transfer models across different languages. In Section 4.1, we proposed concatenated power mean word embeddings as universal cross-lingual sentence embeddings, which are re-usable across different NLP tasks and transfer well across languages. However, our sentence embeddings required training a separate classifier on top, which limits their usefulness in zero-shot setups. In Section 4.2, we then transferred monolingual question similarity models cross-lingually by machine translating German questions to English. These models were, nevertheless, specialized to specific programming and operating systems cQA forums.

Therefore, up to this chapter, we mostly followed the predominant approach in recent NLP literature: we trained models specialized to particular tasks and domains. Obtaining specialized models is also common in recent work on cQA domain transfer,

e.g., Poerner and Schütze (2019) and Shah et al. (2018) both adapt models with target domain data, thus, yielding new models for each target domain.

In this chapter, we challenge this paradigm and are—to the best of our knowledge—the first to study the zero-shot transfer capabilities of text matching models with a large number of source domains in our cQA settings. We thereby address our fourth and last research question:

RQ4: To which extent do text matching models generalize to unseen cQA tasks and domains?

We study this research question in two parts and on a large scale with 140 source domains.

In the **first part** of this chapter, we train 140 domain-specific text matching models and study their zero-shot transfer capabilities to nine benchmark[1] datasets of question similarity and answer selection tasks. By leveraging self-supervised training signals of question title-body pairs, we can analyze a large number of models specialized on diverse domains. We utilize the training method SSTB, which we have presented in Chapter 6, and train adapter modules (Rebuffi et al., 2017; Houlsby et al., 2019) within BERT (Devlin et al., 2019) for *each* of the 140 English StackExchange forums. Adapters considerably reduce storage requirements by training only a small number of additional parameters while keeping the pre-trained BERT weights fixed. Further, they have been shown to train up to 60% *faster* than full model fine-tuning (Rücklé et al., 2020a) and they are composable after training (Pfeiffer et al., 2021). In our extensive analysis, we show that our approach for zero-shot transfer is effective—on six benchmarks *all* 140 models outperform common IR baselines. More importantly, we revisit and analyze the traditional strategy of leveraging large datasets from intuitively similar domains to train models for zero-shot transfer. We establish that, in our setting, *neither* training data size *nor* domain similarity seem to be suitable choices for predicting the best models, stressing the need for more elaborate strategies to identify optimal training sets. This also demonstrates that considering a broad selection of source domains is crucial, which contrasts the standard practice of merely relying on the most similar or largest ones.

In the **second** part of this chapter, we study how to best *combine* multiple source domains with multi-task learning and AdapterFusion (Pfeiffer et al., 2021). Our analysis reveals that both approaches are not affected by catastrophic interference across training sets. In particular, our combination of *all* available source domains—despite the large data imbalance, see Figure 7.1—is the most effective and outperforms the respective best of 140 single-domain models on six out of nine benchmarks. Finally, we combine unlabeled with labeled data for training in a self-supervised and supervised fashion, which considerably improves the zero-shot transfer performances in 16 out of 18 cases. Our best model substantially outperforms the *in-domain* BERT and RoBERTa (Liu et al., 2019) models on six benchmarks, which demonstrates its versatility across tasks and domains. We also show that our model is an effective initialization for further in-domain fine-tuning, which achieves the best results

[1] We refer to our evaluation datasets as "benchmark" datasets here to clearly distinguish them from source datasets used for training and validation.

overall.

7.1 Background: Zero-Shot Transfer

Previous research has investigated the zero-shot transfer capabilities of models in other NLP areas. We briefly review the most relevant works and outline differences to our studies.

In **machine reading comprehension** (MRC; span-based answer extraction from given text passages), Talmor and Berant (2019) study the generalization capabilities of models across several datasets, e.g., SQuAD (Rajpurkar et al., 2018), NewsQA (Trischler et al., 2017), and DROP (Dua et al., 2019). In their zero-shot transfer experiments, they train individual models on five large datasets and transfer them to ten benchmark datasets. They find that the best zero-shot transfer performances are considerably lower than in-domain models (31.5% on average). They also train a single model on all five datasets jointly, which considerably closes this gap (to 3.7% on average). They provide a thorough analysis including important insights on the differences between various MRC benchmarks and on the generalization capabilities in regard to the amount of training data. They find that pre-training a single model on several large MRC datasets and then fine-tuning it on smaller ones achieves state-of-the-art performances. Based on this work, the MRQA 2019 workshop carried out a shared task for improving the zero-shot transfer capabilities of MRC models (Fisch et al., 2019). The most successful submissions leverage larger pre-trained models and ensembles thereof. For instance, the winning team (Li et al., 2019a) uses a combination of XLNet (Yang et al., 2019a) and ERNIE (Sun et al., 2020a), which they pre-train and fine-tune on multiple MRC datasets.

Our work differs considerably in that we deal with an entirely different category of tasks. Instead of identifying the span of a factoid answer within a given passage, we determine whether two questions or a question-answer pair express the same information (or the same information need).

Other researchers have studied zero-shot transfer in **retrieval settings** (some in parallel to our work). Recent approaches are based on embeddings for dense retrieval, specialized to question-answer data. Yang et al. (2020) train a sentence encoder using question-answer data from StackExchange, Yahoo! Answers, and Reddit, as well as natural language inference data (Bowman et al., 2015). They train a transformer model (USE-QA) in a multi-task setting, however, their exact process for collecting and balancing the training data remains unknown and statistics are not given in their publication. They report promising results for retrieving answer sentences and similar questions in two zero-shot scenarios. They achieve better performances compared to the IR baselines and the word embedding averaging model of Gillick et al. (2018) on their Quora and AskUbuntu datasets.

Based on that, authors from the same institution proposed ReQA, a task for end-to-end answer sentence retrieval (Ahmad et al., 2019). They identify three important aspects of this task: (1) Performing sentence-level retrieval, i.e., to retrieve only a single sentence that contains the answer to the question, potentially leveraging

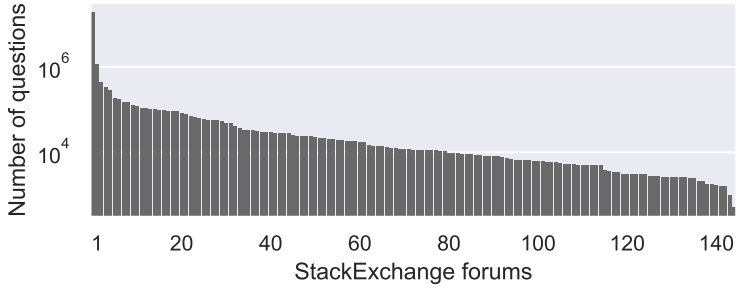

Figure 7.1: The number of questions in StackExchange forums (log scale) that can be used for self-supervised training.

information from the document context; (2) Being computationally efficient, e.g., by using dual encoders that learn question-answer representations suitable for end-to-end retrieval; (3) Obtaining general-purpose models in a sense that they should generalize well to unseen domains. For evaluation, they propose converting existing MRC datasets to ReQA, such that models are judged by how well they retrieve the sentence that contains the correct answer span from a large corpus (e.g., Wikipedia). ReQA has recently been extended in (Guo et al., 2020) with a larger set of eight MRC datasets from the MRQA shared task (see above). Notably, they also conducted zero-shot transfer experiments in which they find no clear advantage of USE-QA compared to a variant of BM25 that uses the WordPiece vocabulary of BERT. However, after fine-tuning USE-QA on the target datasets, they observe substantial improvements on 3 of 5 benchmarks and conclude that in-domain training data is important for answer sentence retrieval.

Our work considerably differs in three essential aspects:

1. Instead of retrieving answers, we re-rank questions and answers, i.e., we use the result of a retrieval component and select the final output. Thus, we deal with a smaller candidate pool and can use computationally intensive cross-encoders.

2. We deal with long texts instead of sentences. As we have seen in §5.5.1, these settings may require different approaches to achieve optimal results.

3. We study a substantially larger quantity of 140 source domains and evaluate the zero-shot transfer capabilities of 140 individual models on nine datasets. We provide detailed insights on the zero-shot transfer capabilities in regard to domain similarity and training sizes. We also study several different techniques for optimally combining the data of all 140 domains.

7.2 Data and Setup

7.2.1 Training Data

The StackExchange network consists of 172 cQA forums,[2] each devoted to a particular topic such as programming, traveling, finance, etc. As in the previous Chapter 6, we refer to these forums as *domains*. From those 172 forums, 140 are in English and contain more than 1000 unlabeled questions.

We use data from each of the 140 forums and train domain-specific models that can be used for both answer selection and question similarity tasks. This has become possible with the training methods that we investigated in Chapter 6, of which we leverage SSTB here. This requires no labeled training instances—such as duplicate questions or question-answer pairs—and thus allows us to scale our experiments to 140 source domains, i.e., domains from which we transfer.

To recap, SSTB trains models with positive instances x^+ and negative training instances x^-:

$$x_n^+ = (\text{title}(q_n), \text{body}(q_n))$$
$$x_n^- = (\text{title}(q_n), \text{body}(q_m))$$

in which $q_n \neq q_m$ (see §6.2.2 for more details). We randomly sample q_m from the entire corpus. We also explored sampling more similar negative instances during runtime as in (Tan et al., 2016). This, however, did not lead to noticeable gains in preliminary experiments while considerably increasing the training time. However, we note that *retrieving* difficult negative examples with BM25 could be performed prior to training time without increasing computational complexity. We leave this for future work. For computational reasons, we use a maximum of 100k positive training instances.

Our different domains are clearly separated by topic. Because not all domains are equally popular, the training sizes are heavily imbalanced, see Figure 7.1. This allows us to analyse the impact of data size in regard to the transfer performances.

7.2.2 Evaluation Benchmarks

We transfer all models to nine evaluation datasets from different domains (and four data sources). We refer to these datasets as *evaluation benchmarks* in order to clearly distinguish them from the datasets we use for model training. We categorize the evaluation benchmarks into our two main tasks that we have tackled in the course of this thesis: non-factoid answer selection and question similarity.

Most of the evaluation benchmarks have already been presented in the previous chapters. We briefly summarize them again to provide the reader with a better overview. The statistics can be found in Table 7.1.

[2] See https://stackexchange.com/sites. The data from all forums is publicly available https://archive.org/details/stackexchange; both last accessed 11 Dec. 2020.

	Train	Dev	Test	Source
Non-Factoid Answer Selection				
InsuranceQA	12 889	1592	1625	Insurance Library
WikiPassageQA	3332	417	416	Wikipedia
LAS-Apple	5831	1249	1250	StackExchange
LAS-Cooking	3692	791	792	StackExchange
LAS-Academia	2856	612	612	StackExchange
LAS-Travel	3572	765	766	StackExchange
LAS-Aviation	3035	650	652	StackExchange
Question Similarity				
SemEval17	267	-	88	QatarLiving Forums
AskUbuntu	12 584	189	186	StackExchange

Table 7.1: The statistics of the evaluation benchmarks. Train/dev/test refers to the number of query-questions in the splits. For instance, each question in LAS datasets has 100 candidate answers.

7.2.2.1 Non-Factoid Answer Selection (AS)

We re-rank a pool of candidate answers A in regard to a question q. The questions in all datasets are short and do not contain question bodies.

- *InsuranceQA* (Feng et al., 2015) is a benchmark crawled from an FAQ community forum,[3] in which licensed insurance practitioners answer the users' questions. The domain is narrow and only contains questions about insurance topics in the US. We use the recent version 2 of the dataset with $|A| = 500$ candidate answers (retrieved with BM25). Typically, one answer is correct. We used this dataset in Section 3.2 and Chapter 5.
- *WikiPassageQA* (Cohen et al., 2018) was crowd-sourced using Wikipedia articles and is not restricted to a particular domain (although many questions are about history topics). Candidate answers are passages from a single document, based on which the question was formulated. $|A| = 58$ of which 1.6 passages represent correct answers (on average). We used this dataset in Chapter 5.
- *Long Answer Selection (LAS)* datasets were created as part of Chapter 5 from apple, cooking, academia, travel, and aviation StackExchange forums. For a user question, its accepted answer is considered as correct, and negative candidates were collected by retrieving the accepted answers to similar questions (using a search engine with BM25). $|A| = 100$.

We follow the common practice on these benchmarks measuring mean average precision (MAP) on WikiPassageQA and accuracy (P@1) otherwise.

7.2.2.2 Question similarity (QS)

We re-rank a pool of candidate questions C in regard to a query question q. All questions contain titles and bodies—which we concatenate—and are thus long multi-

[3] https://www.insurancelibrary.com/

sentence texts.

- *SemEval17* (Nakov et al., 2017) refers to Task 3b of the SemEval 2017 challenge. This question similarity benchmark contains instances crawled from QatarLiving forums[4]. For each question q, $|C| = 10$ potentially similar questions were retrieved with a search engine and manually labeled for similarity in regard to q.
- *AskUbuntu* (Lei et al., 2016b) is an extension of the dataset by Dos Santos et al. (2015), crawled from the AskUbuntu forum. The train split contains noisy community-labeled duplicate annotations, and the (smaller) dev/test splits were manually annotated for similarity. $|C| = 20$. We used this dataset in Section 4.2 and Chapter 6.

On all question similarity benchmarks, we measure MAP.

7.2.3 Models and Training

7.2.3.1 BERT Models

We use a pointwise ranking architecture based on pre-trained language models. This means that we concatenate the two input texts (separated with SEP token), and learn a linear classifier on top of the final CLS representation for scoring.[5] We optimize the binary cross-entropy loss. This achieves state-of-the-art in-domain results on related datasets with similar training techniques (Zhang et al., 2020; Garg et al., 2020; Mass et al., 2019).

For our zero-shot transfer experiments from single domains in Section 7.3, we use BERT base (Devlin et al., 2019). Later in Section 7.4, we additionally investigate BERT large and RoBERTa large (Liu et al., 2019). All models operate on lowercased texts. The difference between small and large models is that large models have considerably higher capacity. For instance, instead of 12 layers they consist of 24 layers and their internal representations are 1024d instead of 768d.

7.2.3.2 Training

We train our models with SSTB (see §7.2.1). To obtain *in-domain models*, we fine-tune BERT with the respective training data of the benchmark datasets. We train the models for 20 epochs with early stopping for in-domain BERT, and without early stopping for zero-shot transfer. We report the average result over five runs for the in-domain models in AskUbutu and SemEval (due to small evaluation splits) and over two runs for the remaining benchmarks. Following Mass et al. (2019), we sample a maximum of 10 negative candidate answers for each question in WikiPassageQA (new samples in each epoch). For the LAS datasets, we randomly sample 10 negative candidates from the corpus.[6] For InsuranceQA and AskUbuntu, we randomly sample one negative candidate due to their larger training sizes.

[4] https://www.qatarliving.com/forum

[5] These special tokens are part of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019).

[6] We find that this procedure achieves better results compared to our experiment setup in §6.4.3 (∼3pp), where we only sampled one negative candidate.

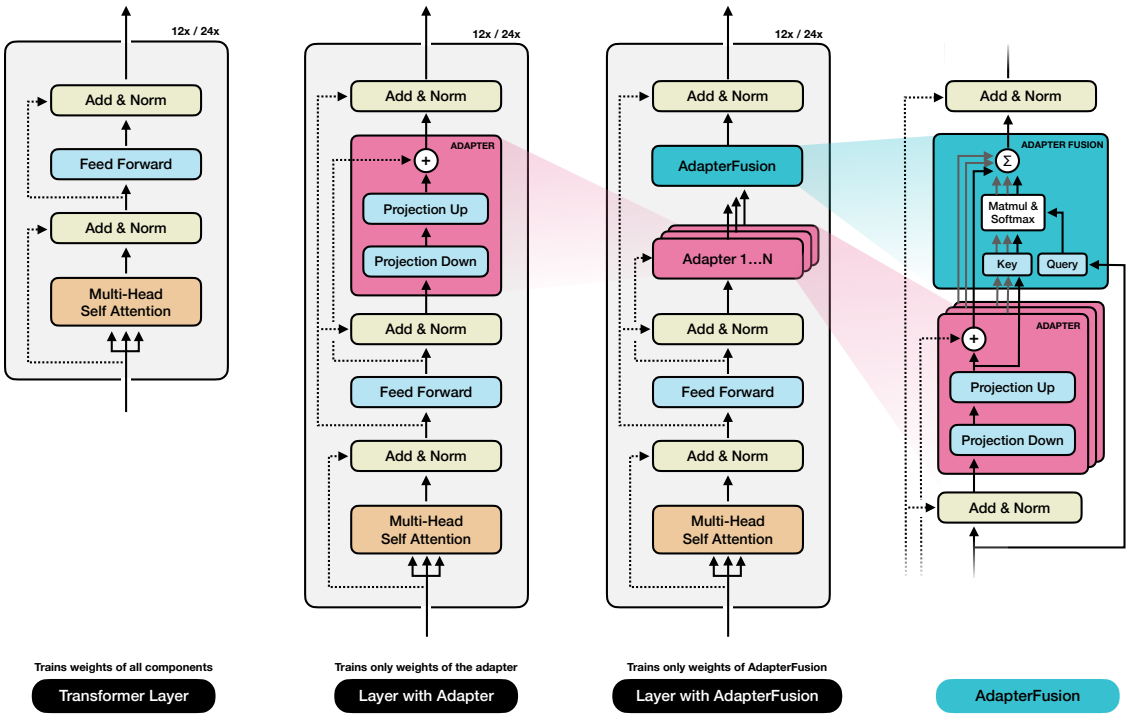

Figure 7.2: A transformer layer without adapter (left), with adapter (middle), and with AdapterFusion (right). Full models consist of 12 (base model) or 24 (large model) layers. The adapter module injects a tiny number of task-specific parameters that adapt the transformer representations by down- and up-projection. Only the adapter weights are trained while the other parameters are fixed. On the right, we visualize AdapterFusion, which learns a weighted average over the output representations of multiple pre-trained adapters (only adapter fusion weights are learned). The adapter architecture is the one proposed by Pfeiffer et al. (2021) and the AdapterFusion is their proposed variant without a value matrix.

7.2.3.3 Adapters

To reduce the storage requirements, and to efficiently distribute our models to the community, we train adapters (Houlsby et al., 2019; Rebuffi et al., 2017) instead of full fine-tuning for our 140 single-domain BERT models. Figure 7.2 visualizes adapter layers, as used in this chapter, in comparison to a standard transformer. Adapters share the parameters of a large pre-trained model—in our case BERT—and introduce a small number of task-specific parameters. With that, adapters transform the intermediate representations in every BERT layer to the training task while keeping the pre-trained model itself unchanged. Adapters have been applied to machine translation (Bapna and Firat, 2019; Philip et al., 2020), (zero-shot) cross-lingual transfer (Pfeiffer et al., 2020b,c; Üstün et al., 2020; Vidoni et al., 2020), and to different transfer learning settings (Stickland and Murray, 2019; Wang et al., 2020b; Lauscher et al., 2020a).

We use the recent architecture proposed by Pfeiffer et al. (2021), which injects fewer layers into the model than the one proposed by Houlsby et al. (2019), and is, thus, also more efficient (Rücklé et al., 2020a). In preliminary experiments, we find that

using adapters in comparison to full model fine-tuning does not decrease the model performance while drastically reducing the number of parameters (one model is ∼5 MB compared to the 440 MB of BERT).

Later in Section 7.4, we then combine different subsets of our 140 adapters with AdapterFusion (Pfeiffer et al., 2021) and compare this approach to multi-task learning. AdapterFusion learns a weighted average over the adapter outputs (see Figure 7.2 on the right) and has been shown to mitigate common pitfalls of multi-task learning, e.g., catastrophic interference between tasks. We skip the value matrix of the original AdapterFusion architecture to avoid introducing additional capacity to the model (such that we are less likely to overfit on our source domains).

7.2.3.4 Hyperparameters

For computational and memory reasons we limit the maximum sequence length to 300 tokens (instead of the maximum of 512 in BERT) for all our models. Similar sequence lengths are commonly used on the benchmarks that we study (e.g., Mass et al., 2019; Tan et al., 2016). For all experiments, we use a batch size of 32 and a linear warmup schedule over one epoch. We train all models for 20 epochs with early stopping of in-domain models, and without early stopping for zero-shot transfer.

For full model fine-tuning on SemEval17, we use a learning rate of 5×10^{-5}, due to the small size of the data set. In all other cases with full model fine-tuning, we use learning rates that we optimized on WikiPassageQA and InsuranceQA. For this, we explored the manual selection of learning rates of 0.001, 0.0001, and 5×10^{-5}. The development scores on InsuranceQA were 43.25, 40.00, and 39.25 (accuracy), respectively. The development scores on WikiPassageQA were 72.69, 71.93, 72.26 (MAP), respectively. We thus chose 0.001 as a learning rate when fine-tuning BERT (and RoBERTa) models.

For the training of adapters and AdapterFusion, we use the learning rates as recommended in Pfeiffer et al. (2021), which are 0.0001 and 5×10^{-5}, respectively.

7.3 Zero-Shot Transfer from 140 Domains

In this section, we study the zero-shot transfer performances of all models trained on individual domains (§7.3.1) and investigate whether domain similarity and training data size are suitable for predicting the best models (§7.3.2).

7.3.1 Performance Scores

Figure 7.3 shows the performances of all models in our zero-shot transfer settings to all nine benchmark datasets. Except for SemEval17, all results are for the dev split.[7] Diamonds ⋄ show the performance of IR baselines and the in-domain BERT.

[7] SemEval17 does not contain a separate dev split.

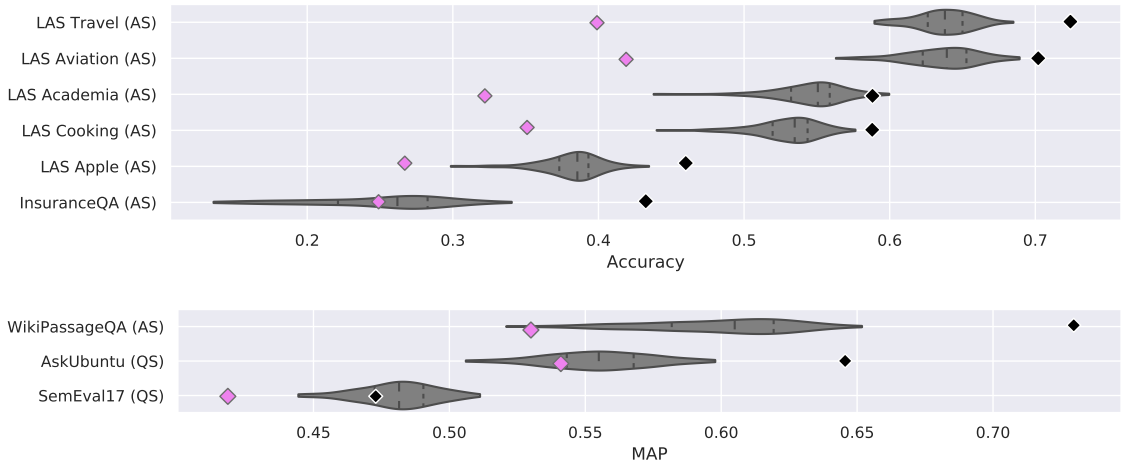

Figure 7.3: Zero-shot transfer performances of all 140 models to the evaluation benchmarks. For benchmarks that contain StackExchange data, we exclude the model from the respective source domain. The violin range visualizes the observed transfer scores, without extension or cut-off for extreme datapoints. Vertical lines show the mean and the quartiles. Diamonds ⋄ show the performances of IR baselines (violet) and in-domain BERT models (black).

IR baselines are the ones that achieved the best results on the benchmark datasets: TF*IDF for LAS and AskUbuntu, BM25 for WikiPassageQA and InsuranceQA, and a search engine ranking for SemEval17 (the official shared task baseline).

7.3.1.1 Zero-Shot Transfer vs. IR Baselines

We observe that the wide range of domain-specific models transfer extremely well to all evaluation datasets. For instance, all models largely outperform IR baselines on six benchmarks. This suggests that learning a *general* similarity function in BERT for our type of data—i.e., short questions and long answers, or pairs of long questions—is important and indeed learned by the models. The low variances of the model performances, especially for more general domains such as Travel, Cooking, and SemEval17, indicate that the domain-specific factors either have a smaller impact, or were already learned during BERT pre-training. This is in line with recent work in ad-hoc retrieval, which showed that BERT models trained on tweets and Wikipedia data transfer surprisingly well to news articles (Akkalyoncu Yilmaz et al., 2019). Other work has shown that IR baselines are often hard to beat, e.g., most neural models trained in-domain on WikiPassageQA perform below BM25 (Cohen et al., 2018). In contrast, we show that a large number of BERT models from a variety of 140 domains outperform these baselines without requiring any in-domain supervision.

7.3.1.2 Zero-Shot vs. In-Domain Models

BERT trained in-domain performs the best in most cases. The difference is larger for expert topics with big training sets (InsuranceQA, AskUbuntu), which shows that our setup provides a challenging test-bed for measuring the generalization capabili-

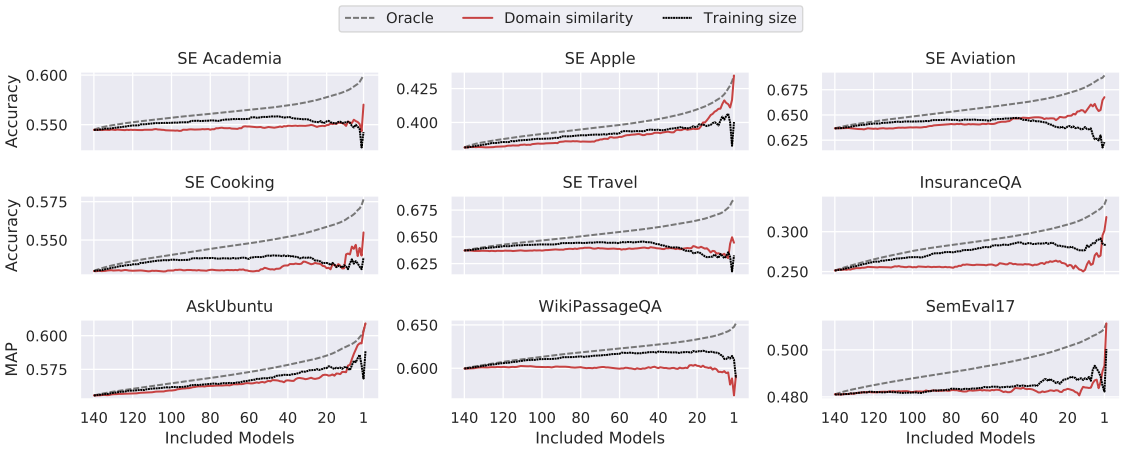

Figure 7.4: The performance scores (y-axis) of subsets of models (x-axis) that are selected by domain similarity or training size (scores are averaged over the included models). The oracle always selects the best models.

ties of models. However, for target domains with few training instances (Table 7.1), the differences of in-domain BERT to the best zero-shot transfer models are much smaller. These setups pose crucial and realistic challenges for text matching approaches (Rücklé et al., 2019a,b; Poerner and Schütze, 2019). For instance, on SemEval17, this results in low performances for in-domain BERT. In contrast, our best zero-shot transfer model achieves a performance of 51.13 MAP—which is 2.13 points better than the best challenge participant in (Nakov et al., 2017).

This clearly demonstrates that zero-shot transfer is a suitable alternative for in-domain models, which also contrasts the large performance degradations often observed with previous models such as LSTMs (Shah et al., 2018). We further find no substantial differences between question similarity and answer selection *tasks*, which are both not explicitly learned during training.

With this, we take an important step towards overcoming the boundaries between individual tasks and domains in cQA. This also contrasts our experiments in previous chapters, where we trained separate models for different setups. We have now presented some evidence that the BERT models trained with our data can be robustly transferred.

7.3.2 Analysis

Due to the large number of 140 domain-specific models, each trained on datasets of different sizes, we are able to perform unique analyses regarding the zero-shot transfer performances to the target tasks.

Ideally, we would like to identify a small number of models that transfer well to a given dataset, without requiring costly evaluations of all models. In the following, we probe two intuitive domain selection techniques in regard to the transfer performances: (1) domain similarity and (2) training size. To simulate an optimal selection, we define an *oracle* that always correctly identifies the best models. Figure 7.4 presents our findings.

Domain similarity. To measure the domain similarity, we learn embeddings for the questions of all datasets with Sentence-RoBERTa (Reimers and Gurevych, 2019). For each dataset, we obtain the centroid of embedding vectors and calculate the domain similarity to other datasets with cosine similarity.

Domain similarity is most effective when selecting models for benchmarks of technical domains, e.g., AskUbuntu, LAS-Apple, and LAS-Aviation in Figure 7.4. However, this does not hold true for benchmarks of non-technical domains such as LAS-Travel or WikiPassageQA. In those cases, only considering the most similar source domains does not improve the average model performance. One reason might be that there do not exist many similar non-technical domains within StackExchange, from which models can transfer domain-specific idiosyncrasies. However, as we have shown in §7.3.1, such knowledge is not essential, i.e., a large number of models from more distant domains achieve good zero-shot transfer performances.

Table 7.2 lists the best models and the most similar domains for all benchmarks. Some of the best models are from distant domains—e.g., "Ethereum" for WikiPassageQA or "SciFi" for LAS-Travel. This illustrates the importance of considering a broad selection of source domains, including ones that are not intuitively close. A potential reason for this observation may be that there is an interaction between the BERT pre-training, the source domain training, and the target domain data. However, it remains unclear how to best leverage or predict the impact of these interactions. Analyzing this in isolation and beyond cQA is a promising direction for future work. In a more general context, Vu et al. (2020) show that additional pre-training of BERT models on distant source tasks can be beneficial for many transfer tasks. For example, they find that part-of-speech tagging as a pre-training task can positively impact the transfer to machine reading comprehension tasks. Understanding the reasons for such interactions may result in better source dataset selection—including in our zero-shot transfer setting.

Training size. The average performance of our models after removing the smallest domains improves more consistently (see WikiPassageQA and InsuranceQA in Figure 7.4). This shows that the training size is more suitable for identifying models that achieve low performance scores—e.g., models that are trained on narrow expert domains. However, the training size alone cannot identify the *best* models for zero-shot transfer. It is thus crucial to not limit the scope to the largest datasets at hand when exploring suitable training tasks. Notably, this contrasts the common procedure of only including the largest domains for transfer (Shah et al., 2018).

An interesting future direction could be explore the size of specific data regions. For example, it has been shown that training with "ambiguous" (and thus challenging) examples can yield models with better generalization capabilities (Swayamdipta et al., 2020). If this also holds in our setup, the number of such examples may be better suited to predict the models' zero-shot transfer capabilities than the overall training data size. This could also be useful for further *improving* the zero-shot transfer capabilities of the different models, e.g., by filtering the training data.

	Best Models	Most Similar Domains
LAS-Travel	scifi(40k, 0.61); money(18k, 0.64); diy(36k, 0.61); space(7k, 0.55); cooking(14k, 0.45);	expatriates(3k, 0.90); law(9k, 0.75); civicrm(7k, 0.75); eosio(1k, 0.73); expressionengine(7k, 0.73);
LAS-Aviation	biology(15k, 0.69); diy(36k, 0.68); sports(3k, 0.62); physics(158k, 0.73); rpg(26k, 0.71);	space(7k, 0.81); engineering(5k, 0.76); ham(1k, 0.76); worldbuilding(14k, 0.75); gaming(65k, 0.74);
LAS-Academia	scifi(40k, 0.54); money(18k, 0.50); android(35k, 0.51); aviation(11k, 0.52); superuser(442k, 0.58);	writers(6k, 0.78); matheducators(1k, 0.76); workplace(13k, 0.75); softwareengineering(38k, 0.74); pm(3k, 0.74);
LAS-Cooking	gardening(8k, 0.70); money(18k, 0.43); security(36k, 0.52); academia(19k, 0.38); space(7k, 0.49);	homebrew(3k, 0.82); sustainability(1k, 0.72); health(4k, 0.72); skeptics(6k, 0.71); gardening(8k, 0.70);
LAS-Apple	superuser(442k, 0.94); askubuntu(344k, 0.89); android(35k, 0.91); unix(181k, 0.90); gis(88k, 0.74);	superuser(442k, 0.94); windowsphone(2k, 0.93); elementaryos(3k, 0.92); android(35k, 0.91); unix(181k, 0.90);
InsuranceQA	cooking(14k, 0.38); travel(24k, 0.51); android(35k, 0.39); diy(36k, 0.49); security(36k, 0.51);	money(18k, 0.78); law(9k, 0.69); economics(5k, 0.66); freelancing(1k, 0.66); quant(8k, 0.62);
WikiPassageQA	politics(6k, 0.81); ethereum(16k, 0.61); physics(158k, 0.66); money(18k, 0.57); travel(24k, 0.60);	history(7k, 0.91); literature(2k, 0.84); movies(13k, 0.83); mythology(1k, 0.82); politics(6k, 0.81);
AskUbuntu	superuser(442k, 0.81); apple(61k, 0.79); blender(37k, 0.54); magento(70k, 0.54); electronics(85k, 0.52);	superuser(442k, 0.81); elementaryos(3k, 0.81); apple(61k, 0.79); unix(181k, 0.76); serverfault(288k, 0.76);
SemEval17	travel(24k, 0.69); diy(36k, 0.52); gamedev(31k, 0.53); blender(37k, 0.48); gaming(65k, 0.61);	travel(24k, 0.69); expats(3k, 0.67); webmasters(21k, 0.64); freelancing(1k, 0.63); workplace(13k, 0.62);

Table 7.2: The best models and the most similar domains for all nine benchmarks. Parentheses show the training size and the domain similarity (between 0 and 1). Underlined domains are in the top-5 of most similar and best models.

		Tr	Co	Ap	Ac	Av	IQA	Σ	AU	WP	SEv	Σ
		Accuracy scores							*MAP scores*			
Best models §7.3		65.4	58.1	43.1	56.5	65.0	35.3	53.9	63.3	67.9	51.1	60.8
Self-Supervised Training												
MT	largest	64.5	56.7	40.9	56.0	65.3	32.8	52.7	63.2	66.3	50.1	59.9
AF	largest	63.4	57.7	42.9	59.1	65.9	28.6	52.9	63.1	67.0	**50.9**	60.3
MT	balanced	65.0	60.0	43.2	55.7	65.1	**38.2**	**54.5**	**63.4**	**68.3**	48.3	60.0
AF	balanced	62.2	58.0	**43.5**	**59.4**	66.2	29.8	53.1	62.9	67.5	48.0	59.5
MT	all	**66.1**	**60.3**	43.0	57.0	**66.4**	31.5	54.0	63.3	68.2	49.8	**60.4**
Extended Data												
MT	balanced	67.8	60.9	46.5	58.9	**69.1**	**34.9**	56.3	**65.1**	**67.7**	48.6	60.5
MT	all	**72.4**	**63.1**	**45.8**	**61.1**	68.0	34.7	**57.5**	64.1	66.8	**52.3**	**61.1**

Table 7.3: Results of MT and AF with different sets of source domains for MultiCQA^{B}. The first five columns are LAS-Travel, Cooking, Apple, Academia, and Aviation. AU is AskUbuntu, IQA is InsuranceQA, WP is WikiPassageQA, SEv is SemEval. Σ shows the average performance of benchmarks that use the same performance measure.

Summary We have established that neither domain similarity nor training data size alone are suitable for identifying the best models in our setting. This demonstrates the importance of considering a broad selection of source domains instead of merely relying on the most similar or largest ones. These insights could also be beneficial for researchers in related areas, e.g., to consider a wider range of domains and source datasets prior to domain adaptation.

7.4 Zero-Shot Transfer from Combinations of Multiple Domains

We now investigate how to best combine *multiple* source domains for zero-shot transfer. We denote our models as `MultiCQA`.

7.4.1 Setup

Combination methods. We use (1) multi-task learning and share all model layers across the domains. In each minibatch, we sample instances from a single source domain, which we select with a round-robin schedule. Models trained in this manner are denoted as `MT`.

In addition, we (2) combine knowledge from our domain adapters (Section 7.3) with AdapterFusion (`AF`; Pfeiffer et al., 2021). This learns a weighted combination of multiple (fixed) adapters in each BERT layer and is typically trained on the target task. We adapt this approach to our zero-shot setup and train it with multi-task learning as above (we have introduced AF in §7.2.3.3).

Data. We use the training data of §7.2.1 and exclude the domains that are used in any of the evaluation benchmarks.[8] We use three sets of source domains: (1) the set of 18 topically `balanced` domains, consisting of the top-three domains (according to the number of questions asked) from each of the six broad categories as defined by StackExchange[9]; (2) the `largest` 18 domains according to the number of asked questions; (3) `all` including 134 domains.

We additionally study the impact of extending our training data with community-labeled instances from the source domains. For a positive instance of question title and body, we add positive instances of (a) question title and accepted answer, and (b) question title and body of a duplicate question. We name this `extended data`.

Models. If not otherwise noted, we fine-tune BERT base. We also experiment with BERT large and RoBERTa large (all uncased). For MultiCQA models this corresponds to MultiCQA$^{\text{B}}$, MultiCQA$^{\text{B-lg}}$, and MultiCQA$^{\text{RBa-lg}}$. The training procedure, number of runs, and hyperparameters are as in §7.2.3.

We additionally compare our models to the query-response encoder USE-QA (Yang et al., 2020), which is a state-of-the-art model for retrieving answer sentences in zero-shot transfer setups. The IR baselines are the same as in §7.3.1 (TF*IDF for LAS and AskUbuntu, BM25 for WikiPassageQA and InsuranceQA, and a search engine ranking for SemEval17—the official challenge baseline).

7.4.2 Results

Multiple source domains. Table 7.3 shows the results of MultiCQA$^{\text{B}}$ with MT and AF for the different sets of source domains, and compares this to the respective best single-domain models of Section 7.3.

We observe that the balanced set of source domains achieves better results than combining domains with the largest training sets, which shows that diversity is more important than size. Most importantly, MT with data from *all* source domains outperforms the respective best single-domain model in 6 out of 9 benchmarks. This demonstrates that common problems of MT—catastrophic interference between training sets in particular—do not occur in our setup. This also reveals that combining source domains on a massive scale is possible.

MT and AF are both effective combination methods, with minor differences on most datasets. However, MT performs considerably better on InsuranceQA, which is a very narrow expert domain. The reason for this may be that AF combines *fixed* domain-specific adapters, which can lead to reduced performances if all adapters are not related to the target domain. AF can also lead to better results, e.g., on LAS-Academia. We include an analysis of AF for these datasets in Appendix E.2, where we also visualize the learned fusion weights. Interestingly, we find that the fusion weights do not differ much between the two datasets. However, when we remove

[8] AskUbuntu, aviation, travel, cooking, academia, apple.

[9] Technology, culture, life, science, professional, and business. See Appendix E.3 for the list of included domains.

	Tr	Co	Ap	Ac	Av	IQA	Σ	AU	WP	SEv	Σ
	Accuracy scores							*MAP scores*			
IR Baselines	39.9	35.1	26.7	32.2	41.9	24.9	33.4	54.1	53.0	41.9	49.7
Zero-Shot Transfer											
USE-QA	65.3	58.5	44.6	46.2	53.1	35.1	50.4	**67.8**	53.2	52.7	57.9
MultiCQA^{B}	72.4	63.1	45.8	61.1	68.0	34.7	57.5	64.1	66.8	52.3	61.1
MultiCQA$^{B\text{-}lg}$	75.5	64.6	50.0	64.0	72.0	32.8	59.8	66.5	69.8	51.6	62.6
MultiCQA$^{RBa\text{-}lg}$	**77.8**	**72.0**	**56.8**	**70.4**	**76.6**	**41.9**	**65.9**	63.3	**73.3**	**52.9**	**63.2**
In-Domain Models											
Previous best	69.5†	58.3†	47.3†	58.7†	65.5†	49.8‡	58.2	69.1†	74.9*	51.6$^{\diamond}$	65.2
BERT	68.7	59.0	47.0	59.0	64.5	42.2	56.7	67.3	75.1	47.3	63.2
BERT-lg	72.5	62.4	47.2	60.0	68.3	42.7	58.8	67.5	76.2	45.9	63.2
RoBERTa-lg	70.9	68.4	50.7	66.3	68.7	44.9	61.6	70.2	79.7	48.7	66.2
MultiCQA$^{RBa\text{-}lg}$	**80.5**	**76.8**	**60.2**	**72.1**	**81.8**	**50.8**	**70.3**	**72.3**	**81.4**	**53.6**	**69.1**

Table 7.4: The results of zero-shot transfer and in-domain models. The first five columns are LAS-Travel, Cooking, Apple, Academia, and Aviation. AU is AskUbuntu, IQA is InsuranceQA, and WP is WikiPassageQA, SEv is SemEval. Σ shows the average performance of benchmarks that use the same performance measure. † shows the scores of the best BERT models of Chapter 6, ‡ is the MICRON model (Han et al., 2019), * is the BERT model in (Ma et al., 2019), and $^{\diamond}$ is MV-DASE (Poerner and Schütze, 2019).

a single adapter, we also observe that AF automatically replaces it with another adapter from a similar source domain, indicating that this approach is robust.

Additional labeled data. In Table 7.3, we also see that extending the training data of MT models with additional labeled data from question-answer pairs and question duplicates considerably and consistently improves the performances in 16 of 18 cases. This improves the performance of MT with all source domains on the nine benchmarks, which shows that our approach is very effective when combining a large number of smaller domains. Due to these consistent improvements, we train all our large MultiCQA models with *MT all* and the extended data.

Comparison to in-domain models. In Table 7.4, we compare our large MultiCQA models to in-domain models. We find that the additional capacity of the models and the better initialization with RoBERTa considerably improves the zero-shot transfer performances (on average). Our best *zero-shot* MultiCQA$^{RBa\text{-}lg}$ model outperforms USE-QA on eight benchmarks, and performs better than the best *in-domain* models on all LAS datasets and on SemEval17.

Our MultiCQA models are thus highly effective *and* re-usable across different domains and tasks. This clearly demonstrates the effectiveness and feasibility of training suitable models for zero-shot transfer that are widely applicable to different realistic settings.

One downside, however, is that MultiCQA models are cross-encoders that can only

be applied to re-ranking setups—as opposed to efficient end-to-end retrieval with embeddings (e.g., Guo et al., 2020). Furthermore, our best results are obtained with large transformers, which require even more computational resources. It could, therefore, be interesting to explore efficient combinations of retrieval and re-ranking methods—e.g., to re-rank a dynamic number of candidate answers determined by a more efficient retrieval method. Besides, future work could try to apply compression techniques to obtain smaller and thus efficient models, e.g., by knowledge distillation (Hinton et al., 2015) which has been proven effective in many related NLP settings (Sanh et al., 2019; Sun et al., 2020b).

Further in-domain fine-tuning. Finally, we show that MultiCQA$^{\text{RBa-lg}}$ is an effective initialization for in-domain fine-tuning. This leads to large gains and achieves the best results on all nine benchmarks.

7.5 Analysis

We manually inspect 50 instances of InsuranceQA and AskUbuntu for which our zero-shot transfer model MultiCQA$^{\text{RBa-lg}}$ selects a wrong answer or an unrelated question. We find that the texts are always on-topic, i.e., many aspects of the question are included in the selected answers (InsuranceQA) or in the potentially similar questions (AskUbuntu). This includes keywords, phrases (often paraphrased), names, version numbers, etc. The most common source of mistakes is that an important aspect of the question appears to be ignored or is (likely) not understood by the model. For instance, many aspects of the question might be mentioned in a potentially similar question or a selected answer, but in the wrong order or not in relation to each other. Table 7.5–7.8 show examples of such cases. We find that this type of mistake affects 25 of 50 instances in AskUbuntu, and 10 of 50 instances InsuranceQA.[10]

Future work could, thus, achieve further improvements by enhancing the overall understanding of question and answer texts. Current models seemingly match similar keywords or phrases of the questions and answers, often without truly understanding them. One idea could be to augment the training sets with more difficult, artificial examples. For instance in a question-answer pair one could mask out co-occurring words, potentially guiding the model to learn more complex relationships between the texts beyond keyword matching. This may be particularly interesting in zero-shot settings such as ours, as it has been shown that "ambiguous" training examples—i.e., more challenging ones—can improve the generalization capabilities of models in other NLP tasks (Swayamdipta et al., 2020).

7.6 Chapter Summary

In this final chapter, we studied the zero-shot transfer capabilities of text matching models on a massive scale, with 140 different source domains and nine datasets of

[10] In 8/50 cases in AskUbuntu and 30/50 cases in InsuranceQA our model actually selects relevant texts, e.g., correct answers or similar questions (which are not labeled as such).

Query question: Passing parameters to the installer for 14.04? The installer for 14.04 gave me no chance (that I took notice of) to pass parameters [...]
Most similar (MultiCQA$^{\text{RBa-lg}}$): Which key combination would allow me to pass parameters to kernel? During boot I want to pass some parameters like the runlevel , nomodeset to kernel during the booting process [...]
Ground truth: How can i customize the Ubuntu installer? I would like to know how can I customize the Ubuntu installer not customize Ubuntu , I just want to modify the installer [...]

Table 7.5: A mistake of MultiCQA$^{\text{RBa-lg}}$ (zero-shot transfer) on AskUbuntu. The model likely does not understand the intention of the query, which is to change the behavior of the installer (and not merely passing parameters to something).

non-factoid answer selection and question similarity tasks. By investigating such a large number of models, we provided an extensive comparison and fair baselines to the different combination methods, and were able to provide insightful analyses based on our large sample size.

We have shown that:

1. BERT models trained in a self-supervised manner on our cQA forum data transfer well to all our benchmarks, even across distant domains.
2. Training data size and domain similarity alone are not suitable for identifying the best models for zero-shot transfer in our setting, showing that elaborate strategies are necessary for automatically finding the most suitable source domains. Merely relying on the largest or most similar source domains does not achieve the best results in our experiments.
3. Our MultiCQA approach that combines self-supervised and supervised training data across a large set of source domains outperforms many in-domain baselines and achieves the best zero-shot performances on six benchmarks.
4. Fine-tuning MultiCQA$^{\text{RBa-lg}}$ in-domain further improves the performances and achieves the best results overall.

We significantly expanded upon our previous chapters in terms of (a) **scale**, by exploring 140 source domains; (b) **variety**, by studying performances on nine datasets of different tasks, domains, and data sources; (c) **re-use**, by obtaining a single model that generalizes well to several different target datasets.

Our work opens a wide array of possible extensions that could be explored in the future. For instance, combining our approach with additional pre-training objectives such as the Inverse Cloze Task (Chang et al., 2020) could substantially increase the amount of training data for the large quantity of smaller forums. Exploring combinations of different data sources, e.g., Wikipedia data in addition to our cQA forum data could yield even more generalizable models. Researchers could use our 140 domain-specific adapters and investigate further combination techniques to make them more broadly applicable. Finally, we could also study more distant transfer cases, e.g., across domains *and languages* by fine-tuning a multilingual MultiCQA model based on XLM-R (Conneau et al., 2020) and measuring its zero-shot transfer

capabilities to Chinese (He et al., 2018) or Arabic (Nakov et al., 2016) cQA datasets. This could provide us with interesting insights on the interaction of domain and language transfer in cQA.

Question: Can I buy a car without insurance?
Selected answer (MultiCQA$^{\text{RBa-lg}}$): You most certainly can get auto insurance without a car. if you needed toborrow, test drive, rent, or lease a vehicle for whatever reason you would purchase what is called a drive other car policy. [...]
Ground truth: Depending in the state you live in and also if your are financing the car. if you have a loan on the car the financial institution will require insurance before you even leave the car lot. if you are buying from a private party they may not require this but in most states you can not even get your license plates with out insurance.

Table 7.6: A mistake of MultiCQA$^{\text{RBa-lg}}$ (zero-shot transfer) on InsuranceQA. This shows that the model does not interpret the individual keywords within context, i.e., it does not differentiate between *car without insurance* and *insurance without car*. We underline important aspects that differ in the most similar candidate.

Query question: Why is state farm life insurance so expensive?
Selected answer (MultiCQA$^{\text{RBa-lg}}$): State farm offers life insurance, both term and permanent through their captive agents along with property and casualty insurance. However, unlike the latter types of coverage [...]
Ground truth: Every carrier has their own rates - these are based off a long calculation of actuarial values and mortality tables. Some carriers are more aggressive than others and are willing to take on more risk [...] more conservative carriers feature higher rates. So it's hard to say one carrier is just very expensive.

Table 7.7: Another mistake of MultiCQA$^{\text{RBa-lg}}$ (zero-shot transfer) on InsuranceQA. The selected answer describes state farm life insurance, whereas the ground truth explains *why* it can be expensive. We underline important aspects that differ in the most similar candidate.

Query question: How many maximum CPUs does Ubuntu support by default? I think this is kernel dependent and probably will change over time depending on the kernel a release uses, correct me if wrong I'd like to know [...]
Most similar (MultiCQA$^{\text{RBa-lg}}$): Creation of /proc/stat. Which function of the kernel creates and writes the information for /proc/stat. In this, would like to know when kernel gets the CPU information (recognises number of CPUs) [...]
Ground truth: Ubuntu Linux 14.04 LTS server edition information need. I was wondering what's the maximum RAM, and maximum CPUs does the Ubuntu Linux 14.04 LTS server edition can handle [...]

Table 7.8: A mistake of MultiCQA$^{\text{RBa-lg}}$ (zero-shot transfer) on AskUbuntu. The query question asks for the maximum number of CPUs that can be handled by a kernel. The selected similar question, however, asks for information *where* the kernel gets its information about the available CPUs—not the *maximum number* of CPUs. We underline important aspects that differ.

Chapter 8

Conclusion

In this dissertation, we focused on two fundamental challenges arising in the context of community question answering, namely, (1) learning better and more universal text representations; (2) dealing with scenarios where we have access to only little or no labeled training data. We addressed four research questions and thereby tackled these challenges from several different perspectives, which we summarize below.

8.1 Summary

RQ1: How can we learn effective and efficient representations of questions and answers?

Learning dense vector representations that encode the meaning of questions and answers is critical for a wide range of applications. For example, we can use them to efficiently compare new questions to existing answers in cQA forums. In Chapter 3, we argued that previous attention-based models for answer selection tasks have a crucial shortcoming in that they leverage question information when learning answer representations. This procedure of dependent encoding is inefficient at run-time as it requires us to learn new answer representations for each question.

We addressed this shortcoming in Section 3.1 and proposed "LSTM-based importance weighting", a self-attentive approach that learns the importance of text segments in questions and answers *independently* from each other. We showed that our approach achieves on-par or better results in answer selection tasks compared to attention mechanisms with dependent encoding. We also illustrated that self-attentive approaches are, by design, not negatively affected by misleading attention as opposed to calculating the importance based on the similarity of question-answer segments. Our results suggested that this may be beneficial for datasets where questions and incorrect answers have a high lexical overlap.

Motivated by the effectiveness of attention mechanisms, in Section 3.2, we presented a prototypical end-to-end QA system for the interactive visualization of attention weights. Our system allows researchers to explore different attention mechanisms

interactively and compare them side-by-side. We widely used this system throughout our studies, and the experimental framework that we introduced as part of this system has been the foundation for several other works.

RQ2: How can we obtain approaches to representation learning that transfer well to different languages?

Sufficient amounts of data, either labeled instances to train suitable task-specific representations or unlabeled passages for answering questions, are commonly only available in English. In Chapter 4, we, therefore, studied cross-lingual transfer from two orthogonal perspectives to address these challenges. (a) Applying models trained on English data to other languages. (b) Leveraging English forum data to answer questions posed in other languages.

In Section 4.1, we investigated universal sentence embeddings, which are reusable across a wide variety of tasks such as question classification, sentiment analysis, and argumentation mining. Previous work either studied them only monolingually or cross-lingually for only few individual datasets We argued that the notion of "universality" should apply to both, i.e., that sentence embeddings should transfer well across different tasks *and* across languages. We proposed concatenated power mean word embeddings as universal cross-lingual sentence embeddings, a training-free generalization of average word embeddings with two ingredients. (1) The concatenation of multiple word embeddings to capture different kinds of information, e.g., syntactic and semantic information. (2) The extraction of more information from the sequence of word embeddings by leveraging different power means. We demonstrated that our embeddings yield significant gains upon average word embeddings and outperform many other complex techniques monolingually. Our embeddings achieve performances close to that of InferSent (Conneau et al., 2017), a computationally demanding neural model trained on large amounts of high-quality natural language inference data. More importantly, we have shown that we can effortlessly extend our sentence embeddings to new languages by incorporating cross-lingual word embeddings. Due to its simplicity, our approach achieved the best results when transferring from English to German and French, where it outperformed three cross-lingual adaptations of InferSent.

In Section 4.2, we complemented this by studying a different perspective to cross-lingual transfer, namely, to machine translate the input from German to English and continue monolingually. We investigated the impact of a neural machine translation model on the performance of question similarity models in programming and operating systems forums. We found that the translation quality has a considerable impact on the cross-lingual question similarity performance, and we improved this by adapting the translation model to our specialized domains with back-translation.

In summary, we addressed different monolingual and cross-lingual perspectives of representation learning. We obtained effective attention-based models that can also be more efficiently applied, and we contributed to making representations more broadly accessible across different languages—with universal sentence embeddings and through machine translation.

RQ3: How can we train cQA models in settings with limited labeled training data?

The second fundamental challenge we focused on was to deal with scenarios in which labeled training data is scarce. Many cQA forums do not offer enough annotated question-answer pairs and no duplicate questions for model training, which poses considerable challenges to previous approaches. We addressed this from two perspectives: (a) training models with small amounts of labeled data, and (b) training models with only unlabeled data.

In Chapter 5, we presented COALA, a shallow task-specific network architecture specialized in answer selection, containing only one trainable layer. This layer learns representations of question and answer aspects, modeled as word n-grams, and determines the coverage of question aspects by the answer for scoring. We showed that COALA scales well to long answers and outperforms the more complex compare-aggregate architecture of Wang and Jiang (2017) by 4.5 percentage points (on average) over six datasets from different domains. Furthermore, we demonstrated that COALA is extensible with learned power mean operations for capturing more information about the sequence of covered question aspects. Finally, we have established that COALA outperforms standard IR baselines already when trained with as little as 25 labeled question-answer pairs, which shows our model's broad applicability.

In Chapter 6, we then studied how to train cQA models *without* labeled training data using two methods: (1) we proposed duplicate question generation, where we generate a new title from a body and consider it as a duplicate to the original question's title for training; (2) we broadly studied self-supervised training where we predict whether a title and body are from the same or from different questions. Both methods yield better question similarity models compared to adversarial domain adaptation in an optimal transfer setup, even when using the same number of training instances. We have shown that by leveraging larger amounts of unlabeled data, duplicate question generation and self-supervised training can achieve substantial improvements and outperformed supervised in-domain training with 9k labeled instances on two datasets. Finally, we have found that both methods are widely applicable to cQA. For instance, duplicate question generation transfers well to unseen domains, and self-supervised training with title-body information yields models suitable for answer selection. Finally, we were also able to fine-tune BERT models with self-supervised training, resulting in improved performances compared to previous network architectures.

RQ4: To which extent do text matching models generalize to unseen cQA tasks and domains?

Besides training domain- and task-specific models, either under small data conditions or with unlabeled data, we argued that it is desirable to obtain one model that generalizes well to several unseen scenarios. Generalizable models are especially useful in the context of cQA because there exist a large number of forums with different topics (i.e., different domains).

In our last Chapter 7, we broadly studied the zero-shot transfer capabilities of cQA text matching models, by training 140 models on different forums and transferring them to nine evaluation datasets of question similarity and answer selection tasks. This has been made possible by self-supervised training with title-body pairs, which we applied to almost all English StackExchange forums. We found that the large majority of models transfer surprisingly well, largely outperforming standard IR baselines. For instance, on six datasets, all 140 models achieved better performance scores. More importantly, our large sample size allowed us to perform unique analyses in regard to domain similarity and training data size. We revealed that both may not be optimal in predicting the best zero-shot transferability in our setting, stressing the need for more elaborate strategies to identify the best source domains. Moreover, we studied techniques to incorporate information from multiple source domains, namely, using multi-task learning and AdapterFusion. We demonstrated that our setup is not affected by catastrophic interference between source tasks, which allowed us to scale multi-task learning to all source domains. We proposed combining self-supervised and supervised training signals to obtain our MultiCQA model, which outperformed in-domain BERT on six benchmarks. We have shown that MultiCQA is a suitable initialization for further in-domain fine-tuning.

In summary, the second part of our thesis covered several complementary perspectives on how to deal with limited labeled training data. We addressed this from the perspective of training shallow models, leveraging only unlabeled data for model training, and from the perspective of obtaining generalizable models for zero-shot transfer in cQA.

8.2 Outlook

Over the past years, we have witnessed an unprecedented disruption in the field of NLP, which parallels the one observed in image recognition just a few years earlier (cf., Ruder, 2018) with the emergence of large pre-trained models such as ResNet (He et al., 2016) and GoogLeNet (Szegedy et al., 2015). Transfer learning in NLP, likewise, and most prominently with the emergence of BERT (Devlin et al., 2019) and its several well-known successors (e.g., Liu et al., 2019; Lewis et al., 2020), have transformed the field—we have leveraged these pre-trained models in the final chapters of this thesis. Going forward, we believe that this new paradigm will unquestionably continue to dominate NLP research. Importantly, this also presents us with novel research challenges, opening up a variety of future research directions. Many of these are closely related to the topics we touched upon in this thesis. In the following, we will highlight potential future directions that stand at the intersection to what may be some of the most important directions in NLP.

Efficiency. Efficiency can be considered from a wide range of perspectives, for instance, training and sample efficiency (including few-shot learning), model efficiency at inference time (latency, throughput), and parameter efficiency. The motivations are manifold, e.g., environmental aspects (Strubell et al., 2019), the democratization of modern NLP (i.e., making state-of-the-art models accessible to researchers with little resources), cost-effectiveness, or being real-time capable (and other feature-

related aspects). This strongly relates to the increasingly popular area of *Green AI*, which "[...] refers to AI research that yields novel results without increasing computational cost [...]" (Schwartz et al., 2019).

Efficiency is also at the core of many research ideas presented in this thesis. In Section 3.1, we have learned attentive representations of questions and answers independently, making it possible to pre-compute representations of all answers within a corpus. In Section 4.1 we have proposed training-free sentence embeddings, which are cheap to compute, yet highly effective. In Section 4.2 we have adapted a machine translation model to our target domain, making it possible to continue monolingually without requiring further cross-lingual training. In Chapters 5–7, we have trained models with little labeled data, and we have obtained one model that is re-usable across many different domains. What we have *not* done, however, is to improve the computational efficiency of pre-trained transformers—even though we leveraged them in our final chapters. We believe that this is an important topic going forward. For instance, we have seen that our MultiCQA model based on BERT-large achieves better results than the one based on the smaller BERT-base model. If we continue this trajectory, we will likely obtain even better models that, at some point, will be unusable. We believe that future work should focus on obtaining better models at *equal size*.

Managing the scale of our models in NLP is a challenge that gained considerable popularity recently. Large pre-trained transformers are notoriously deep and contain up to hundreds of billions of parameters (Brown et al., 2020), making them slow at training and at inference time. Training such large models, however, has a central advantage: independent of their architecture, the test loss, and sample efficiency scales with the allocated parameter budget (Kaplan et al., 2020). Techniques for efficiently pruning such models—optimally before training time, cf. the lottery ticket hypothesis (Frankle and Carbin, 2018)—are needed. Other strategies to mitigate these drawbacks include dropping layers (Fan et al., 2020; Rücklé et al., 2020a), distillation (Sanh et al., 2019; Sun et al., 2020b), efficient attention (Katharopoulos et al., 2020; Choromanski et al., 2021), and adapters (Houlsby et al., 2019; Bapna and Firat, 2019; Pfeiffer et al., 2020a).

Researching which techniques are well-suited for our downstream task, e.g., question similarity and answer selection in cQA, may yield additional important insights. How many model weights can we prune for cQA tasks? Do we need global attention or is local attention sufficient? How does this differ from other classification and semantic similarity tasks? If there are any differences, why? In conclusion, there exists a great need and considerable potential for research in the direction of obtaining efficient approaches.

Question and answer generation. Large pre-trained models do not only have a continued impact on classification and regression tasks. With T5 (Raffel et al., 2020), BART (Lewis et al., 2020), and GPT-3 (Brown et al., 2020) they have also brought significant progress to text generation research. Two such application scenarios are particularly interesting in the context of cQA: (1) automatic question generation, (2) automatic answer generation.

In Chapter 6, we proposed *duplicate* question generation (DQG) to obtain synthetic training data for question similarity tasks. Do the generated examples improve when we leverage large pre-trained models? How many training examples do we need to obtain good question generation models for DQG with pre-trained transformers? Do they transfer better to other domains? What is a good generated question? With accessible models such as the aforementioned ones (and others), we could now address those important research questions. Indeed, question generation has very recently become popular in the context of end-to-end information retrieval (which we address next), where such models are being used to generate synthetic training queries for unlabeled documents (Ma et al., 2020; Liang et al., 2020). Notably, Ma et al. (2020) use a very similar approach to ours, which is to train a question generation model on question-answer pairs from cQA forums. However, the above research questions are still largely unexplored, opening up a considerable gap for future work.[1]

Conversely, one can also generate *answers* for given questions. For instance, Nakatsuji (2019) generate "love advice" for questions posed in cQA forums. Similarly, Fan et al. (2019) generate explanatory answers for non-factoid questions, optionally leveraging additional support documents that have been retrieved from the Internet. The advantage of generative approaches as opposed to answer selection techniques is that they can (theoretically) answer the questions in more detail and can also address small subtleties of the question. However, not much evaluation has been conducted to quantify the correctness of generated answers. Interesting research directions are then (a) how to ensure that the model does not hallucinate critical information (see, e.g., Zhou et al., 2020) and thus generates an incorrect answer; (b) how to properly combine and reason over the retrieved support documents.

Dense Information Retrieval. In this thesis, we have assumed a retrieval component that presents us with potentially similar questions or candidate answers from a large background corpus (as outlined in our background Chapter 2). It is still common practice to use BM25 (Robertson and Zaragoza, 2009) for this purpose—or more generally, sparse retrieval techniques—due to their low latency (realized via inverted index). Our answer selection models then used the retrieved candidates and ranked them according to how well they answer the given question. What if we could replace BM25 with a better method that takes semantic similarity into account? Can we improve the performance of the whole cQA pipeline with it?

This is the goal of Dense IR. One of the pioneering work in Dense IR has been done by Gillick et al. (2018) who learn dense vector representations of questions by averaging over learned word embeddings. For a cQA forum (Quora or AskUbuntu), they then pre-compute representations of all questions and perform a nearest neighbor search by calculating the cosine similarity between the query question and all pre-

[1] For instance, Ma et al. (2020) show that improved question generators do not lead to better retrieval results, and Liang et al. (2020) show that better sampling strategies in contrast do. However, what is a good synthetic query? Analyzing this in more depth seems critical to make future progress in the area. This is in particular due to the 220 million generated synthetic training examples of Liang et al. (2020), where one can suspect that now is a good time to move the focus from quantitative to qualitative considerations.

computed representations. With this procedure, they considerably outperformed BM25, presenting some evidence for the feasibility of dense IR. Importantly, this is related to our approach in Section 3.1, where we enable pre-computing all representations of answers within a large corpus, independently of the question (which is necessary for Dense IR).

This trend has been continued and several other works have proposed more effective sentence encoders for Dense IR (e.g., Yang et al., 2020; Karpukhin et al., 2020; Guo et al., 2020). However, it has also been shown that zero-shot transfer of such encoders to new datasets is very challenging (Liang et al., 2020), requiring large amounts of training instances before better performances than BM25 can be achieved (opposite to cross-encoders, see Chapter 7). How can we improve Dense IR in zero-shot transfer settings? Can we leverage large amounts of cQA forum data from different domains for learning more robust models? What is the impact of domain shift? What is the impact when transferring between different text types (e.g., tweets vs. long documents)? These represent interesting research questions that could complement the insights of our last thesis chapter, where we transferred cross-encoders between domains and tasks.

Intermediate pre-training. Intermediate pre-training of language models can yield large performance gains when finding compatible source and target tasks (Phang et al., 2018; Vu et al., 2020; Poth et al., 2021). This procedure may also benefit our cQA setting: we could pre-train a model on a pairwise similarity task such as STS (Cer et al., 2017) before fine-tuning it for answer selection or question similarity. With the availability of large model repositories (Wolf et al., 2020; Pfeiffer et al., 2020a) the *automatic* selection of suitable source tasks is gaining high practical relevance in NLP. We can now choose between an abundance of existing (intermediately) trained models, however, fine-tuning all of them on our target task (to find the best one) is computationally infeasible. Previous work leverages task embeddings (Vu et al., 2020) and general-purpose sentence embeddings (Poth et al., 2021) to automatically identify beneficial transfer relations. Even though both approaches are effective, important research questions remain: *why* are some pre-training tasks beneficial while others are not? Can we observe the same trends across different transformer architectures? Do existing approaches scale to large repositories with hundreds of pre-trained models?

In summary, we have now outlined four directions for future work related to cQA, which correlate with highly relevant topics in NLP: (1) model efficiency, (2) text generation, (3) dense IR, (4) intermediate pre-training. We believe that these topics will play central roles in the coming years and that cQA offers several interesting research settings to study them. Our work presented in this thesis can be widely extended and may serve as a starting point for related future work.

Appendix

A Data Handling

In accordance with DFG's "Principles for the Handling of Research Data".[2]

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 under the Apache 2.0 license (see the repositories for licensing details):

1. The experimental source-code to reproduce our experiments of Section 3.1: https://github.com/UKPLab/iwcs2017-answer-selection
2. The source code of our prototypical cQA system and experimental framework of Section 3.2: https://github.com/UKPLab/acl2017-non-factoid-qa
3. The code to induce our cross-lingual sentence embeddings of Section 4.1: https://github.com/UKPLab/arxiv2018-xling-sentence-embeddings
4. Instructions and code necessary to reproduce our experiments of Section 4.2: https://github.com/UKPLab/www19-xling-question-retrieval
5. The experimental source code to reproduce our experiments of Chapter 5: https://github.com/UKPLab/aaai2019-coala-cqa-answer-selection
6. The experimental source code to reproduce our experiments of Chapter 6: https://github.com/UKPLab/emnlp2019-duplicate_question_detection
7. The experimental source code to reproduce our experiments of Chapter 7: https://github.com/UKPLab/emnlp2020-multicqa

The following other data (binary, text) is archived long-term (for at least 10 years) by the Universitäts- und Landesbibliothek Darmstadt through their "TUDatalib" offering:

1. All translated data to reproduce our cross-lingual experiments in Section 4.1: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2352. This is a restricted dataset, meaning that it is not publicly accessible. The reason is that we cannot publish some of the translated versions of datasets due to licensing resrictions. We made a subset of this data publicly available:

[2] https://www.dfg.de/download/pdf/foerderung/antragstellung/forschungsdaten/richtlinien_forschungsdaten.pdf

https://github.com/UKPLab/arxiv2018-xling-sentence-embeddings/tree/master/data

2. The translations and synthetic training data of our cross-lingual experiments in Section 4.2: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2351 (License: Creative Commons Attribution Share-Alike 4.0).

3. The StackExchange answer selection datasets of Chapter 5: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2350 (License: Creative Commons Attribution Share-Alike 4.0).

The following other data (binary, text) is distributed via UKP Lab's public webserver (due to the large size of the data):

1. Our re-mapped cross-lingual word embeddings are available on: https://public.ukp.informatik.tu-darmstadt.de/arxiv2018-xling-sentence-embeddings/ (License: Creative Commons Attribution Share-Alike 4.0).

2. All our generated duplicates (including the source and intermediate data files) of Chapter 6 are available on: https://public.ukp.informatik.tu-darmstadt.de/emnlp2019-duplicate_question_detection/ (License: Creative Commons Attribution Share-Alike 4.0).

3. The model checkpoints of our MultiCQA models of Chapter 7 are available on: https://public.ukp.informatik.tu-darmstadt.de/rueckle/multicqa/ (License: Creative Commons Attribution Share-Alike 4.0).

All publications related to this thesis are publicly available on the ACL Anthology (aclweb.org/anthology/), the ACM digital library (https://dl.acm.org/), on the website of the Association for the Advancement of Artificial Intelligence (https://aaai.org/), or the ArXiv pre-print server (http://arxiv.org):

1. https://www.aclweb.org/anthology/W17-6935/.

2. https://www.aclweb.org/anthology/P17-4004/.

3. https://www.aclweb.org/anthology/D19-1171/.

4. https://www.aclweb.org/anthology/2020.emnlp-main.194/.

5. https://arxiv.org/abs/1803.01400.

6. https://dl.acm.org/doi/abs/10.1145/3308558.3313502.

7. https://ojs.aaai.org//index.php/AAAI/article/view/4671.

Moreover, all research results of the aforementioned publications are documented in the present thesis, which is archived by the Universitäts- und Landesbibliothek Darmstadt.

B Detailed Cross-Lingual Results of Concatenated Power Mean Word Embeddings

Parts of this appendix are quoted verbatim from our publication: Rücklé et al. (2018).

We report results for the individual language transfer across en→de and en→fr in Table B.1. The cross-lingual performance degradation (per-language) is shown in Figure B.1 (en→fr) and Figure B.2 (en→de).

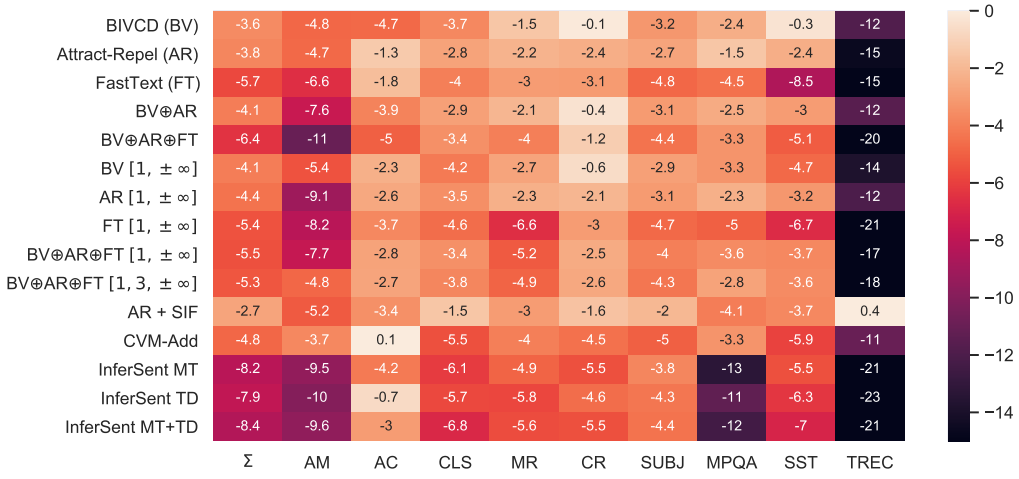

Figure B.1: The cross-lingual performance degradation for en→fr.

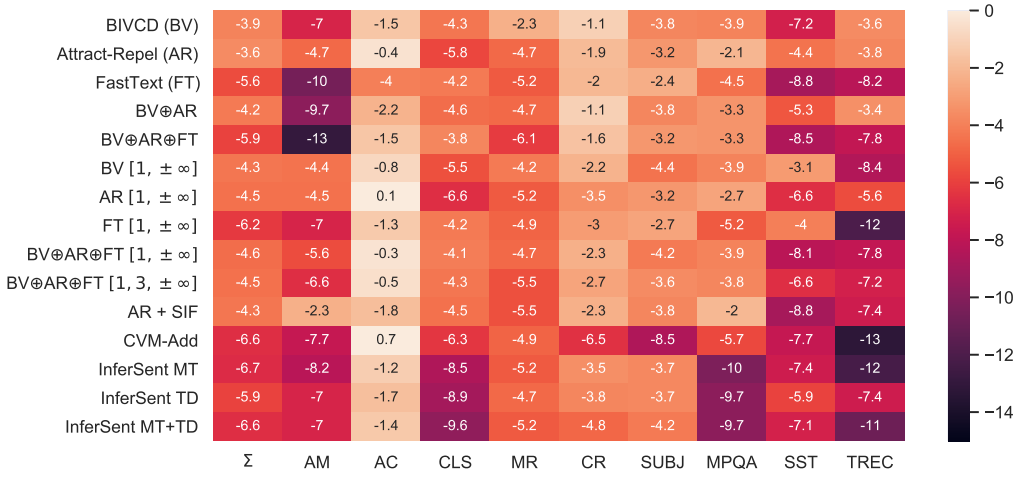

Figure B.2: Cross-lingual performance degradation for en→de.

Model	**Σ**		**AM**		**AC**		**CLS**		**MR**		**CR**		**SUBJ**		**MPQA**		**SST**		**TREC**	
Transfer Language	de	fr	de	fr	de	fr	de	fr	de	fr	de	fr	de	fr	de	fr	de	fr	de	fr
Arithmetic mean																				
BIVCD (BV)	65.5	68.1	39.2	41.9	68.9	66.4	65.0	67.7	62.2	66.5	70.6	72.9	79.8	82.4	79.8	83.3	61.2	70.2	72.2	61.8
Attract-Repel (AR)	69.4	68.9	39.0	38.2	**71.1**	66.5	67.4	70.4	66.6	69.7	73.7	74.0	81.8	83.8	84.0	84.8	71.3	73.6	69.6	59.4
FastText (FT)	68.7	68.0	36.9	40.0	63.8	62.9	70.1	70.0	68.3	69.9	73.9	72.3	86.3	84.0	81.7	81.3	69.5	69.2	67.8	62.4
BV⊕AR	70.9	**71.3**	**40.8**	**42.3**	70.0	**67.6**	69.4	70.9	67.2	70.5	75.4	76.5	83.3	84.8	84.1	**85.4**	**72.5**	**75.6**	**75.8**	**68.0**
BV⊕AR⊕FT	**71.9**	70.6	39.2	40.7	71.0	64.5	**71.2**	**72.1**	**69.8**	**70.8**	**76.6**	**76.9**	**86.8**	**85.7**	**84.6**	84.9	71.8	74.7	**75.8**	65.2
***p*-mean** [p-values]																				
BV $[1, \pm\infty]$	68.0	69.5	48.0	47.9	70.7	**66.8**	64.4	67.3	60.5	66.8	71.1	73.3	81.1	83.9	79.9	82.7	64.4	69.4	72.0	67.0
AR $[1, \pm\infty]$	71.3	71.0	45.7	42.7	70.5	65.1	67.1	70.3	67.4	70.2	75.3	75.7	83.7	84.9	84.0	84.8	71.2	74.9	76.6	70.4
FT $[1, \pm\infty]$	70.2	68.5	42.7	45.1	67.1	61.3	69.6	69.2	68.3	67.0	73.4	73.4	86.7	84.9	81.6	81.2	**74.4**	72.0	68.2	62.8
BV⊕AR⊕FT $[1, \pm\infty]$	73.7	72.7	50.8	49.6	**72.0**	66.6	**70.9**	**72.0**	**70.6**	70.2	**77.3**	76.1	86.6	86.8	84.1	84.9	73.4	**77.0**	77.6	71.0
BV⊕AR⊕FT $[1, 3, \pm\infty]$	**74.0**	**73.3**	**51.4**	**53.6**	**72.0**	66.3	70.8	71.5	70.5	**70.7**	77.1	**76.2**	**87.7**	**87.3**	**84.2**	**85.6**	74.1	76.9	**78.4**	**71.2**
Baselines																				
AR + SIF	67.5	68.7	40.2	36.5	70.3	65.1	**67.7**	**70.4**	66.2	69.2	73.7	73.8	80.0	83.2	**82.9**	80.5	68.0	71.9	58.4	68.0
CVM-Add	65.3	69.4	45.6	**49.9**	69.6	68.1	62.3	66.0	60.7	66.1	68.6	72.0	76.6	82.3	76.1	**82.5**	62.7	67.7	65.2	**70.4**
InferSent MT	71.8	**70.2**	50.3	48.3	70.9	68.7	67.0	68.9	69.3	69.2	**76.7**	75.8	84.4	**84.9**	77.9	74.9	**72.5**	**74.3**	77.4	67.2
InferSent TD	72.1	70.0	**52.7**	49.5	73.4	**70.6**	66.8	69.1	68.6	69.2	74.9	74.4	84.3	84.2	77.4	76.3	**72.5**	72.8	78.2	63.8
InferSent MT+TD	**72.4**	**70.2**	52.0	48.4	**73.5**	69.2	66.6	68.8	**69.5**	**69.7**	76.4	**76.0**	**84.7**	84.2	78.3	75.7	72.1	72.2	**78.8**	67.6

Table B.1: Individual cross-lingual results for the language transfer en→de and en→fr. Numbers in parentheses are the in-language results minus the given cross-language value..⊕denotes the concatenation of different embeddings (or p-means), brackets show the different p-means of the model.

C Cross-lingual Projection of Word Embeddings

Parts of this appendix are quoted verbatim from our publication: Rücklé et al. (2018).

Here we describe the conceptual and technical details to reproduce the results of our non-linear projection method that we use to map word embeddings of two languages into a shared embedding space (cf. §4.1.4).

Formalization We learn a projection of two embedding spaces $\mathbb{E}^l$ and $\mathbb{E}^k$ with dimensionality e and f, respectively, into a shared space of dimensionality d using two non-linear transformations:

$$f_l(\mathbf{x}_l) = \tanh\left(\mathbf{W}_l\mathbf{x}_l + \mathbf{b}_l\right)$$
$$f_k(\mathbf{x}_k) = \tanh\left(\mathbf{W}_k\mathbf{x}_k + \mathbf{b}_k\right)$$

where $\mathbf{x}_l \in \mathbb{R}^e$, $\mathbf{x}_k \in \mathbb{R}^f$ are original input embeddings and $\mathbf{W}_l \in \mathbb{R}^{d\times e}, \mathbf{W}_k \in \mathbb{R}^{d\times f}$, $\mathbf{b}_l \in \mathbb{R}^d, \mathbf{b}_k \in \mathbb{R}^d$ are parameters to be learned. Here $\mathbf{x}_l$ and $\mathbf{x}_k$ are monolingual representations.

For each sentence s and its translation t we randomly sample one unrelated sentence u from our data and obtain sentence representations $\mathbf{r}_s = f_l(\mathbf{x}_s)$, $\mathbf{r}_t = f_k(\mathbf{x}_t)$, and $\mathbf{r}_u = f_k(\mathbf{x}_u)$. We then optimize the following max-margin hinge loss:

$$\mathcal{L} = \max\left(0,\ m - \mathsf{sim}(\mathbf{r}_s, \mathbf{r}_t) + \mathsf{sim}(\mathbf{r}_s, \mathbf{r}_u)\right)$$

where sim is cosine similarity and m is the margin parameter. This objective moves embeddings of translations closer to and embeddings of random cross-lingual sentences further away from each other.

Training We use minibatched SGD with the Adam optimizer (Kingma and Ba, 2015) for training. We train on >130K bilingually aligned sentence pairs from the TED corpus (Hermann and Blunsom, 2014), which consists of translated transcripts from TED talks. Each sentence s is represented by its average (monolingual) word embedding, i.e., H_1.

We set the margin parameter to $m = 0.5$ as we have observed that higher values lead to a faster convergence. We furthermore randomly set 50% of the input embedding dimensions to zero during training (dropout).

Training of one epoch usually takes less than a minute in our TensorFlow implementation (on CPU), and convergence is usually achieved after less than 100 epochs.

Application Even though we learn our non-linear projection on the sentence level, we later apply it on the word level, i.e., we map individual word embeddings from each of two languages via $f_\psi(\mathbf{x}_\psi)$ where $\psi = l, k$. This is valid because average word embeddings live in the same space as individual word embeddings. The reason for doing so is that otherwise we would have to learn individual transformations for

Model	$\sum$ X-Ling	$\sum$ In-Language
FT (monolingual)	-	80.8
FT (CCA‡)	71.1	79.3
FT (our projection)	74.6	79.7
BV (orig)	70.9	75.8
BV (our projection)	71.0	74.6
AR (orig)	61.8	79.3
AR (our projection)	74.5	77.9

Table C.2: The downstream task performance of average word embeddings with our projection method in comparison to other approaches. ‡CCA was trained on word-alignments extracted from TED transcripts using fast_align (thus using the same data source as our method).

each of our power means, not only the average ($= H_1$), which would be too costly particularly when incorporating many different p-values. Working on the word-level, in general, also allows us to resort to word-level projection techniques using, e.g., word-alignments rather than sentence alignments.

However, in preliminary experiments, we found that our suggested approach produces considerably better cross-lingual word embeddings in our setup. Results are shown in Table C.2, where we report the performance of average word embeddings for cross-lingual en→de task transfer (averaged over MR, CR, SUBJ, MPQA, SST, TREC). Compared to the word-level projection method CCA we obtain substantially better cross-lingual sentence embeddings, and even stronger improvements when re-mapping AR embeddings, even though these are already bilingual.

D Data Filtering for Duplicate Question Generation

Parts of this appendix are quoted verbatim from our publication: Rücklé et al. (2019b).

Filtering and paragraph selection is necessary to obtain less noisy title-body pairs for QG. This consists of three steps: (1) filtering out questions that are not suitable for QG, (2) extracting paragraphs from bodies, and (3) only keeping one paragraph with the highest similarity to the title. The details are given below.

Filtering. We discard all questions that:

- contain bodies with less than 10 words,
- are downvoted, i.e., have a score on StackExchange that is below zero ("bad" questions).

Paragraph extraction. Some questions contain multiple long paragraphs, which is too much information to train suitable question generation or duplicate detection models. We thus extract paragraphs from the text to filter them in a later step.

In StackExchange platforms, users can freely add new lines, new paragraphs (the text then appears in HTML paragraph tags), lists, images, and code. This freedom results in many different ways of writing text. For instance, some users prefer to use paragraph tags and other users separate every sentence with a new-line character and all paragraphs with two or more new-line characters. Further, many users include code and enumerations in their questions.

This makes it difficult to extract actual paragraphs of the text. Thus, we first apply a preprocessing step to remove all HTML tags:

- We remove all code and images from the description.
- We then extract the text of each item from enumerations and append a new-line character.
- Likewise, we extract the text in paragraph tags and append a new-line character. We retain all new-line characters that appear in the paragraph.

We then analyze the new-line characters in the text to form the paragraphs for extraction. We read the input line-by-line:

- If the current line contains only one sentence it is merged with the previous paragraph.
- If the current line contains more than one sentence it is considered as a new paragraph.

Paragraph selection. After extracting N paragraphs $p_1 \dots p_N$ from the description, we select one paragraph according to $\underset{p_n}{\text{argmax}}\, f(p_n, \text{title}(q))$. The function f scores each p_n by calculating the maximum cosine similarity of a sentence s in p_n to

the question title title(q) using a sentence encoder (enc):

$$f(p_i, t) = \max_{s \in p_i} [\cos(\text{enc}(s), \text{enc}(t))]$$

In our experiments, enc corresponds to our concatenated power-mean word embeddings of Section 4.1. We calculate the maximum similarity of individual sentences to determine the semantic similarity independent of the paragraph length.

E Details on Our Zero-Shot Transfer Setup

Parts of this appendix are quoted verbatim from our publication: Rücklé et al. (2020b).

E.1 Computing Infrastructure

We used a heterogeneous cluster with different types of GPUs for our experiments. Our most demanding experiments with RoBERTa-large were performed with one NVIDIA Tesla V100 GPU and 32GB memory (per experiment). To train the models with a batchsize of 32, we used accumulation of gradients over two smaller mini-batches of size 16. One epoch with all source domains trains for on average 97 minutes. The remaining experiments were split across NVIDIA Tesla V100/P100 GPUs (32GB), and NVIDIA Titan RTX (24GB).

E.2 AdapterFusion (AF) on LAS-Academia and InsuranceQA

AdapterFusion learns a weighted combination of adapter outputs in each BERT layer, which is dependent on the layer input. Similar to Pfeiffer et al. (2021), we can thus plot the activations of the individual adapters for different benchmarks in order to analyze which source domains are most impactful. Further, this allows us to observe how the activations differ across different benchmarks.

In Figure E.3 and Figure E.4, we plot the activations for *AF balanced* on LAS-Academia and on InsuranceQA, which were the best and worst transfer datasets of this approach, respectively (compared to MT; see §7.4.2). We find that the activations are very similar across the two benchmarks, which indicates that our model learns to focus less on the model input. This shows that some adapters are better suited than others for individual BERT layers, e.g., the adapter for the "English" domain dominates layers 9 and 10, and "OpenSource" as well as "StackExchange" adapters dominate layer 11.

When transferring to the narrow expert domain InsuranceQA, interestingly, the same adapters are activated in BERT layers, with slightly different strengths as compared to LAS-Academia. This means that specific combinations of the same adapters are helpful for a variety of downstream tasks.

To investigate the impact of single most important adapters and how they affect the performance of AF, we *remove* the adapter of the *English* domain—which has the strongest activations in AF balanced—and plot the result for LAS-Academia in Figure E.5. We observe that AF, now increases the activation of the "Ell" (English language learners) adapter (see layer 9). This shows that AF has learned to utilize particular types of information encoded in adapters that exploit similar attributes, rather than combining a fixed selection of adapters. If, like in this scenario, the adapter is no longer available, AF extracts the information from other, similar adapters. This validates the effectiveness of AF as well as that different kinds of information are stored within the different layers of adapters.

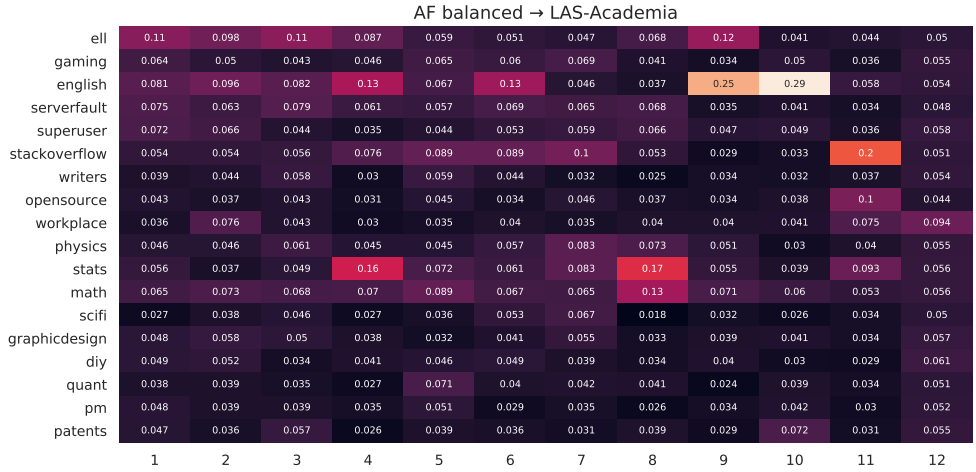

Figure E.3: Adapter activations in individual BERT layers for AF balanced when transferring to LAS-Academia.

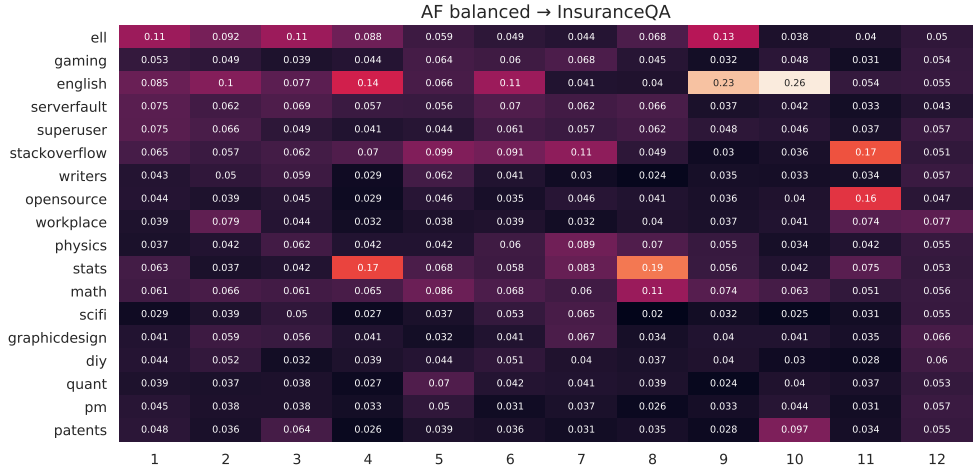

Figure E.4: Adapter activations in individual BERT layers for AF balanced when transferring to InsuranceQA.

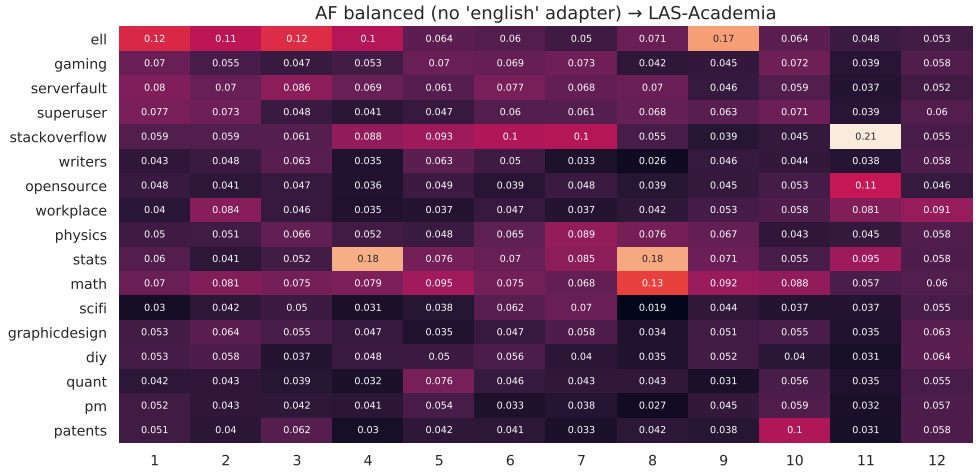

Figure E.5: Adapter activations in individual BERT layers for AF balanced (excluding the adapter from the "English" domain) when transferring to LAS-Academia.

E.3 List of Domains in Combination Experiments

The list of all domains is available on the web: https://stackexchange.com/sites. We list the domains used for our two subsets in §7.4 below.

Balanced contains the top-3 domains (according to the number of asked questions) within the six broad categories as defined by StackExchange (technology, culture/recreation, life/arts, science, professional, business). Included domains are:

ell (English language learners), gaming, english, serverfault, superuser, stackoverflow, writers, opensource, workplace, physics, stats, math, scifi, graphicdesign, diy (do-it-yourself), quant (quantitative finance), pm (project management), patents

Largest contains the top-18 largest domains (according to the number of asked questions). The included domains are:

stackoverflow, math, superuser, serverfault, latex, unix, physics, statistics, electronics, gis (geographic information systems), english, salesforce, wordpress, magento, sharepoint, gaming, dba (database administrators), drupal

List of Figures

List of Tables

Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng: 'TensorFlow: A System for Large-Scale Machine Learning', in: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2016)*, pp. 265–283, USENIX Association, 2016, Online: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman: 'Knowledge Sharing and Yahoo Answers: Everyone Knows Something', in: *Proceedings of the 17th International Conference on World Wide Web (WWW 2008)*, p. 665–674, Association for Computing Machinery, 2008, Online: https://doi.org/10.1145/1367497.1367587.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg: 'Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks', in: *5th International Conference on Learning Representations (ICLR 2017)*, 2017, Online: http://arxiv.org/abs/1608.04207.

Eugene Agichtein, David Carmel, Dan Pelleg, Yuval Pinter, and Donna Harman: 'Overview of the TREC 2015 LiveQA Track.', in: *Proceedings of the 2015 Text REtrieval Conference (TREC 2015)*, 2015, Online: https://trec.nist.gov/pubs/trec24/papers/Overview-QA.pdf.

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne: 'Finding High-Quality Content in Social Media', in: *Proceedings of the 2008 International Conference on Web Search and Data Mining*, p. 183–194, Association for Computing Machinery, 2008, Online: https://doi.org/10.1145/1341531.1341557.

Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer: 'ReQA: An Evaluation for End-to-End Answer Retrieval Models', in: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering (MRQA 2019)*, pp. 137–146, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-5819.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin: 'Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 3490–3496, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-1352.

David Allen and Tim D Wilson: 'Information overload: context and causes', *The New Review of Information Behaviour Research* 4 (1): 31–44, 2003.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma: 'A Simple but Tough-to-Beat Baseline for Sentence Embeddings', in: *5th International Conference on Learning Representations (ICLR 2017)*, 2017, Online: https://openreview.net/pdf?id=SyK00v5xx.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre: 'Learning principled bilingual mappings of word embeddings while preserving monolingual invariance', in: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pp. 2289–2294, Association for Computational Linguistics, 2016, Online: http://www.aclweb.org/anthology/D16-1250.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre: 'Learning bilingual word embeddings with (almost) no bilingual data', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 451–462, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/P17-1042.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre: 'A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 789–798, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/P18-1073.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio: 'Neural machine translation by jointly learning to align and translate', in: *3rd International Conference on Learning Representations (ICLR 2015)*, 2015, Online: https://arxiv.org/abs/1409.0473.

Ankur Bapna and Orhan Firat: 'Simple, Scalable Adaptation for Neural Machine Translation', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 1538–1548, Association for Computational Linguistics, Hong Kong, China, 2019, Online: https://www.aclweb.org/anthology/D19-1165.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin: 'A Neural Probabilistic Language Model', *Journal of machine learning research* 3: 1137–1155, 2003, Online: http://dl.acm.org/citation.cfm?id=944919.944966.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang: 'Semantic Parsing on Freebase from Question-Answer Pairs', in: *Proceedings of the 2013 Conference*

on Empirical Methods in Natural Language Processing (EMNLP 2013), pp. 1533–1544, Association for Computational Linguistics, 2013, Online: https://www.aclweb.org/anthology/D13-1160.

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal: 'Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding', in: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, p. 192–199, Association for Computing Machinery, 2000, Online: https://doi.org/10.1145/345508.345576.

Adam Berger and John Lafferty: 'Information Retrieval as Statistical Translation', in: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*, p. 222–229, Association for Computing Machinery, 1999, Online: https://doi.org/10.1145/312624.312681.

Delphine Bernhard and Iryna Gurevych: 'Combining Lexical Semantic Resources with Question & Answer Archives for Translation-Based Answer Finding', in: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-AFNLP 2009)*, pp. 728–736, Association for Computational Linguistics, 2009, Online: https://www.aclweb.org/anthology/P09-1082.

Nicola Bertoldi and Marcello Federico: 'Domain Adaptation for Statistical Machine Translation with Monolingual Resources', in: *Proceedings of the Fourth Workshop on Statistical Machine Translation (WMT 2009)*, pp. 182–189, Association for Computational Linguistics, 2009, Online: http://aclweb.org/anthology/W09-0432.

Steven Bird and Edward Loper: 'NLTK: The Natural Language Toolkit', in: *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pp. 214–217, Association for Computational Linguistics, 2004, Online: https://www.aclweb.org/anthology/P04-3031.

Mohan John Blooma, Alton Yeow-Kuan Chua, and Dion Hoe-Lian Goh: 'Selection of the Best Answer in CQA Services', in: *Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations*, p. 534–539, IEEE Computer Society, 2010, Online: https://doi.org/10.1109/ITNG.2010.127.

Dasha Bogdanova, Jennifer Foster, Daria Dzendzik, and Qun Liu: 'If You Can't Beat Them Join Them: Handcrafted Features Complement Neural Nets for Non-Factoid Answer Reranking', in: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pp. 121–131, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/E17-1012.

Florian Böhm, Yang Gao, Christian M. Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych: 'Better Rewards Yield Better Summaries: Learning to Summarise

Without References', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 3110–3120, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-1307.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov: 'Enriching Word Vectors with Subword Information', *Transactions of the Association of Computational Linguistics* 5: 135–146, 2017, Online: http://www.aclweb.org/anthology/Q17-1010.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia: 'Findings of the 2013 Workshop on Statistical Machine Translation', in: *Proceedings of the Eighth Workshop on Statistical Machine Translation (WMT 2013)*, pp. 1–44, Association for Computational Linguistics, 2013, Online: https://www.aclweb.org/anthology/W13-2201.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna: 'Findings of the 2014 Workshop on Statistical Machine Translation', in: *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT 2014)*, pp. 12–58, Association for Computational Linguistics, 2014, Online: https://www.aclweb.org/anthology/W14-3302.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz: 'Findings of the 2018 Conference on Machine Translation (WMT18)', in: *Proceedings of the Third Conference on Machine Translation (WMT 2018)*, pp. 272–303, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/W18-6401.

Daniele Bonadiman, Antonio Uva, and Alessandro Moschitti: 'Effective shared representations with Multitask Learning for Community Question Answering', in: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pp. 726–732, Association for Computational Linguistics, 2017, Online: http://aclweb.org/anthology/E17-2115.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning: 'A large annotated corpus for learning natural language inference', in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 632–642, Association for Computational Linguistics, 2015, Online: http://www.aclweb.org/anthology/D15-1075.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya

Sutskever, and Dario Amodei: 'Language Models are Few-Shot Learners', *arXiv preprint arXiv:2005.14165* 2020, Online: https://arxiv.org/abs/2005.14165.

Michael Bugert, Yevgeniy Puzikov, Andreas Rücklé, Judith Eckle-Kohler, Teresa Martin, Eugenio Martínez-Cámara, Daniil Sorokin, Maxime Peyrard, and Iryna Gurevych: 'LSDSem 2017: Exploring Data Generation Methods for the Story Cloze Test', in: *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem 2017)*, pp. 56–61, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/W17-0908.

Steven Cao, Nikita Kitaev, and Dan Klein: 'Multilingual Alignment of Contextual Word Representations', in: *8th International Conference on Learning Representations (ICLR 2020)*, 2020, Online: https://arxiv.org/pdf/2002.03518.pdf.

Xin Cao, Gao Cong, Bin Cui, Christian S. Jensen, and Quan Yuan: 'Approaches to Exploring Category Information for Question Retrieval in Community Question-Answer Archives', *ACM Transactions on Information Systems* 30 (2), 2012, Online: https://doi.org/10.1145/2180868.2180869.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia: 'SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation', in: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, pp. 1–14, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/S17-2001.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil: 'Universal Sentence Encoder for English', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2018)*, pp. 169–174, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/D18-2029.

Wen Chan, Xiangdong Zhou, Wei Wang, and Tat-Seng Chua: 'Community Answer Summarization for Multi-Sentence Question with Group L1 Regularization', in: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pp. 582–591, Association for Computational Linguistics, 2012, Online: https://www.aclweb.org/anthology/P12-1061.

Sarath Chandar, Mitesh M. Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha: 'Multilingual Deep Learning', *Deep Learning Workshop at NeurIPS* 2013.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar: 'Pre-training Tasks for Embedding-based Large-scale Retrieval', in: *8th International Conference on Learning Representations (ICLR 2020)*, 2020, Online: https://arxiv.org/abs/2002.03932.

Delphine Charlet and Géraldine Damnati: 'SimBow at SemEval-2017 Task 3: Soft-Cosine Semantic Similarity between Questions for Community Question Answering', in: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, pp. 315–319, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/S17-2051.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio: 'On the Properties of Neural Machine Translation: Encoder–Decoder Approaches', in: *Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST 2014)*, pp. 103–111, Association for Computational Linguistics, 2014a, Online: https://www.aclweb.org/anthology/W14-4012.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio: 'Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation', in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1724–1734, Association for Computational Linguistics, 2014b, Online: https://www.aclweb.org/anthology/D14-1179.

Minseok Cho, Gyeongbok Lee, and Seung-won Hwang: 'Explanatory and Actionable Debugging for Machine Learning: A TableQA Demonstration', in: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*, p. 1333–1336, Association for Computing Machinery, 2019, Online: https://doi.org/10.1145/3331184.3331404.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant: 'Coarse-to-Fine Question Answering for Long Documents', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 209–220, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/P17-1020.

Sumit Chopra, Michael Auli, and Alexander M. Rush: 'Abstractive Sentence Summarization with Attentive Recurrent Neural Networks', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pp. 93–98, 2016, Online: http://aclweb.org/anthology/N16-1012.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser et al.: 'Rethinking Attention with Performers', *9th International Conference on Learning Representations (ICLR 2021)* 2021, Online: https://openreview.net/pdf?id=Ua6zuk0WRH.

Chenhui Chu and Rui Wang: 'A Survey of Domain Adaptation for Neural Machine Translation', in: *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pp. 1304–1319, Association for Computational Linguistics, 2018, Online: http://aclweb.org/anthology/C18-1111.

Daniel Cohen and W. Bruce Croft: 'End to End Long Short Term Memory Networks for Non-Factoid Question Answering', in: *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR 2016)*, p. 143–146, Association for Computing Machinery, 2016, Online: https://doi.org/10.1145/2970398.2970438.

Daniel Cohen, Liu Yang, and W. Bruce Croft: 'WikiPassageQA: A Benchmark Collection for Research on Non-Factoid Answer Passage Retrieval', in: *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018)*, p. 1165–1168, Association for Computing Machinery, 2018, Online: https://doi.org/10.1145/3209978.3210118.

Ronan Collobert and Jason Weston: 'A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning', in: *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pp. 160–167, Association for Computing Machinery, 2008, Online: http://doi.acm.org/10.1145/1390156.1390177.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov: 'Unsupervised Cross-lingual Representation Learning at Scale', in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pp. 8440–8451, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.acl-main.747.

Alexis Conneau and Douwe Kiela: 'SentEval: An Evaluation Toolkit for Universal Sentence Representations', in: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association, 2018, Online: https://www.aclweb.org/anthology/L18-1269.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes: 'Supervised Learning of Universal Sentence Representations from Natural Language Inference Data', in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pp. 681–691, Association for Computational Linguistics, 2017, Online: http://www.aclweb.org/anthology/D17-1071.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov: 'XNLI: Evaluating Cross-lingual Sentence Representations', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pp. 2475–2485, Association for Computational Linguistics, 2018, Online: http://aclweb.org/anthology/D18-1269.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu: 'Attention-over-Attention Neural Networks for Reading Comprehension', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguis-*

tics (ACL 2017), pp. 593–602, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/P17-1055.

Giovanni Da San Martino, Alberto Barrón Cedeño, Salvatore Romeo, Antonio Uva, and Alessandro Moschitti: 'Learning to Re-Rank Questions in Community Question Answering Using Advanced Features', in: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016)*, p. 1997–2000, Association for Computing Machinery, 2016, Online: https://doi.org/10.1145/2983323.2983893.

Giovanni Da San Martino, Salvatore Romeo, Alberto Barroón-Cedeño, Shafiq Joty, Lluís Maàrquez, Alessandro Moschitti, and Preslav Nakov: 'Cross-Language Question Re-Ranking', in: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*, p. 1145–1148, Association for Computing Machinery, 2017, Online: https://doi.org/10.1145/3077136.3080743.

Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen: 'Joint Learning of Answer Selection and Answer Summary Generation in Community Question Answering', in: *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, pp. 7651–7658, 2020a, Online: https://arxiv.org/abs/1911.09801.

Yang Deng, Wenxuan Zhang, Yaliang Li, Min Yang, Wai Lam, and Ying Shen: 'Bridging Hierarchical and Sequential Context Modeling for Question-Driven Extractive Answer Summarization', in: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*, p. 1693–1696, Association for Computing Machinery, 2020b, Online: https://doi.org/10.1145/3397271.3401208.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova: '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 (NAACL 2019)*, pp. 4171–4186, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/N19-1423.

Yingqi Qu Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang: 'RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering', *arXiv preprint arXiv:2010.08191* 2020, Online: https://arxiv.org/abs/2010.08191.

William B. Dolan and Chris Brockett: 'Automatically Constructing a Corpus of Sentential Paraphrases', in: *Proceedings of the Third International Workshop on Paraphrasing (IWP 2005)*, 2005, Online: https://www.aclweb.org/anthology/I05-5002.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata: 'Learning to Paraphrase for Question Answering', in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pp. 875–886,

Association for Computational Linguistics, 2017, Online: http://aclweb.org/anthology/D17-1091.

Cícero Dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny: 'Learning Hybrid Representations to Retrieve Semantically Equivalent Questions', in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pp. 694–699, Association for Computational Linguistics, 2015, Online: https://www.aclweb.org/anthology/P15-2114.

Cicero Dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou: 'Attentive Pooling Networks', in: *arXiv preprint arXiv:1602.03609*, 2016, Online: https://arxiv.org/abs/1602.03609.

Xinya Du and Claire Cardie: 'Identifying Where to Focus in Reading Comprehension for Neural Question Generation', in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pp. 2067–2073, Association for Computational Linguistics, 2017, Online: http://aclweb.org/anthology/D17-1219.

Xinya Du and Claire Cardie: 'Harvesting Paragraph-level Question-Answer Pairs from Wikipedia', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 1907–1917, 2018, Online: http://aclweb.org/anthology/P18-1177.

Xinya Du, Junru Shao, and Claire Cardie: 'Learning to Ask: Neural Question Generation for Reading Comprehension', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 1342–1352, 2017, Online: http://aclweb.org/anthology/P17-1123.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner: 'DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs', in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*, pp. 2368–2378, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/N19-1246.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou: 'Question Generation for Question Answering', in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pp. 866–874, 2017, Online: http://aclweb.org/anthology/D17-1090.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier: 'Understanding Back-Translation at Scale', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pp. 489–500, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/D18-1045.

Steffen Eger, Johannes Daxenberger, Christian Stab, and Iryna Gurevych: 'Cross-lingual Argumentation Mining: Machine Translation (and a bit of Projection) is

All You Need!', in: *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pp. 831–844, Association for Computational Linguistics, 2018a, Online: https://www.aclweb.org/anthology/C18-1071.

Steffen Eger, Andreas Rücklé, and Iryna Gurevych: 'PD3: Better Low-Resource cross-lingual Transfer By Combining Direct Transfer and Annotation Projection', in: *Proceedings of the 5th Workshop on Argument Mining (ArgMin 2018)*, pp. 131–143, Association for Computational Linguistics, 2018b, Online: https://www.aclweb.org/anthology/W18-5216.

Steffen Eger, Andreas Rücklé, and Iryna Gurevych: 'Pitfalls in the Evaluation of Sentence Embeddings', in: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pp. 55–60, Association for Computational Linguistics, 2019a, Online: https://www.aclweb.org/anthology/W19-4308.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych: 'Text Processing Like Humans Do: Visually Attacking and Shielding NLP Systems', in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*, pp. 1634–1647, Association for Computational Linguistics, 2019b, Online: https://www.aclweb.org/anthology/N19-1165.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni: 'Paraphrase-Driven Learning for Open Question Answering', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 1608–1618, Association for Computational Linguistics, 2013, Online: https://www.aclweb.org/anthology/P13-1158.

Angela Fan, Edouard Grave, and Armand Joulin: 'Reducing Transformer Depth on Demand with Structured Dropout', in: *8th International Conference on Learning Representations (ICLR 2020)*, 2020, Online: https://openreview.net/forum?id=SylO2yStDr.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli: 'ELI5: Long Form Question Answering', in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pp. 3558–3567, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/P19-1346.

M. Amin Farajian, Marco Turchi, Matteo Negri, Nicola Bertoldi, and Marcello Federico: 'Neural vs. Phrase-Based Machine Translation in a Multi-Domain Scenario', in: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pp. 280–284, Association for Computational Linguistics, 2017, Online: http://aclweb.org/anthology/E17-2045.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith: 'Retrofitting Word Vectors to Semantic Lexicons', in: *Proceedings of the 2015 Conference of the North American Chapter of the Association for*

Computational Linguistics (NAACL 2015), pp. 1606–1615, Association for Computational Linguistics, 2015, Online: https://www.aclweb.org/anthology/N15-1184.

Manaal Faruqui and Chris Dyer: 'Improving Vector Space Word Representations Using Multilingual Correlation', in: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pp. 462–471, Association for Computational Linguistics, Gothenburg, Sweden, April 2014, Online: http://www.aclweb.org/anthology/E14-1049.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou: 'Applying Deep Learning to Answer Selection: A Study and An Open Task', in: *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 813–820, 2015, Online: https://arxiv.org/abs/1508.01585.

Wenzheng Feng, Yu Wu, Wei Wu, Zhoujun Li, and Ming Zhou: 'Beihang-MSRA at SemEval-2017 Task 3: A Ranking System with Neural Matching Features for Community Question Answering', in: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, pp. 280–286, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/S17-2045.

D. A. Ferrucci: 'Introduction to "This is Watson"', *IBM Journal of Research and Development* 56 (3.4): 1:1–1:15, 2012.

Pnina Fichman: 'A comparative assessment of answer quality on four question answering sites', *Journal of Information Science* 37 (5): 476–486, 2011, Online: https://doi.org/10.1177/0165551511415584.

Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili: 'KeLP at SemEval-2016 Task 3: Learning Semantic Relations between Questions and Answers', in: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, pp. 1116–1123, Association for Computational Linguistics, 2016, Online: http://aclweb.org/anthology/S16-1172.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen: 'MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension', in: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering (MRQA 2019)*, pp. 1–13, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-5801.

Jonathan Frankle and Michael Carbin: 'The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks', 2018, Online: https://openreview.net/pdf?id=rJl-b3RcF7.

Daniel Fried, Peter Jansen, Gustave Hahn-Powell, Mihai Surdeanu, and Peter Clark: 'Higher-order Lexical Semantic Models for Non-factoid Answer Reranking', *Transactions of the Association for Computational Linguistics* 3: 197–210, 2015, Online: https://www.aclweb.org/anthology/Q15-1015.

Wee Chung Gan and Hwee Tou Ng: 'Improving the Robustness of Question Answering Systems to Question Paraphrasing', in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6065–6075, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/P19-1610.

Lizheng Gao, Gang Zhou, and Junyong Luo: 'A BOW-Based Sentence Embedding Method for Chinese Event Identification', *Journal of Physics: Conference Series* 1486: 042012, apr 2020, Online: https://doi.org/10.1088/1742-6596/1486/4/042012.

Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min: 'On Making Reading Comprehension More Comprehensive', in: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering (MRQA 2019)*, pp. 105–112, Association for Computational Linguistics, 2019b, Online: https://www.aclweb.org/anthology/D19-5815.

Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min: 'Question Answering is a Format; When is it Useful?', *arXiv preprint arXiv:1909.11291* 2019a, Online: https://arxiv.org/abs/1909.11291.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti: 'TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection', in: *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, pp. 7780–7788, Association for the Advancement of Artificial Intelligence, 2020, Online: https://arxiv.org/abs/1911.04118.

Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin: 'A Convolutional Encoder Model for Neural Machine Translation', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 123–135, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/P17-1012.

Mor Geva, Ankit Gupta, and Jonathan Berant: 'Injecting Numerical Reasoning Skills into Language Models', in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pp. 946–958, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.acl-main.89.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant: 'Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies', *arXiv preprint arXiv:2101.02235* 2021, Online: https://arxiv.org/abs/2101.02235.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar: 'End-to-end retrieval in continuous space', *arXiv preprint arXiv:1811.08008* 2018, Online: https://arxiv.org/abs/1811.08008.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić: 'How to (Properly) Evaluate Cross-Lingual Word Embeddings: On Strong Baselines, Comparative Analyses, and Some Misconceptions', in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pp. 710–721, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/P19-1070.

Goran Glavaš and Ivan Vulić: 'Explicit Retrofitting of Distributional Word Vectors', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 34–45, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/P18-1004.

Goran Glavaš and Ivan Vulić: 'Non-Linear Instance-Based Cross-Lingual Mapping for Non-Isomorphic Embedding Spaces', in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pp. 7548–7555, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.acl-main.675.

Goran Glavaš and Ivan Vulicćtajner: 'Zero-Shot Language Transfer for Cross-Lingual Sentence Retrieval with the Bidirectional Attention Model', in: *Proceedings of the 41st European Conference on Information Retrieval (ECIR '19)*, pp. 523–538, Springer, April 2019, Online: https://doi.org/10.1007/978-3-030-15712-8_34.

Mandy Guo, Yinfei Yang, Daniel Cer, Qinlan Shen, and Noah Constant: 'MultiReQA: A Cross-Domain Evaluation for Retrieval Question Answering Models', *arXiv preprint arXiv:2005.02507* 2020, Online: https://arxiv.org/abs/2005.02507.

Deepak Gupta, Rajkumar Pujari, Asif Ekbal, Pushpak Bhattacharyya, Anutosh Maitra, Tom Jain, and Shubhashis Sengupta: 'Can Taxonomy Help? Improving Semantic Question Matching using Question Taxonomy', in: *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pp. 499–513, Association for Computational Linguistics, 2018, Online: http://aclweb.org/anthology/C18-1042.

Hojae Han, Seungtaek Choi, Haeju Park, and Seung-won Hwang: 'MICRON: Multigranular Interaction for Contextualizing RepresentatiON in Non-factoid Question Answering', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 5890–5895, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-1601.

G.H. Hardy, J.E. Littlewood, and G. Pólya: *Inequalities*, Cambridge University Press, Cambridge, England, 1952.

F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A. Konstan: 'Predictors of Answer Quality in Online Q&A Sites', in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 865–874, Association for

Computing Machinery, 2008, Online: https://doi.org/10.1145/1357054.1357191.

Sven Hartrumpf, Ingo Glöckner, and Johannes Leveling: 'Efficient Question Answering with Question Decomposition and Multiple Answer Streams', in: *Evaluating Systems for Multilingual and Multimodal Information Access*, pp. 421–428, Springer Berlin Heidelberg, 2009, Online: https://doi.org/10.1007/978-3-642-04447-2_49.

Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W Bruce Croft: 'ANTIQUE: A non-factoid question answering benchmark', in: *Proceedings of the 2020 European Conference on Information Retrieval (ECIR 2020)*, pp. 166–173, 2020, Online: https://arxiv.org/abs/1905.08957.

Hua He and Jimmy Lin: 'Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pp. 937–948, Association for Computational Linguistics, 2016, Online: https://www.aclweb.org/anthology/N16-1108.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun: 'Deep Residual Learning for Image Recognition', in: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR 2016)*, pp. 770–778, 2016, Online: https://arxiv.org/abs/1512.03385.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang: 'DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications', in: *Proceedings of the Workshop on Machine Reading for Question Answering (MRQA 2018)*, pp. 37–46, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/W18-2605.

Karl Moritz Hermann and Phil Blunsom: 'Multilingual Models for Compositional Distributed Semantics', in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pp. 58–68, Association for Computational Linguistics, 2014, Online: http://www.aclweb.org/anthology/P14-1006.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom: 'Teaching Machines to Read and Comprehend', in: *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS 2015)*, p. 1693–1701, 2015.

Jonathan Herzig and Jonathan Berant: 'Span-based Semantic Parsing for Compositional Generalization', *arXiv preprint arXiv:2009.06040* 2020, Online: https://arxiv.org/abs/2009.06040.

Ryuichiro Higashinaka and Hideki Isozaki: 'Corpus-based Question Answering for why-Questions', in: *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pp. 418–425, Association for Computational Linguistics, 2008, Online: http://aclweb.org/anthology/I08-1055.

Felix Hill, Kyunghyun Cho, and Anna Korhonen: 'Learning Distributed Representations of Sentences from Unlabelled Data', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pp. 1367–1377, 2016, Online: http://aclanthology.coli.uni-saarland.de/pdf/N/N16/N16-1162.pdf.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean: 'Distilling the Knowledge in a Neural Network', in: *NeurIPS Deep Learning and Representation Learning Workshop*, 2015, Online: http://arxiv.org/abs/1503.02531.

Sepp Hochreiter and Jürgen Schmidhuber: 'Long short-term memory', *Neural computation* 9 (8): 1735–1780, 1997.

Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin: 'CQADupStack: A Benchmark Data Set for Community Question-Answering Research', in: *Proceedings of the 20th Australasian Document Computing Symposium (ADCS 2015)*, Association for Computing Machinery, 2015, Online: https://doi.org/10.1145/2838931.2838934.

Doris Hoogeveen, Karin M Verspoor, and Timothy Baldwin: 'CQADupStack: Gold or Silver?', in: *Proceedings of the SIGIR 2016 Workshop on Web Question Answering Beyond Factoids (WebQA 2016)*, 2016.

Doris Hoogeveen, Li Wang, Timothy Baldwin, and Karin M. Verspoor: 'Web Forum Retrieval and Text Analytics: A Survey', *Foundations and Trends in Information Retrieval* 12 (1): 1–163, 2018, Online: https://doi.org/10.1561/1500000062.

Enamul Hoque, Shafiq Joty, Lluís Màrquez, Alberto Barrón-Cedeño, Giovanni Da San Martino, Alessandro Moschitti, Preslav Nakov, Salvatore Romeo, and Giuseppe Carenini: 'An Interactive System for Exploring Community Question Answering Forums', in: *Proceedings of the 26th International Conference on Computational Linguistics: System Demonstrations (COLING 2016)*, pp. 1–5, 2016, Online: https://www.aclweb.org/anthology/C16-2001.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzkebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly: 'Parameter-Efficient Transfer Learning for NLP', in: *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, pp. 2790–2799, 2019, Online: http://proceedings.mlr.press/v97/houlsby19a.html.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen: 'Convolutional Neural Network Architectures for Matching Natural Language Sentences', in: *Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS 2014)*, p. 2042–2050, MIT Press, 2014, Online: https://arxiv.org/abs/1503.03244.

Haifeng Hu, Bingquan Liu, Baoxun Wang, Ming Liu, and Xiaolong Wang: 'Multimodal DBN for Predicting High-Quality Answers in cQA portals', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 843–847, Association for Computational Linguistics, 2013, Online: https://www.aclweb.org/anthology/P13-2146.

Shengli Hu: 'Somm: Into the Model', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pp. 1153–1159, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/D18-1146.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston: 'Poly-Encoders: Transformer Architectures and Pre-Training Strategies for Fast and Accurate Multi-Sentence Scoring', in: *8th International Conference on Learning Representations (ICLR 2020)*, 2020.

Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Shrivastava, Manish Gupta, and Vasudeva Varma: 'Fermi at SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media using Sentence Embeddings', in: *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 611–616, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/S19-2109.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer: 'Adversarial Example Generation with Syntactically Controlled Paraphrase Networks', in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018)*, pp. 1875–1885, Association for Computational Linguistics, 2018, Online: http://aclweb.org/anthology/N18-1170.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang: 'Search-based Neural Structured Learning for Sequential Question Answering', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 1821–1831, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/P17-1167.

Peter Jansen, Mihai Surdeanu, and Peter Clark: 'Discourse Complements Lexical Semantics for Non-factoid Answer Reranking', in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pp. 977–986, Association for Computational Linguistics, 2014, Online: https://www.aclweb.org/anthology/P14-1092.

Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee: 'Finding Similar Questions in Large Question and Answer Archives', in: *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM 2005)*, p. 84–90, Association for Computing Machinery, 2005, Online: https://doi.org/10.1145/1099554.1099572.

Zongcheng Ji, Fei Xu, Bin Wang, and Ben He: 'Question-Answer Topic Model for Question Retrieval in Community Question Answering', in: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012)*, p. 2471–2474, Association for Computing Machinery, 2012, Online: https://doi.org/10.1145/2396761.2398669.

Thorsten Joachims: 'Optimizing Search Engines Using Clickthrough Data', in: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge*

Discovery and Data Mining (KDD 2002), p. 133–142, Association for Computing Machinery, 2002, Online: https://doi.org/10.1145/775047.775067.

Jeff Johnson, Matthijs Douze, and Hervé Jégou: 'Billion-scale similarity search with GPUs', *IEEE Transactions on Big Data* 2019, Online: https://arxiv.org/abs/1702.08734.

Shafiq Joty, Preslav Nakov, Lluís Màrquez, and Israa Jaradat: 'Cross-language Learning with Adversarial Neural Networks', in: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 226–237, Association for Computational Linguistics, 2017, Online: http://aclweb.org/anthology/K17-1024.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave: 'Loss in Translation: Learning Bilingual Word Mapping with a Retrieval Criterion', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pp. 2979–2984, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/D18-1330.

Nal Kalchbrenner and Phil Blunsom: 'Recurrent Continuous Translation Models', in: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 1700–1709, Association for Computational Linguistics, 2013, Online: https://www.aclweb.org/anthology/D13-1176.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei: 'Scaling Laws for Neural Language Models', *arXiv preprint arXiv:2001.08361* 2020, Online: https://arxiv.org/abs/2001.08361.

Mladen Karan and Jan Šnajder: 'Paraphrase-Focused Learning to Rank for Domain-Specific Frequently Asked Questions Retrieval', *Expert Systems with Applications: An International Journal* 91 (C): 418–433, 2018, Online: https://doi.org/10.1016/j.eswa.2017.09.031.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih: 'Dense Passage Retrieval for Open-Domain Question Answering', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 6769–6781, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.emnlp-main.550.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret: 'Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention', in: *International Conference on Machine Learning (ICML 2020)*, pp. 5156–5165, PMLR, 2020, Online: https://arxiv.org/abs/2006.16236.

Subhradeep Kayal and George Tsatsaronis: 'EigenSent: Spectral sentence embeddings using higher-order Dynamic Mode Decomposition', in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL*

2019), pp. 4536–4546, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/P19-1445.

Tom Kenter, Alexey Borisov, and Maarten de Rijke: 'Siamese CBOW: Optimizing Word Embeddings for Sentence Representations', in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pp. 941–951, Association for Computational Linguistics, 2016, Online: http://www.aclweb.org/anthology/P16-1089.

Yoon Kim: 'Convolutional Neural Networks for Sentence Classification', in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1746–1751, Association for Computational Linguistics, 2014, Online: https://www.aclweb.org/anthology/D14-1181.

Diederik P. Kingma and Jimmy Lei Ba: 'Adam: A Method for Stochastic Optimization', in: *3rd International Conference on Learning Representations (ICLR 2015)*, 2015, Online: https://arxiv.org/abs/1412.6980.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler: 'Skip-thought Vectors', in: *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS 2015)*, pp. 3294–3302, MIT Press, 2015, Online: https://arxiv.org/abs/1506.06726.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych: 'The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation', in: *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations (COLING 2018)*, pp. 5–9, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/C18-2002.

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych: 'From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains', in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pp. 6982–6993, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.acl-main.624.

Philipp Koehn: 'Europarl: A Parallel Corpus for Statistical Machine Translation', in: *In Proceedings of the tenth Machine Translation Summit*, pp. 79–86, 2005, Online: http://homepages.inf.ed.ac.uk/pkoehn/publications/europarl-mtsummit05.pdf.

Alexandros Komninos and Suresh Manandhar: 'Dependency Based Embeddings for Sentence Classification Tasks', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pp. 1490–1500, Association for Computational Linguistics, 2016, Online: http://www.aclweb.org/anthology/N16-1175.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky: 'Revealing the Dark Secrets of BERT', in: *Proceedings of the 2019 Conference on Empirical*

Methods in Natural Language Processing (EMNLP 2019), pp. 4365–4374, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-1445.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov: 'Natural Questions: A Benchmark for Question Answering Research', *Transactions of the Association for Computational Linguistics* 7: 452–466, 2019, Online: https://www.aclweb.org/anthology/Q19-1026.

Patrik Lambert, Holger Schwenk, Christophe Servan, and Sadaf Abdul-Rauf: 'Investigations on Translation Model Adaptation Using Monolingual Data', in: *Proceedings of the Sixth Workshop on Statistical Machine Translation (WMT 2011)*, pp. 284–293, Association for Computational Linguistics, 2011, Online: http://aclweb.org/anthology/W11-2132.

Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš: 'Common Sense or World Knowledge? Investigating Adapter-Based Knowledge Injection into Pretrained Transformers', in: *Proceedings of Deep Learning Inside Out: The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures (DeeLIO 2020)*, pp. 43–49, Association for Computational Linguistics, November 2020a, Online: https://www.aclweb.org/anthology/2020.deelio-1.5.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš: 'From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 4483–4499, Association for Computational Linguistics, 2020b, Online: https://www.aclweb.org/anthology/2020.emnlp-main.363.

Quoc V. Le and Tomas Mikolov: 'Distributed Representations of Sentences and Documents', in: *Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML 2014)*, pp. 1188–1196, Association for Computing Machinery, 2014, Online: http://dl.acm.org/citation.cfm?id=3044805.3045025.

Yann LeCun, Leon Bottou, Genevieve B. Orr, and Klaus Robert Müller: *Efficient BackProp*, pp. 9–50, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, Online: https://doi.org/10.1007/3-540-49430-8_2.

Gyeongbok Lee, Sungdong Kim, and Seung-won Hwang: 'QADiver: Interactive Framework for Diagnosing QA Models', in: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, pp. 9861–9862, 2019a, Online: https://doi.org/10.1609/aaai.v33i01.33019861.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova: 'Latent Retrieval for Weakly Supervised Open Domain Question Answering', in: *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics (ACL 2019), pp. 6086–6096, Association for Computational Linguistics, 2019b, Online: https://www.aclweb.org/anthology/P19-1612.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez: 'Semi-supervised Question Retrieval with Gated Convolutions', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pp. 1279–1289, Association for Computational Linguistics, 2016a, Online: https://www.aclweb.org/anthology/N16-1153.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez: 'Semi-supervised Question Retrieval with Gated Convolutions', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pp. 1279–1289, Association for Computational Linguistics, 2016b, Online: https://www.aclweb.org/anthology/N16-1153.

Omer Levy and Yoav Goldberg: 'Dependency-Based Word Embeddings', in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pp. 302–308, Association for Computational Linguistics, 2014, Online: http://www.aclweb.org/anthology/P14-2050.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer: 'BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension', in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pp. 7871–7880, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.acl-main.703.

Baichuan Li, Tan Jin, Michael R. Lyu, Irwin King, and Barley Mak: 'Analyzing and Predicting Question Quality in Community Question Answering Services', in: *Proceedings of the 21st International Conference on World Wide Web (WWW 2012)*, p. 775–782, Association for Computing Machinery, 2012, Online: https://doi.org/10.1145/2187980.2188200.

Hongyu Li, Xiyuan Zhang, Yibing Liu, Yiming Zhang, Quan Wang, Xiangyang Zhou, Jing Liu, Hua Wu, and Haifeng Wang: 'D-NET: A Pre-Training and Fine-Tuning Framework for Improving the Generalization of Machine Reading Comprehension', in: *Proceedings of the 2nd Workshop on Machine Reading for Question Answering (MRQA 2019)*, pp. 212–219, Association for Computational Linguistics, 2019a, Online: https://www.aclweb.org/anthology/D19-5828.

Peng Li, Wei Li, Zhengyan He, Xuguang Wang, Ying Cao, Jie Zhou, and Wei Xu: 'Dataset and Neural Recurrent Sequence Labeling Model for Open-Domain Factoid Question Answering', 2016, Online: http://arxiv.org/abs/1607.06275.

Zeyu Li, Jyun-Yu Jiang, Yizhou Sun, and Wei Wang: 'Personalized Question Routing via Heterogeneous Network Embedding', in: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, pp. 192–199, 2019b, Online: https://doi.org/10.1609/aaai.v33i01.3301192.

Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang: 'Embedding-based Zero-shot Retrieval through Query Generation', *arXiv preprint arXiv:2009.10270* 2020, Online: https://arxiv.org/abs/2009.10270.

Martin Lichtblau: 'User-Centered Evaluation of End-to-End cQA Systems', *Master's Thesis, Computer Science Dpt., Technische Universitat Darmstadt* 2020.

Chuan-Jie Lin and Yu-Min Kuo: 'Description of the NTOU Complex QA System.', in: *Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access (NTCIR-8)*, pp. 47–54, 2010.

Zhouhan Lin, Minwei Feng, Cicero Nogueira Dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio: 'A Structured Self-attentive Sentence Embedding', in: *5th International Conference on Learning Representations (ICLR 2017)*, 2017, Online: http://arxiv.org/abs/1703.03130.

Robert Litschko, Goran Glavaš, Ivan Vulic, and Laura Dietz: 'Evaluating Resource-Lean Cross-Lingual Embedding Models in Unsupervised Retrieval', in: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*, p. 1109–1112, Association for Computing Machinery, 2019, Online: https://doi.org/10.1145/3331184.3331324.

Shusen Liu, Tao Li, Zhimin Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer: 'Visual Interrogation of Attention-Based Models for Natural Language Inference and Machine Comprehension', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2018)*, pp. 36–41, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/D18-2007.

Yandong Liu and Eugene Agichtein: 'On the Evolution of the Yahoo! Answers QA Community', in: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, p. 737–738, Association for Computing Machinery, 2008, Online: https://doi.org/10.1145/1390334.1390478.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang: 'Learning Natural Language Inference using Bidirectional LSTM model and Inner-Attention', *arXiv preprint arXiv:1605.09090* 2016, Online: https://arxiv.org/abs/1605.09090.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov: 'RoBERTa: A Robustly Optimized BERT Pretraining Approach', *arXiv preprint arXiv:1907.11692* 2019, Online: https://arxiv.org/abs/1907.11692.

Minh-Thang Luong and Christopher D Manning: 'Stanford Neural Machine Translation Systems for Spoken Language Domains', in: *Proceedings of the International Conference on Spoken Language Translation (IWSLT 2015)*, pp. 76–79, 2015.

Thang Luong, Hieu Pham, and Christopher D. Manning: 'Bilingual Word Representations with Monolingual Quality in Mind', in: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 151–159, Association for Computational Linguistics, 2015b, Online: https://www.aclweb.org/anthology/W15-1521.

Thang Luong, Hieu Pham, and Christopher D. Manning: 'Effective Approaches to Attention-based Neural Machine Translation', in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 1412–1421, Association for Computational Linguistics, 2015a, Online: https://www.aclweb.org/anthology/D15-1166.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald: 'Zero-shot Neural Retrieval via Domain-targeted Synthetic Query Generation', *arXiv preprint arXiv:2004.14503* 2020, Online: https://arxiv.org/abs/2004.14503.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang: 'Universal Text Representation from BERT: An Empirical Study', *arXiv preprint arXiv:1910.07973* 2019, Online: https://arxiv.org/abs/1910.07973.

Sean MacAvaney, Andrew Yates, Kai Hui, and Ophir Frieder: 'Content-Based Weak Supervision for Ad-Hoc Re-Ranking', in: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*, p. 993–996, Association for Computing Machinery, 2019, Online: https://doi.org/10.1145/3331184.3331316.

Yosi Mass, Haggai Roitman, Shai Erera, Or Rivlin, Bar Weiner, and David Konopnicki: 'A Study of BERT for Non-Factoid Question-Answering under Passage Length Constraints', *arXiv preprint arXiv:1908.06780* 2019, Online: https://arxiv.org/abs/1908.06780.

Julian McAuley and Alex Yang: 'Addressing Complex and Subjective Product-Related Queries with Customer Reviews', in: *Proceedings of the 25th International Conference on World Wide Web (WWW 2016)*, p. 625–635, 2016, Online: https://doi.org/10.1145/2872427.2883044.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher: 'The Natural Language Decathlon: Multitask Learning as Question Answering', in: *arXiv preprint arXiv:1806.08730*, 2018, Online: https://arxiv.org/abs/1806.08730.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean: 'Efficient Estimation of Word Representations in Vector Space', *arXiv preprint arXiv:1301.3781* 2013a, Online: http://arxiv.org/abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean: 'Distributed Representations of Words and Phrases and their Compositionality', *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2013)* pp. 3111–3119, 2013b, Online: http://arxiv.org/abs/1310.4546.

Aditya Mogadala and Achim Rettinger: 'Bilingual Word Embeddings from Parallel and Non-parallel Corpora for Cross-Language Text Classification', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pp. 692–702, Association for Computational Linguistics, 2016, Online: https://www.aclweb.org/anthology/N16-1083.

Nafise Sadat Moosavi, Angela Fan, Vered Shwartz, Goran Glavaš, Shafiq Joty, Alex Wang, and Thomas Wolf (Eds.): *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.sustainlp-1.0.

Nafise Sadat Moosavi, Prasetya Ajie Utama, Andreas Rücklé, and Iryna Gurevych: 'Improving Generalization by Incorporating Coverage in Natural Language Inference', *arXiv preprint arXiv:1909.08940* 2019, Online: https://arxiv.org/abs/1909.08940.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young: 'Semantic Specialisation of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints', *Transactions of the Association of Computational Linguistics* 5: 309–324, 2017, Online: http://www.aclweb.org/anthology/Q17-1022.

Makoto Nakatsuji: 'Can AI Generate Love Advice?: Toward Neural Answer Generation for Non-Factoid Questions', *arXiv preprint arXiv:1912.10163* 2019, Online: https://arxiv.org/abs/1912.10163.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor: 'SemEval-2017 Task 3: Community Question Answering', in: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, pp. 27–48, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/S17-2003.

Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree: 'SemEval-2015 Task 3: Answer Selection in Community Question Answering', in: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 269–281, Association for Computational Linguistics, 2015, Online: https://www.aclweb.org/anthology/S15-2047.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree: 'SemEval-2016

Task 3: Community Question Answering', in: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, pp. 525–545, Association for Computational Linguistics, 2016, Online: https://www.aclweb.org/anthology/S16-1083.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu: 'Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction', in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pp. 454–459, Association for Computational Linguistics, 2016a, Online: https://www.aclweb.org/anthology/P16-2074.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng: 'MS MARCO: A Human Generated MAchine Reading COmprehension Dataset', in: *arXiv preprint arXiv:1611.09268*, 2016b, Online: https://arxiv.org/abs/1611.09268.

Massimo Nicosia and Alessandro Moschitti: 'Accurate Sentence Matching with Hybrid Siamese Networks', in: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM 2017)*, p. 2235–2238, Association for Computing Machinery, 2017, Online: https://doi.org/10.1145/3132847.3133156.

Adi Omari, David Carmel, Oleg Rokhlenko, and Idan Szpektor: 'Novelty Based Ranking of Human Answers for Community Questions', in: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*, p. 215–224, Association for Computing Machinery, 2016, Online: https://doi.org/10.1145/2911451.2911506.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi: 'Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features', in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018)*, pp. 528–540, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/N18-1049.

Martha Palmer, Daniel Gildea, and Paul Kingsbury: 'The Proposition Bank: An Annotated Corpus of Semantic Roles', *Computational Linguistics* 31 (1): 71–106, 2005, Online: https://www.aclweb.org/anthology/J05-1004.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit: 'A Decomposable Attention Model for Natural Language Inference', in: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pp. 2249–2255, Association for Computational Linguistics, 2016, Online: https://www.aclweb.org/anthology/D16-1244/.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig: 'Bilingual Lexicon Induction with Semi-supervision in Non-Isometric Embedding Spaces', in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pp. 184–193, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/P19-1018.

Andreas Peldszus and Manfred Stede: 'An Annotated Corpus of Argumentative Microtexts', in: *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation*, pp. 801–815, Lisbon, Portugal, 2015.

Jeffrey Pennington, Richard Socher, and Christopher Manning: 'GloVe: Global Vectors for Word Representation', in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1532–1543, Association for Computational Linguistics, 2014, Online: https://www.aclweb.org/anthology/D14-1162.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych: 'AdapterFusion: Non-Destructive Task Composition for Transfer Learning', in: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, pp. 487–503, Association for Computational Linguistics, 2021, Online: https://www.aclweb.org/anthology/2021.eacl-main.39.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych: 'AdapterHub: A Framework for Adapting Transformers', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020)*, pp. 46–54, Association for Computational Linguistics, 2020a, Online: https://www.aclweb.org/anthology/2020.emnlp-demos.7.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder: 'MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 7654–7673, Association for Computational Linguistics, 2020b, Online: https://www.aclweb.org/anthology/2020.emnlp-main.617.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder: 'UNKs Everywhere: Adapting Multilingual Language Models to New Scripts', *arXiv preprint* 2020c, Online: https://arxiv.org/pdf/2012.15562.pdf.

Hieu Pham, Thang Luong, and Christopher Manning: 'Learning Distributed Representations for Multilingual Text Sequences', in: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 88–94, Association for Computational Linguistics, 2015, Online: https://www.aclweb.org/anthology/W15-1512.

Jason Phang, Thibault Févry, and Samuel R Bowman: 'Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks', *arXiv preprint arXiv:1811.01088* 2018, Online: https://arxiv.org/abs/1811.01088.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier: 'Monolingual Adapters for Zero-Shot Neural Machine Translation', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 4465–4470, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.emnlp-main.361.

Nina Poerner and Hinrich Schütze: 'Multi-View Domain Adapted Sentence Embeddings for Low-Resource Unsupervised Duplicate Question Detection', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 1630–1641, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-1173.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban: 'Investigating Backtranslation in Neural Machine Translation', *Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT 2018)* pp. 249–258, 2018, Online: http://arxiv.org/abs/1804.06189.

Jay M. Ponte and W. Bruce Croft: 'A Language Modeling Approach to Information Retrieval', in: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, p. 275–281, Association for Computing Machinery, 1998, Online: https://doi.org/10.1145/290941.291008.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych: 'What to Pre-Train on? Efficient Intermediate Task Selection', *arXiv preprint arXiv:2104.08247* 2021, Online: https://arxiv.org/abs/2104.08247.

Horst Pöttker: 'News and its communicative quality: the inverted pyramid—when and why did it appear?', *Journalism Studies* 4: 501–511, 11 2003.

Peter Prettenhofer and Benno Stein: 'cross-lingual Adaptation using Structural Correspondence Learning', in: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 1118–1127, Association for Computational Linguistics, 2010, Online: http://www.aclweb.org/anthology/P10-1114.

Yevgeniy Puzikov and Iryna Gurevych: 'E2E NLG Challenge: Neural Models vs. Templates', in: *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 463–471, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/W18-6557.

Xipeng Qiu and Xuanjing Huang: 'Convolutional Neural Tensor Network Architecture for Community-based Question Answering', in: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 15)*, pp. 1305–1311, 2015, Online: http://dl.acm.org/citation.cfm?id=2832415.2832431.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu: 'Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer', *Journal of Machine Learning Research* 21 (140): 1–67, 2020, Online: https://arxiv.org/abs/1910.10683.

Pranav Rajpurkar, Robin Jia, and Percy Liang: 'Know What You Don't Know: Unanswerable Questions for SQuAD', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 784–789, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/P18-2124.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang: 'SQuAD: 100,000+ Questions for Machine Comprehension of Text', in: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pp. 2383–2392, Association for Computational Linguistics, 2016, Online: https://www.aclweb.org/anthology/D16-1264.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy: 'Few-Shot Question Answering by Pretraining Span Selection', *arXiv preprint arXiv:2101.00438* 2021, Online: https://arxiv.org/abs/2101.00438.

Jinfeng Rao, Hua He, and Jimmy Lin: 'Noise-Contrastive Estimation for Answer Selection with Deep Neural Networks', in: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016)*, p. 1913–1916, Association for Computing Machinery, 2016, Online: https://doi.org/10.1145/2983323.2983872.

Sudha Rao and Hal Daumé III: 'Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 2737–2746, Association for Computational Linguistics, 2018a, Online: https://www.aclweb.org/anthology/P18-1255.

Sudha Rao and Hal Daumé III: 'Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 2737–2746, Association for Computational Linguistics, 2018b, Online: http://aclweb.org/anthology/P18-1255.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi: 'Learning multiple visual domains with residual adapters', in: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2017)*, pp. 506–516, 2017, Online: https://arxiv.org/abs/1705.08045.

Nils Reimers and Iryna Gurevych: 'Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 3982–3992, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-1410.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu: 'Statistical Machine Translation for Query Expansion in Answer Retrieval', in: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pp. 464–471, Association for Computational Linguistics, 2007, Online: https://www.aclweb.org/anthology/P07-1059.

Stephen Robertson and Hugo Zaragoza: 'The Probabilistic Relevance Framework: BM25 and Beyond', *Foundations and Trends in Information Retrieval* 3 (4): 333–389, 2009, Online: https://doi.org/10.1561/1500000019.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom: 'Reasoning about Entailment with Neural Attention', in: *4th International Conference on Learning Representations (ICLR 2016)*, 2016, Online: https://arxiv.org/abs/1509.06664.

Salvatore Romeo, Giovanni Da San Martino, Alberto Barrón-Cedeño, and Alessandro Moschitti: 'A Flexible, Efficient and Accurate Framework for Community Question Answering Pipelines', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL 2018)*, pp. 134–139, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/P18-4023.

Salvatore Romeo, Giovanni Da San Martino, Alberto Barrón-Cedeño, Alessandro Moschitti, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Mitra Mohtarami, and James Glass: 'Neural Attention for Learning to Rank Questions in Community Question Answering', in: *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, pp. 1734–1745, 2016, Online: https://www.aclweb.org/anthology/C16-1163.

Sascha Rothe and Hinrich Schütze: 'AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes', in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pp. 1793–1803, Association for Computational Linguistics, 2015, Online: https://www.aclweb.org/anthology/P15-1173.

Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych: 'Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations', *arXiv preprint arXiv:1803.01400* 2018, Online: https://arxiv.org/abs/1803.01400.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych: 'AdapterDrop: On the Efficiency of Adapters in Transformers', *arXiv preprint arXiv:2010.11918* 2020a, Online: https://arxiv.org/abs/2010.11918.

Andreas Rücklé and Iryna Gurevych: 'End-to-End Non-Factoid Question Answering with an Interactive Visualization of Neural Attention Weights', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL 2017)*, pp. 19–24, Association for Computational Linguistics, 2017b, Online: https://www.aclweb.org/anthology/P17-4004.

Andreas Rücklé and Iryna Gurevych: 'Real-Time News Summarization with Adaptation to Media Attention', in: *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP 2017)*, pp. 610–617, 2017c, Online: https://doi.org/10.26615/978-954-452-049-6_079.

Andreas Rücklé and Iryna Gurevych: 'Representation Learning for Answer Selection with LSTM-Based Importance Weighting', in: *12th International Conference on Computational Semantics (IWCS 2017)*, 2017a, Online: https://www.aclweb.org/anthology/W17-6935.

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych: 'COALA: A Neural Coverage-Based Approach for Long Answer Selection with Small Data.', in: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, pp. 6932–6939, Association for the Advancement of Artificial Intelligence, 2019a, Online: https://aaai.org/ojs/index.php/AAAI/article/view/4671/4549.

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych: 'Neural Duplicate Question Detection without Labeled Training Data', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 1607–1617, Association for Computational Linguistics, 2019b, Online: https://www.aclweb.org/anthology/D19-1171.

Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych: 'MultiCQA: Zero-Shot Transfer of Self-Supervised Text Matching Models on a Massive Scale', in: *Proceedings of The 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 2471–2486, Association for Computational Linguistics, 2020b, Online: https://www.aclweb.org/anthology/2020.emnlp-main.194.

Andreas Rücklé, Krishnkant Swarnkar, and Iryna Gurevych: 'Improved cross-lingual Question Retrieval for Community Question Answering', in: *The World Wide Web Conference (WWW 2019)*, pp. 3179–3186, Association for Computing Machinery, 2019c, Online: http://doi.acm.org/10.1145/3308558.3313502.

Sebastian Ruder: 'NLPs ImageNet moment has arrived', *The Gradient* 2018, Online: https://thegradient.pub/nlp-imagenet/.

Alexander M. Rush, Sumit Chopra, and Jason Weston: 'A Neural Attention Model for Abstractive Sentence Summarization', in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 379–389, 2015, Online: http://aclweb.org/anthology/D15-1044.

Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek: 'Image Classification with the Fisher Vector: Theory and Practice', *International Journal of Computer Vision* 105 (3): 222–245, 2013, Online: https://doi.org/10.1007/s11263-013-0636-x.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf: 'DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter', *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing* 2019, Online: http://arxiv.org/abs/1910.01108.

Timo Schick and Hinrich Schütze: 'Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference', *arXiv preprint arXiv:2001.07676* 2020, Online: https://arxiv.org/abs/2001.07676.

Fabian David Schmidt, Markus Dietsche, Simone Paolo Ponzetto, and Goran Glavaš: 'SEAGLE: A Platform for Comparative Evaluation of Semantic Encoders for Information Retrieval', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2019)*, pp. 199–204, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-3034.

Sebastian Schuster and Christopher D. Manning: 'Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks', in: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 16)*, pp. 2371–2378, European Language Resources Association, 2016, Online: https://www.aclweb.org/anthology/L16-1376.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni: 'Green AI', *arXiv preprint arXiv:1907.10597* 2019, Online: https://arxiv.org/abs/1907.10597.

Holger Schwenk and Matthijs Douze: 'Learning Joint Multilingual Sentence Representations with Neural Machine Translation', in: *Proceedings of the 2nd Workshop on Representation Learning for NLP (Repl4NLP 2017)*, pp. 157–167, Association for Computational Linguistics, 2017, Online: http://www.aclweb.org/anthology/W17-2619.

Abigail See, Peter J. Liu, and Christopher D. Manning: 'Get To The Point: Summarization with Pointer-Generator Networks', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 1073–1083, 2017, Online: http://aclweb.org/anthology/P17-1099.

Rico Sennrich, Barry Haddow, and Alexandra Birch: 'Improving Neural Machine Translation Models with Monolingual Data', in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pp. 86–96, Association for Computational Linguistics, 2016, Online: http://aclweb.org/anthology/P16-1009.

Aliaksei Severyn and Alessandro Moschitti: 'Structural Relationships for Large-scale Learning of Answer Re-ranking', in: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012)*, pp. 741–750, Association for Computing Machinery, 2012, Online: http://doi.acm.org/10.1145/2348283.2348383.

Aliaksei Severyn and Alessandro Moschitti: 'Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks', in: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2015)*, pp. 373–382, Association for Computing Machinery, 2015, Online: http://doi.acm.org/10.1145/2766462.2767738.

Chirag Shah and Jefferey Pomerantz: 'Evaluating and Predicting Answer Quality in Community QA', in: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 411–418, Association for Computing Machinery, 2010, Online: https://doi.org/10.1145/1835449.1835518.

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov: 'Adversarial Domain Adaptation for Duplicate Question Detection', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pp. 1056–1063, Association for Computational Linguistics, 2018, Online: http://aclweb.org/anthology/D18-1131.

John Shawe-Taylor and Nello Cristianini: *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

Libin Shen and Aravind K. Joshi: 'Ranking and Reranking with Perceptron', *Machine Learning* 60 (1-3): 73–96, 2005, Online: https://doi.org/10.1007/s10994-005-0918-9.

Yikang Shen, Wenge Rong, Zhiwei Sun, Yuanxin Ouyang, and Zhang Xiong: 'Question/Answer Matching for CQA System via Combining Lexical and Sequential Information', in: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, p. 275–281, 2015, Online: https://dl.acm.org/doi/10.5555/2887007.2887046.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh: 'AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 4222–4235, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.emnlp-main.346.

Amirreza Shirani, Bowen Xu, David Lo, Thamar Solorio, and Amin Alipour: 'Question Relatedness on Stack Overflow: The Task, Dataset, and Corpus-inspired Models', *AAAI 2019 Reasoning for Complex Question Answering Workshop* 2019, Online: https://arxiv.org/abs/1905.01966.

Grigori Sidorov, Alexander Gelbukh, Helena Gómez-Adorno, and David Pinto: 'Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model', *Computación y Sistemas* 18 (3): 491–504, 2014.

Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández: 'Syntactic Dependency-Based N-grams as Classification Features', in: *Proceedings of the 2012 Mexican International Conference on Artificial Intelligence (MICAI 2012)*, pp. 1–11, 2012.

Edwin Simpson, Yang Gao, and Iryna Gurevych: 'Interactive Text Ranking with Bayesian Optimization: A Case Study on Community QA and Summarization', *Transactions of the Association for Computational Linguistics* 8: 759–775, 2020, Online: https://doi.org/10.1162/tacl_a_00344.

Hongya Song, Zhaochun Ren, Shangsong Liang, Piji Li, Jun Ma, and Maarten de Rijke: 'Summarizing Answers in Non-Factoid Community Question-Answering', in: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM 2017)*, p. 405–414, Association for Computing Machinery, 2017a, Online: https://doi.org/10.1145/3018661.3018704.

Linfeng Song, Zhiguo Wang, and Wael Hamza: 'A Unified Query-based Generative Model for Question Generation and Question Answering', *arXiv preprint arXiv:1709.01058* 2017b, Online: http://arxiv.org/abs/1709.01058.

Daniil Sorokin and Iryna Gurevych: 'Modeling Semantics with Gated Graph Neural Networks for Knowledge Base Question Answering', in: *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pp. 3306–3317, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/C18-1280.

Ivan Srba and Maria Bielikova: 'A Comprehensive Survey and Classification of Approaches for Community Question Answering', *ACM Transactions on the Web* 10 (3), 2016, Online: https://doi.org/10.1145/2934687.

Christian Stab and Iryna Gurevych: 'Parsing Argumentation Structures in Persuasive Essays', *Computational Linguistics* 43 (3): 619–659, 2017, Online: https://www.aclweb.org/anthology/J17-3005.

Asa Cooper Stickland and Iain Murray: 'BERT and PALs: Projected Attention Layers for Efficient Adaptation in Multi-Task Learning', in: *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, pp. 5986–5995, 2019, Online: https://arxiv.org/abs/1902.02671.

Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M Rush: 'Seq2seq-Vis: A Visual Debugging Tool for Sequence-to-Sequence Models', *IEEE Transactions on Visualization and Computer Graphics* 25 (1): 353–363, 2018, Online: https://ieeexplore.ieee.org/document/8494828.

Emma Strubell, Ananya Ganesh, and Andrew McCallum: 'Energy and Policy Considerations for Deep Learning in NLP', in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pp. 3645–3650, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/P19-1355.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal: 'Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning', in: *6th International Conference on Learning Representations (ICLR 2018)*, 2018a, Online: https://arxiv.org/abs/1804.00079.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio: 'Neural Models for Key Phrase Extraction and Question Generation', in: *Proceedings of the Workshop on Machine Reading for Question Answering (MRQA 2018)*, pp. 78–88, 2018b, Online: http://aclweb.org/anthology/W18-2609.

Sai Praneeth Suggu, Kushwanth Naga Goutham, Manoj K. Chinnakotla, and Manish Shrivastava: 'Hand in Glove: Deep Feature Fusion Network Architectures for Answer Quality Prediction in Community Question Answering', in: *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, pp. 1429–1440, 2016, Online: https://www.aclweb.org/anthology/C16-1135.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang: 'ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding.', in: *Proceedings of the 34th AAAI Conference on Artificial Intelligence*

(AAAI 2020), pp. 8968–8975, 2020a, Online: https://arxiv.org/abs/1907.12412.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou: 'MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices', in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pp. 2158–2170, 2020b, Online: https://www.aclweb.org/anthology/2020.acl-main.195/.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza: 'Learning to Rank Answers on Large Online QA Collections', in: *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics (ACL 2008)*, pp. 719–727, Association for Computational Linguistics, 2008, Online: https://www.aclweb.org/anthology/P08-1082.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza: 'Learning to Rank Answers to Non-Factoid Questions from Web Collections', *Computational Linguistics* 37 (2): 351–383, 2011, Online: https://www.aclweb.org/anthology/J11-2003.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le: 'Sequence to Sequence Learning with Neural Networks', in: *Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS 2014)*, p. 3104–3112, MIT Press, 2014.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi: 'Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 9275–9293, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.emnlp-main.746.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich: 'Going Deeper with Convolutions', in: *Computer Vision and Pattern Recognition (CVPR 2015)*, 2015, Online: http://arxiv.org/abs/1409.4842.

Alon Talmor and Jonathan Berant: 'The Web as a Knowledge-Base for Answering Complex Questions', in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018)*, pp. 641–651, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/N18-1059.

Alon Talmor and Jonathan Berant: 'MultiQA: An Empirical Investigation of Generalization and Transfer in Reading Comprehension', in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pp. 4911–4921, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/P19-1485.

Ming Tan, Cicero Dos Santos, Bing Xiang, and Bowen Zhou: 'Improved Representation Learning for Question Answer Matching', in: *Proceedings of the 54th*

Annual Meeting of the Association for Computational Linguistics (ACL 2016), pp. 464–473, Association for Computational Linguistics, 2016, Online: http://aclweb.org/anthology/P16-1044.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou: 'LSTM-based Deep Learning Models for Non-factoid Answer Selection', *arXiv preprint arXiv:1511.04108* 2015, Online: https://arxiv.org/abs/1511.04108.

Duyu Tang, Nan Duan, Tao Qin, and Ming Zhou: 'Question Answering and Question Generation as Dual Tasks', *arXiv preprint arXiv:1706.02027* abs/1706.02027, 2017, Online: http://arxiv.org/abs/1706.02027.

Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui: 'Learning to Rank Question Answer Pairs with Holographic Dual LSTM Architecture', in: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*, p. 695–704, Association for Computing Machinery, 2017, Online: https://doi.org/10.1145/3077136.3080790.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui: 'Hyperbolic Representation Learning for Fast and Efficient Neural Question Answering', in: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM 2018)*, p. 583–591, Association for Computing Machinery, 2018a, Online: https://doi.org/10.1145/3159652.3159664.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui: 'Multi-Cast Attention Networks', in: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2018)*, p. 2299–2308, Association for Computing Machinery, 2018b, Online: https://doi.org/10.1145/3219819.3220048.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych: 'Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks', *arXiv preprint arXiv:2010.08240* 2020, Online: https://arxiv.org/abs/1905.01969.pdf.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych: 'BEIR: Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models', *arXiv preprint arxiv:2104.08663* 2021, Online: https://arxiv.org/abs/2104.08663.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman: 'NewsQA: A Machine Comprehension Dataset', in: *Proceedings of the 2nd Workshop on Representation Learning for NLP (Repl4NLP 2017)*, pp. 191–200, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/W17-2623.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos et al.: 'An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition', *BMC bioinformatics* 16 (1): 1–28, 2015.

Joseph Turian, Lev Ratinov, and Yoshua Bengio: 'Word Representations: A Simple and General Method for Semi-supervised Learning', in: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 384–394, Association for Computational Linguistics, 2010, Online: https://www.aclweb.org/anthology/P10-1040.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth: 'Cross-lingual Models of Word Embeddings: An Empirical Comparison', in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pp. 1661–1670, Association for Computational Linguistics, 2016, Online: http://www.aclweb.org/anthology/P16-1157.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord: 'UDapter: Language Adaptation for Truly Universal Dependency Parsing', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 2302–2315, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.emnlp-main.180.

Antonio Uva, Daniele Bonadiman, and Alessandro Moschitti: 'Injecting Relational Structural Representation in Neural Networks for Question Similarity', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 285–291, Association for Computational Linguistics, 2018, Online: http://aclweb.org/anthology/P18-2046.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit: 'Tensor2Tensor for Neural Machine Translation', *arXiv preprint arXiv:1803.07416* 2018, Online: https://arxiv.org/abs/1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin: 'Attention Is All You Need', in: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2017)*, pp. 5998–6008, 2017, Online: https://arxiv.org/abs/1706.03762.

Jana Vatter: 'Exploring Training Strategies for cQA Retrieval Tasks', *Bachelor's Thesis, Computer Science Dpt., Technische Universitat Darmstadt* 2019.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen: 'Evaluating Discourse-Based Answer Extraction for Why-Question Answering', in: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, p. 735–736, Association for Computing Machinery, 2007, Online: https://doi.org/10.1145/1277741.1277883.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen: 'What Is Not in the Bag of Words for Why-QA?', *Computational Linguistics* 36 (2): 229–245, 2010, Online: https://www.aclweb.org/anthology/J10-2003.

M. Vidoni, Ivan Vulić, and Goran Glavavs: 'Orthogonal Language and Task Adapters in Zero-Shot Cross-Lingual Transfer', in: *arXiv preprint arXiv:2012.06460*, 2020, Online: https://arxiv.org/pdf/2012.06460.pdf.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer: 'Exploring and Predicting Transferability across NLP Tasks', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 7882–7926, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.emnlp-main.635.

Ivan Vulić, Goran Glavaš, Nikola Mrkšić, and Anna Korhonen: 'Post-Specialisation: Retrofitting Vectors of Words Unseen in Lexical Resources', in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018)*, pp. 516–527, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/N18-1048.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen: 'Do We Really Need Fully Unsupervised Cross-Lingual Embeddings?', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 4407–4418, Association for Computational Linguistics, 2019, Online: https://www.aclweb.org/anthology/D19-1449.

Ivan Vulić and Marie-Francine Moens: 'Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction', in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pp. 719–725, Association for Computational Linguistics, 2015, Online: http://www.aclweb.org/anthology/P15-2118.

Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen: 'Morph-fitting: Fine-Tuning Word Vector Spaces with Simple Language-Specific Rules', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 56–68, Association for Computational Linguistics, 2017, Online: http://www.aclweb.org/anthology/P17-1006.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman: '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*, pp. 353–355, Association for Computational Linguistics, Brussels, Belgium, November 2018, Online: https://www.aclweb.org/anthology/W18-5446.

Bin Wang, Fenxiao Chen, Yuncheng Wang, and C-C Jay Kuo: 'Efficient Sentence Embedding via Semantic Subspace Analysis', *arXiv preprint arXiv:2002.09620* 2020a, Online: https://arxiv.org/abs/2002.09620.

Bingning Wang, Kang Liu, and Jun Zhao: 'Inner Attention based Recurrent Neural Networks for Answer Selection', in: *Proceedings of the 54th Annual Meeting of*

the Association for Computational Linguistics (ACL 2016), pp. 1288–1297, Association for Computational Linguistics, 2016a, Online: http://aclweb.org/anthology/P16-1122.

Di Wang and Eric Nyberg: 'A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering', in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pp. 707–712, Association for Computational Linguistics, 2015, Online: http://aclweb.org/anthology/P15-2116.

Kai Wang and Tat-Seng Chua: 'Exploiting Salient Patterns for Question Detection and Question Retrieval in Community-based Question Answering', in: *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pp. 1155–1163, 2010, Online: https://www.aclweb.org/anthology/C10-1130.

Kai Wang, Zhao-Yan Ming, Xia Hu, and Tat-Seng Chua: 'Segmentation of Multi-Sentence Questions: Towards Effective Question Retrieval in CQA Services', in: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*, p. 387–394, Association for Computing Machinery, 2010, Online: https://doi.org/10.1145/1835449.1835515.

Kai Wang, Zhaoyan Ming, and Tat-Seng Chua: 'A Syntactic Tree Matching Approach to Finding Similar Questions in Community-Based Qa Services', in: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 187–194, Association for Computing Machinery, 2009, Online: https://doi.org/10.1145/1571941.1571975.

Pengwei Wang, Yong Zhang, Lei Ji, Jun Yan, and Lianwen Jin: 'Concept Embedded Convolutional Semantic Model for Question Retrieval', in: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM 2017)*, pp. 395–403, 2017a, Online: http://doi.acm.org/10.1145/3018661.3018687.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou et al.: 'K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters', *arXiv preprint arXiv:2002.01808* 2020b, Online: https://arxiv.org/abs/2002.01808.

Shuohang Wang and Jing Jiang: 'A Compare-Aggregate Model for Matching Text Sequences', in: *5th International Conference on Learning Representations (ICLR 2017)*, 2017, Online: https://openreview.net/pdf?id=HJTzHtqee.

Tong Wang, Xingdi Yuan, and Adam Trischler: 'A Joint Model for Question Answering and Question Generation', *Learning to Generate Natural Language Workshop* 2017b, Online: http://arxiv.org/abs/1706.01450.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah: 'Sentence Similarity Learning by Lexical Decomposition and Composition', in: *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, pp. 1340–1349, 2016b, Online: https://www.aclweb.org/anthology/C16-1127.

Johann Wiedmeier: 'Enhanced Representation Learning for Question Retrieval with Transfer Learning', *Master's Thesis, Computer Science Dpt., Technische Universitat Darmstadt* 2017.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu: 'From Paraphrase Database to Compositional Paraphrase Model and Back', *Transactions of the Association of Computational Linguistics* 3: 345–358, 2015, Online: http://www.aclweb.org/anthology/Q15-1025.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu: 'Towards Universal Paraphrastic Sentence Embeddings', in: *4th International Conference on Learning Representations (ICLR 2016)*, 2016, Online: http://arxiv.org/abs/1511.08198.

John Wieting, Jonathan Mallinson, and Kevin Gimpel: 'Learning Paraphrastic Sentence Embeddings from Back-Translated Bitext', in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pp. 274–285, Association for Computational Linguistics, 2017, Online: http://aclweb.org/anthology/D17-1026.

Adina Williams, Nikita Nangia, and Samuel Bowman: '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 (NAACL 2018)*, pp. 1112–1122, Association for Computational Linguistics, 2018, Online: https://www.aclweb.org/anthology/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush: 'Transformers: State-of-the-Art Natural Language Processing', in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020)*, pp. 38–45, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant: 'Break It Down: A Question Understanding Benchmark', *Transactions of the Association for Computational Linguistics* 8: 183–198, 2020, Online: https://www.aclweb.org/anthology/2020.tacl-1.13.

Mingzhu Wu, Nafise Sadat Moosavi, Andreas Rücklé, and Iryna Gurevych: 'Improving QA Generalization by Concurrent Modeling of Multiple Biases', in: *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 839–853,

Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.findings-emnlp.74.

Wei Wu, Xu Sun, and Houfeng Wang: 'Question Condensing Networks for Answer Selection in Community Question Answering', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 1746–1755, 2018, Online: https://www.aclweb.org/anthology/P18-1162.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power: 'End-to-End Neural Ad-Hoc Ranking with Kernel Pooling', in: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*, p. 55–64, Association for Computing Machinery, 2017, Online: https://doi.org/10.1145/3077136.3080809.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk: 'Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval', *arXiv preprint arXiv:2007.00808* 2020, Online: https://arxiv.org/abs/2007.00808.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio: 'Show, Attend and Tell: Neural Image Caption Generation with Visual Attention', in Francis Bach and David Blei (Eds.): *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, pp. 2048–2057, PMLR, 2015, Online: http://proceedings.mlr.press/v37/xuc15.html.

Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft: 'Retrieval Models for Question and Answer Archives', in: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, p. 475–482, Association for Computing Machinery, 2008, Online: https://doi.org/10.1145/1390334.1390416.

Yi Yang, Wen-tau Yih, and Christopher Meek: 'WikiQA: A Challenge Dataset for Open-Domain Question Answering', in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 2013–2018, Association for Computational Linguistics, 2015, Online: http://aclweb.org/anthology/D15-1237.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil: 'Multilingual Universal Sentence Encoder for Semantic Retrieval', in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL 2020)*, pp. 87–94, Association for Computational Linguistics, 2020, Online: https://www.aclweb.org/anthology/2020.acl-demos.12.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le: 'XLNet: Generalized Autoregressive Pretraining for Language Understanding', in: *Proceedings of the Advances in Neural Information Processing*

Systems (NeurIPS 2019), pp. 5753–5763, 2019a, Online: https://arxiv.org/abs/1906.08237.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen: 'Semi-Supervised QA with Generative Domain-Adaptive Nets', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 1040–1050, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/P17-1096.

Ziyi Yang, Chenguang Zhu, and Weizhu Chen: 'Parameter-free Sentence Embedding via Orthogonal Basis', in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, pp. 638–648, Association for Computational Linguistics, 2019b, Online: https://www.aclweb.org/anthology/D19-1059.

Wen-Tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak: 'Question Answering Using Enhanced Lexical Semantic Models', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 1744–1753, Association for Computational Linguistics, 2013, Online: http://aclweb.org/anthology/P13-1171.

Wenpeng Yin and Hinrich Schütze: 'Task-Specific Attentive Pooling of Phrase Alignments Contributes to Sentence Matching', in: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pp. 699–709, Association for Computational Linguistics, 2017, Online: https://www.aclweb.org/anthology/E17-1066.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou: 'ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs', *Transactions of the Association for Computational Linguistics* 4: 259–272, 2016, Online: https://www.aclweb.org/anthology/Q16-1019.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le: 'QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension', in: *6th International Conference on Learning Representations (ICLR 2018)*, 2018.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman: 'Deep Learning for Answer Sentence Selection', in: *NeurIPS Deep Learning Workshop*, 2014, Online: http://arxiv.org/abs/1412.1632.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler: 'Machine Comprehension by Text-to-Text Neural Question Generation', in: *Proceedings of the 2nd Workshop on Representation Learning for NLP (Repl4NLP 2017)*, pp. 15–25, Association for Computational Linguistics, 2017, Online: http://aclweb.org/anthology/W17-2603.

Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou: 'Question Retrieval with High Quality Answers in Community Question Answering', in: *Proceedings*

of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM 2014), CIKM '14, p. 371–380, Association for Computing Machinery, 2014, Online: https://doi.org/10.1145/2661829.2661908.

Kaitao Zhang, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu: 'Selective Weak Supervision for Neural Information Retrieval', in: *Proceedings of The Web Conference 2020 (WWW 2020)*, p. 474–485, Association for Computing Machinery, 2020, Online: https://doi.org/10.1145/3366423.3380131.

Minghua Zhang and Yunfang Wu: 'An Unsupervised Model with Attention Autoencoders for Question Retrieval', in: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, pp. 4978–4986, 2018, Online: https://arxiv.org/abs/1803.03476.

Wei Emma Zhang, Quan Z. Sheng, Jey Han Lau, and Ermyas Abebe: 'Detecting Duplicate Posts in Programming QA Communities via Latent Semantics and Association Rules', in: *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*, p. 1221–1229, 2017a, Online: https://doi.org/10.1145/3038912.3052701.

Xiaodong Zhang, Sujian Li, Lei Sha, and Houfeng Wang: 'Attentive Interactive Neural Networks for Answer Selection in Community Question Answering', in: *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, p. 3525–3531, AAAI Press, 2017b, Online: https://dl.acm.org/citation.cfm?id=3298080.

Ye Zhang, Stephen Roller, and Byron C. Wallace: 'MGNC-CNN: A Simple Approach to Exploiting Multiple Word Embeddings for Sentence Classification', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pp. 1522–1527, Association for Computational Linguistics, 2016a, Online: http://www.aclweb.org/anthology/N16-1178.

Yuan Zhang, David Gaddy, Regina Barzilay, and Tommi Jaakkola: 'Ten Pairs to Tag – Multilingual POS Tagging via Coarse Mapping between Embeddings', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pp. 1307–1317, Association for Computational Linguistics, 2016b, Online: https://www.aclweb.org/anthology/N16-1156.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke: 'Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pp. 3901–3910, 2018, Online: http://aclweb.org/anthology/D18-1424.

Chunting Zhou, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad: 'Detecting Hallucinated Content in Conditional Neural

Sequence Generation', *arXiv preprint arXiv:2011.02593* 2020, Online: https://arxiv.org/abs/2011.02593.

Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu: 'Phrase-Based Translation Model for Question Retrieval in Community Question Answer Archives', in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pp. 653–662, Association for Computational Linguistics, 2011, Online: https://www.aclweb.org/anthology/P11-1066.

Guangyou Zhou, Yubo Chen, Daojian Zeng, and Jun Zhao: 'Towards Faster and Better Retrieval Models for Question Search', in: *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013)*, p. 2139–2148, Association for Computing Machinery, 2013, Online: https://doi.org/10.1145/2505515.2505550.

Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu: 'Learning Continuous Word Embedding with Metadata for Question Retrieval in Community Question Answering', in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pp. 250–259, Association for Computational Linguistics, 2015, Online: https://www.aclweb.org/anthology/P15-1025.

Guangyou Zhou, Yin Zhou, Tingting He, and Wensheng Wu: 'Learning Semantic Representation with Neural Networks for Community Question Answering Retrieval', *Knowledge-Based Systems* 93 (C): 75–83, 2016a, Online: https://doi.org/10.1016/j.knosys.2015.11.002.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou: 'Neural Question Generation from Text: A Preliminary Study', in: *National CCF Conference on Natural Language Processing and Chinese Computing*, pp. 662–671, Springer, 2017, Online: https://arxiv.org/abs/1704.01792.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao: 'Cross-Lingual Sentiment Classification with Bilingual Document Representation Learning', in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pp. 1403–1412, Association for Computational Linguistics, 2016b, Online: https://www.aclweb.org/anthology/P16-1133.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych: 'A Multi-Dimensional Model for Assessing the Quality of Answers in Social Q&A Sites', *Proceedings of 14th International Conference on Information Quality (ICIQ 2009)* 1: 264–265, 2009.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen: 'The United Nations Parallel Corpus v1.0', in: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 3530–3534, European Language Resources Association, May 2016, Online: https://www.aclweb.org/anthology/L16-1561.