MultiCQA: Zero-Shot Transfer of Self-Supervised Text Matching Models on a Massive Scale

Andreas Rücklé and **Jonas Pfeiffer** and **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab (UKP)
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

Abstract

We study the zero-shot transfer capabilities of text matching models on a massive scale, by self-supervised training on 140 source domains from community question answering forums in English. We investigate the model performances on nine benchmarks of answer selection and question similarity tasks, and show that *all* 140 models transfer surprisingly well, where the large majority of models substantially outperforms common IR baselines. We also demonstrate that considering a broad selection of source domains is crucial for obtaining the best zero-shot transfer performances, which contrasts the standard procedure that merely relies on the largest and most similar domains. In addition, we extensively study how to best combine multiple source domains. We propose to incorporate self-supervised with supervised multi-task learning on all available source domains. Our best zero-shot transfer model considerably outperforms *in-domain* BERT and the previous state of the art on six benchmarks. Fine-tuning of our model with in-domain data results in additional large gains and achieves the new state of the art on all nine benchmarks.

1 Introduction

Semantic matching of two text sequences is crucial among a wide range of NLP problems, such as question answering (Nakov et al., 2017; Wang and Jiang, 2017) or semantic textual similarity (Cer et al., 2017). Due to the ubiquity of applications, it is crucial to study how to obtain re-usable text matching models that transfer well to *unseen* domains or tasks.

Zero-shot transfer of text matching models is particularly challenging in setups of non-factoid answer selection (Cohen et al., 2018; Tay et al., 2017; Feng et al., 2015; Verberne et al., 2010) and question similarity (Nakov et al., 2017; Lei et al., 2016). These tasks compare questions and answers, or two potentially related questions in community question answering (cQA) forums, FAQ pages, and general collections of text passages. In contrast to other text matching tasks in NLP, they compare texts of different lengths—e.g., answers can be long explanations or descriptions—and often deal with expert domains. This makes it difficult to transfer models across domains (Shah et al., 2018), and to apply common approaches such as universal sentence embeddings without further domain or task adaptations (Poerner and Schütze, 2019).

Non-factoid answer selection and question similarity are also particularly *promising* to study zero-shot transfer. Reasons are that (1) there exist a large number of domains, and (2) in-domain training data is often scarce. Previous work proposed domain adaptation techniques (Poerner and Schütze, 2019; Shah et al., 2018), training with unlabeled data (Rücklé et al., 2019b), and shallow architectures (Rücklé et al., 2019a). However, these approaches result in entirely separate models that are specialized to individual target domains. One model that is re-usable and targets zero-shot transfer in similar settings is the question-answer encoder of Yang et al. (2020), which has recently been evaluated in cross-domain settings for efficient answer sentence retrieval (Guo et al., 2020). However, they do not study zero-shot transfer with a *large* number of source domains, and they do not assess how to best combine them.

In this work, we address these limitations and are—to the best of our knowledge—the first to study the zero-shot transfer capabilities of re-usable text matching models with a large number of source domains in these challenging setups.

In the **first part**, we investigate the zero-shot transfer capabilities of 140 domain-specific text matching models to nine benchmark datasets. By leveraging self-supervised training signals of ques-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2471–2486,
November 16–20, 2020.

tion title-body pairs, we analyze a large number of models specialized on diverse domains. We utilize the training method provided by Rücklé et al. (2019b) 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. In our extensive analysis, we show that our approach for zero-shot transfer is extremely effective—on six benchmarks *all* 140 models outperform common IR baselines. Most importantly, we revisit and analyze the traditional strategy of leveraging large data sets from intuitively similar domains to train models for zero-shot transfer. We establish that *neither* training data size *nor* domain similarity are suitable for predicting the best models, stressing the need for more elaborate strategies to identify suitable training tasks. 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 work, we study how to best *combine* multiple source domains with multi-task learning and AdapterFusion (Pfeiffer et al., 2020a). 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 imbalances, see Figure 1—is the most effective and outperforms the respective best of 140 single-domain models on six out of nine bechmarks. 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, as well as the previous state of the art on six benchmarks, which demonstrates its versatility across tasks and domains. We also show that our model is an effective initialization for in-domain fine-tuning, which results in large gains and achieves state-of-the-art results on all nine benchmarks.

Our source code and the weights of our best multi-task model is publicly available.[1] Additionally, all 140 source domain adapters are available at AdapterHub.ml (Pfeiffer et al., 2020b).

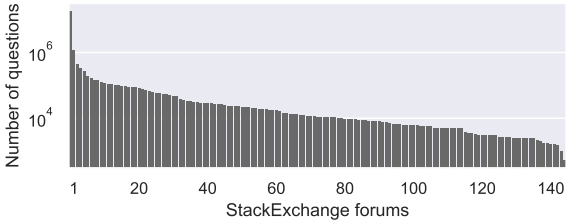

Figure 1: Number of questions in StackExchange forums (log scale) that can be used for self-supervision.

2 Related Work

The predominant method for text matching tasks such as non-factoid answer selection and question similarity is to train a neural architecture on a large quantity of labeled in-domain data. This includes CNN and LSTM models with attention (Tan et al., 2016; Dos Santos et al., 2016; Wang et al., 2016; Rücklé and Gurevych, 2017), compare-aggregate approaches (Wang and Jiang, 2017; Rücklé et al., 2019a), and, more recently, transformer-based models (Hashemi et al., 2020; Yang et al., 2020; Mass et al., 2019). Fine-tuning of large pre-trained transformers such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) currently achieves state-of-the-art performances on many related benchmarks (Garg et al., 2020; Mass et al., 2019; Rochette et al., 2019; Nogueira and Cho, 2019).

However, realistic scenarios often do not provide enough labeled data for supervised in-domain training. Thus, different recent work has focused on improving model performances in small data scenarios. Shah et al. (2018) use adversarial domain adaptation for duplicate question detection. Poerner and Schütze (2019) adapt the combination of different sentence embeddings to individual target domains. Rücklé et al. (2019b) use weakly supervised training, self-supervised training methods, and question generation. Similar approaches were also explored in ad-hoc retrieval (Zhang et al., 2020; Ma et al., 2020; MacAvaney et al., 2019). A crucial limitation of these approaches is that they result in entirely separate models for each dataset and are thus not re-usable. In this work, we therefore explore the zero-shot transfer capabilities of models, to understand how well they generalize to unseen settings.

Previous work of Yang et al. (2020) investigates this on a smaller scale. They propose USE-QA, a sentence encoder for comparing questions and

[1] https://github.com/ukplab/emnlp2020-multicqa

answers, and achieve promising zero-shot results in retrieval tasks. However, it is unclear how this model compares to the zero-shot performances of models trained on several different source domains and how best to combine the data from multiple domains. Other work addresses the generalization of models over several domains in different settings, e.g., for machine reading comprehension (Talmor and Berant, 2019; Fisch et al., 2019). More related to our work, Guo et al. (2020) propose a new evaluation suite with eight datasets for retrieval-based QA, in which they also study the effectiveness of USE-QA. In contrast to them, our work (1) deals with re-ranking setups and uses cross-encoders, which is different to their bi-encoder scenario for retrieval; (2) we deal with question and answer passages instead of answer sentences; (3) we study a large number of 140 source domains and provide important insights on zero-shot transfer performances in relation to domain similarity and data size, and extensively analyze the training of models on many source domains simultaneously.

3 Data and Setup

3.1 Training Data for 140 Domains

StackExchange is a network that consists of 172 cQA forums,[2] referred to as domains in the following, each devoted to a particular topic such as programming, traveling, finance, etc. From those 172 forums, 140 are in English and contain more than 1000 unlabeled questions.

We use data from each of these 140 English forums and train domain-specific models for semantic text matching. This has recently become feasible with self-supervised training methods such as *WS-TB* (Rücklé et al., 2019b), in which the question title is considered as a query to retrieve the question body (the detailed description of the question). This requires no labeled training instances and thus allows us to scale our experiments to 140 source domains which we can transfer from.

Formally, we train 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 $q_n \neq q_m$. We randomly sample q_m from the entire corpus. For computational reasons, we use a maximum of 100k positive training instances. This training technique performs well for duplicate question detection and answer selection (Rücklé et al., 2019b), and similar methods have been used for ad-hoc retrieval (MacAvaney et al., 2019).

Our different domains are clearly separated by topic. Because not all domains are equally popular, the training sizes are heavily imbalanced, see Figure 1. This allows us to analyse the impact of data size in regard to the transfer performances.

	Train	Dev	Test	Source
Non-Factoid Answer Selection				
InsuranceQA	12 889	1592	1625	In.-Library
WikiPassageQA	3332	417	416	Wikipedia
LAS-Apple	5831	765	766	StackEx.
LAS-Cooking	3692	791	792	StackEx.
LAS-Academia	2856	612	612	StackEx.
LAS-Travel	3572	765	766	StackEx.
LAS-Aviation	3035	650	652	StackEx.
Question Similarity				
SemEval17	267	50	88	QatarLiving
AskUbuntu	12 584	189	186	StackEx.

Table 1: The statistics of the evaluation benchmarks.

3.2 Evaluation Benchmarks

We transfer all models to 9 benchmark datasets from different domains. We categorize them in two broad tasks, non-factoid answer selection and question similarity. See Table 1 for the statistics.

Answer selection (AS). The goal is to 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 additional descriptions (question bodies). Answers to non-factoid questions are often long texts such as descriptions, explanations, and advice.

- *InsuranceQA* (Feng et al., 2015) is a benchmark crawled from an FAQ community,[3] in which licensed insurance practitioners answer user 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.
- *WikiPassageQA* (Cohen et al., 2018) was crowd-sourced from Wikipedia articles and is not restricted to a particular domain (although many questions are about history topics). Candidate answers are passages from a single doc-

[2] See https://stackexchange.com/sites. The data from all forums is publicly available https://archive.org/details/stackexchange

[3] https://www.insurancelibrary.com/

ument, on the basis of which the question was formulated. $|A| = 58$ of which 1.6 passages represent correct answers (on average).

- *Long Answer Selection (LAS)* datasets (Rücklé et al., 2019a) were crawled 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 measure mean average precision (MAP) on WikiPassageQA, and accuracy (P@1) otherwise.

Question similarity (QS). The goal is to re-rank a pool of potentially related forum questions C in regard to a query question q. All questions contain titles and bodies—which we concatenate—and are thus long multi-sentence texts. On all question similarity benchmarks we measure MAP.

- *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 related questions were retrieved with a search engine and manually labeled for relatedness in regard to q.
- *AskUbuntu* (Lei et al., 2016) 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 relevance. $|C| = 20$.

3.3 Models and Training

BERT models. We use a pointwise ranking architecture based on pre-trained language models. We concatenate the two input texts (separated with SEP token), and learn a linear classifier on top of the final CLS representation for scoring. We optimize the binary cross-entropy loss. Similar techniques achieve state-of-the-art results on many related datasets (Garg et al., 2020; Mass et al., 2019; Rücklé et al., 2019b).

For our zero-shot transfer experiments from single domains in §4, we use BERT base uncased (Devlin et al., 2019). Later in §5, we additionally investigate BERT large uncased and RoBERTa large (Liu et al., 2019). The hyperparameters for all setups are listed in Appendix A.1.

[4] https://www.qatarliving.com/forum

Training. We train our models with self-supervision, see §3.1. To obtain *in-domain models*, we fine-tune BERT with the respective training data of the benchmark datasets of §3.2. 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 AskUbuntu 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. For InsuranceQA and AskUbuntu, we randomly sample one negative candidate due to their larger training sizes.

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. 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. We use the recent architecture of Pfeiffer et al. (2020a), which makes it possible to investigate their adapter combination technique AdapterFusion in §5. In preliminary experiments, we find that using adapters in contrast to full model fine-tuning does not decrease the model performance while drastically reducing the number of parameters (one model is ∼5 MB).

4 Zero-Shot Transfer from 140 Domains

In this section, we study the zero-shot transfer performances of all models (§4.1) and investigate whether domain similarity and training data size are suitable for predicting the best models (§4.2).

4.1 Results

In Figure 2, we show the zero-shot transfer to all nine benchmarks. Except for SemEval17, all results are for the dev split.[5] Diamonds ⋄ show the performance of IR baselines[6] and in-domain BERT.

[5] SemEval17 does not contain a separate dev split.

[6] TF*IDF for LAS, BM25 for WikiPassageQA and InsuranceQA, and a search engine ranking for SemEval17 (which is the official challenge baseline).

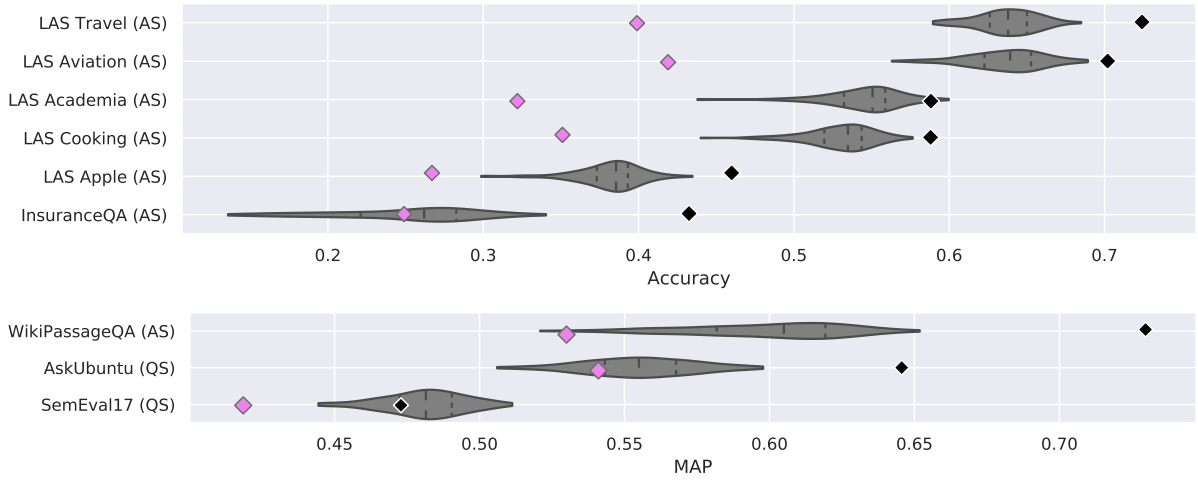

Figure 2: Zero-shot transfer performances of all 140 models to the evaluation benchmarks. For benchmarks that contain StackExchange data, we exclude the model from the respective source domain. The violin range visualizes the observed transfer scores, without extension or cut-off for extreme datapoints. Vertical lines show the mean and the quartiles. Diamonds show the performances of IR baselines (violet) and in-domain BERT models (black).

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.

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 capabilities of models. However, for target domains with few training instances (see Table 1), the differences of in-domain BERT to the best zero-shot transfer models are much smaller. Importantly, these setups pose crucial and realistic challenges for text matching approaches (Rücklé et al., 2019a,b). 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 traditional models such as LSTMs (Shah et al., 2018). Importantly, we find no substantial differences between question similarity and answer selection *tasks*, which are both not explicitly learned during training. We thereby take an important step towards overcoming the boundaries between individual tasks and domains.

4.2 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 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 the two most commonly used domain selection techniques: (1) domain similarity and (2) training size, in regard to the transfer

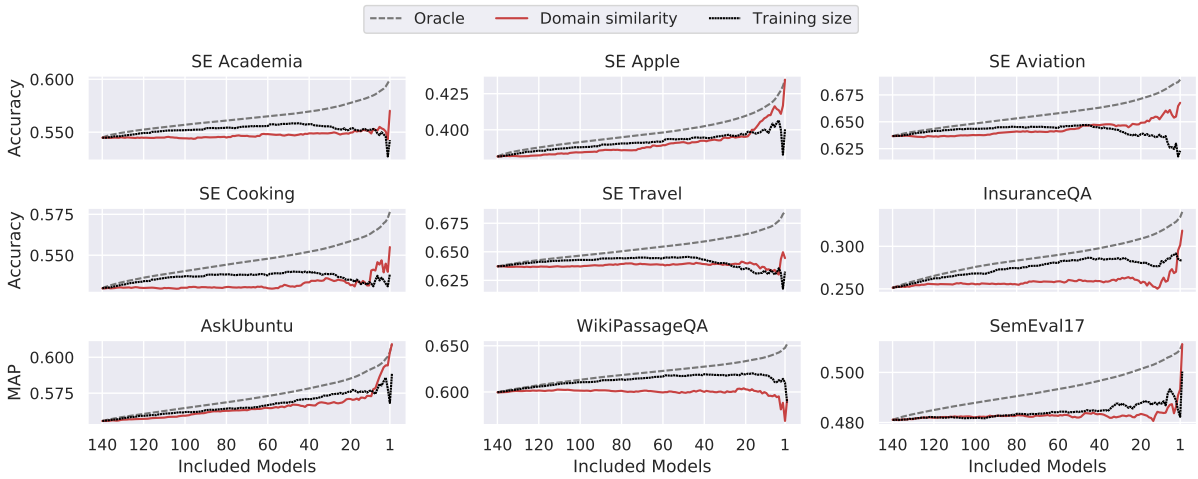

Figure 3: The average performance scores (y-axis) of subsets of models (x-axis) selected by domain similarity or training size (scores are averaged over the included models). The oracle always selects the best models.

performances. To simulate an optimal selection, we define an *oracle* that always identifies the best models. We present our findings in Figure 3.

Domain similarity. To measure the domain similarity, we embed the questions of all datasets with Sentence-RoBERTa (Reimers and Gurevych, 2019). For each dataset, we obtain the mean over all embeddings 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 3. 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 §4.1, such knowledge is not essential, i.e., a large number of models from more distant domains achieve good zero-shot transfer performances.

We provide examples of the best models and the most similar domains for three benchmarks in Table 2 (more are given in Appendix A.4). Many of the best models are from distant domains—e.g., 'Ethereum' for WikiPassageQA or 'SciFi' for LAS-Travel. This shows the importance of considering a broad selection of source domains, including ones that are not intuitively close.

Training size. The average performance of our models after removing the smallest domains improves more consistently (see WikiPassageQA and InsuranceQA in Figure 3). This shows, that the training size is more suitable for identifying models that achieve low performance scores—e.g., models that are trained on very 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. Importantly, this contrasts the common procedure of only including the largest domains for transfer (Shah et al., 2018).

Summary We have established that neither domain similarity nor training data size are suitable for predicting the best models. This shows that elaborate strategies are necessary for automatically identifying the most suitable training sets. Most importantly, we also demonstrate the importance of considering a broad selection of source domains instead of following the standard practice of merely relying on the most similar or largest domains. 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.

5 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**.

	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);
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);

Table 2: The best models and the most similar domains for three 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	WPQA	SemEval	Σ
		Accuracy scores							*MAP scores*			
Best models of §4		65.4	58.1	43.1	56.5	65.0	35.3	53.9	63.29	67.85	51.13	60.75
Self-Supervised Training												
MT	largest	64.5	56.7	40.9	56.0	65.3	32.8	52.7	63.23	66.32	50.11	59.88
AF	largest	63.4	57.7	42.9	59.1	65.9	28.6	52.9	63.11	66.99	**50.88**	60.32
MT	balanced	65.0	60.0	43.2	55.7	65.1	**38.2**	**54.5**	**63.40**	**68.28**	48.31	59.99
AF	balanced	62.2	58.0	**43.5**	**59.4**	66.2	29.8	53.1	62.93	67.51	47.95	59.46
MT	all	**66.1**	**60.3**	43.0	57.0	**66.4**	31.5	54.0	63.32	68.20	49.81	**60.44**
Extended Data												
MT	balanced	67.8	60.9	46.5	58.9	**69.1**	**34.9**	56.3	**65.07**	**67.70**	48.59	60.45
MT	all	**72.4**	**63.1**	**45.8**	**61.1**	68.0	34.7	**57.5**	64.12	66.82	**52.25**	**61.06**

Table 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, WPQA is WikiPassageQA. Σ shows the average performance of benchmarks that use the same performance measure.

5.1 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 (§4) with AdapterFusion (**AF**; Pfeiffer et al., 2020a). 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.[7]

Data. We use the training data of §3.1 and exclude the domains that are used in any of the evaluation datasets[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** included 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^{B}, MultiCQA$^{B\text{-}lg}$, and MultiCQA$^{RBa\text{-}lg}$. The training procedure, number of runs, and hyperparameters are as in §3.3.

We additionally compare our models to the question/answer encoder USE-QA (Yang et al., 2020), which is a state-of-the-art model for retrieving answers in zero-shot transfer setups. The IR baselines are the same as in §4.1 (TF*IDF for LAS, BM25

[7]We use AdapterFusion without value matrix to avoid additional regularization as in (Pfeiffer et al., 2020a).

[8]AskUbuntu, aviation, travel, cooking, academia, apple.

[9]Technology, culture, life, science, professional, and business. See Appendix A.5 for the list of included domains.

for WikiPassageQA and InsuranceQA, and a search engine ranking for SemEval17—the official challenge baseline).

5.2 Results

Multiple source domains. In Table 3, we show the results of MultiCQA^{B} with MT and AF for the different sets of source domains, and compare this to the respective best single-domain models of §4.

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 is 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 A.3, 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 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 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 all on *all 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 4, we compare our large MultiCQA models to the in-domain state of the art. 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$^{\text{RBa-lg}}$ model outperforms USE-QA on eight benchmarks, and performs better than the previous *in-domain* state of the art 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.

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 state-of-the-art results on all nine benchmarks.

6 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 error 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 on AskUbuntu, but in the wrong context. Table 5 shows an example of such a case, and we provide more examples and additional details in Appendix A.6. We find that this type of error 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 in context.

[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).

	Tr	Co	Ap	Ac	Av	IQA	Σ	AU	WPQA	SemEval	Σ
	Accuracy scores							*MAP scores*			
IR Baselines	39.9	35.1	26.7	32.2	41.9	24.9	33.4	54.10	53.00	41.85	49.65
Zero-Shot Transfer											
USE-QA	65.3	58.5	44.6	46.2	53.1	35.1	50.4	**67.81**	53.15	52.69	57.88
MultiCQA^{B}	72.4	63.1	45.8	61.1	68.0	34.7	57.5	64.12	66.82	52.25	61.06
MultiCQA$^{B\text{-lg}}$	75.5	64.6	50.0	64.0	72.0	32.8	59.8	66.48	69.83	51.56	62.62
MultiCQA$^{RBa\text{-lg}}$	**77.8**	**72.0**	**56.8**	**70.4**	**76.6**	**41.9**	**65.9**	63.29	**73.29**	**52.88**	**63.15**
In-Domain Models											
Previous SoTA	69.5†	58.3†	47.3†	58.7†	65.5†	49.8‡	58.2	69.13†	74.90*	51.56$^{\diamond}$	64.76
BERT	68.7	59.0	47.0	59.0	64.5	42.2	56.7	67.31	75.09	47.29	63.23
BERT-lg	72.5	62.4	47.2	60.0	68.3	42.7	58.8	67.54	76.22	45.86	63.20
RoBERTa-lg	70.9	68.4	50.7	66.3	68.7	44.9	61.6	70.18	79.74	48.70	66.20
MultiCQA$^{RBa\text{-lg}}$	**80.5**	**76.8**	**60.2**	**72.1**	**81.8**	**50.8**	**70.3**	**72.28**	**81.41**	**53.61**	**69.10**

Table 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 WPQA is WikiPassageQA. Σ shows the average performance of benchmarks that use the same performance measure. † shows the scores of the best BERT models of (Rücklé et al., 2019b), ‡ 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).

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$^{RBa\text{-lg}}$): 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 5: A mistake of MultiCQA$^{RBa\text{-lg}}$ (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).

7 Conclusion

We studied the zero-shot transfer of text matching models on a massive scale, with 140 different source domains and nine benchmark datasets of 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 combination methods, and were able to extensively analyze a large sample size.

We have shown that (1) BERT models trained in a self-supervised manner on cQA forum data transfer well to all our benchmarks, even across distant domains; (2) training data size and domain similarity are not suitable for predicting the zero-shot transfer performances, revealing that a broad selection of source domains is crucial; (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 state-of-the-art zero-shot performances on six benchmarks; (4) fine-tuning MultiCQA$^{RBa\text{-lg}}$ in-domain further improves the performances and achieves state-of-the-art results on all nine benchmarks.

We clearly demonstrated the effectiveness and the relevance of zero-shot transfer in many realistic scenarios and believe that our work lays foundations for a wide range of research questions. 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. Researchers could also use our 140 domain-specific adapters and investigate further combination techniques to make them even more broadly applicable.

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (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; the German Research Foundation under grant EC 503/1-1 and GU 798/21-1; the BMBF under the promotional reference 01IS17050

(Software Campus); and the LOEWE initiative (Hesse, Germany) within the emergenCITY center.

References

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP 2019)*, pages 3490–3496, Hong Kong, China. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations (ICLR 2020)*, Addis Ababa, Ethiopia.

Daniel Cohen, Liu Yang, and W. Bruce Croft. 2018. 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)*, page 1165–1168, Ann Arbor, MI, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Cicero Dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL 2015)*, pages 694–699. Association for Computational Linguistics.

Cicero Dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive Pooling Networks. *arXiv preprint arXiv:1602.03609*.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2015)*, pages 813–820.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*. Association for the Advancement of Artificial Intelligence.

Mandy Guo, Yinfei Yang, Daniel Cer, Qinlan Shen, and Noah Constant. 2020. MultiReQA: A cross-domain evaluation for retrieval question answering models. *arXiv preprint arXiv:2005.02507*.

Hojae Han, Seungtaek Choi, Haeju Park, and Seungwon Hwang. 2019. MICRON: Multigranular interaction for contextualizing RepresentatiON in non-factoid question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP 2019)*, pages 5890–5895, Hong Kong, China. Association for Computational Linguistics.

Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W Bruce Croft. 2020. ANTIQUE: A non-factoid question answering benchmark. In *European Conference on Information Retrieval (ECIR 2020)*, pages 166–173, Lisbon, Portugal. Springer.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzkebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, pages 2790–2799, Long Beach, California, USA.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*, pages 1279–1289, San Diego, California. Association for Computational Linguistics.

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

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2020. Zero-shot neural retrieval via domain-targeted synthetic query generation. *arXiv preprint arXiv:2004.14503*.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Universal text representation from BERT: An empirical study. *arXiv preprint arXiv:1910.07973*.

Sean MacAvaney, Andrew Yates, Kai Hui, and Ophir Frieder. 2019. 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)*, page 993–996, Paris, France. Association for Computing Machinery.

Yosi Mass, Haggai Roitman, Shai Erera, Or Rivlin, Bar Weiner, and David Konopnicki. 2019. A study of BERT for non-factoid question-answering under passage length constraints. *arXiv preprint arXiv:1908.06780*.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, Vancouver, Canada. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterFusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020b. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, Online.

Nina Poerner and Hinrich Schütze. 2019. 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP 2019)*, pages 1630–1641, Hong Kong, China. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*, pages 506–516, Long Beach, CA, USA.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP 2019)*, pages 3980–3990, Hong Kong, China. Association for Computational Linguistics.

Alexandre Rochette, Yadollah Yaghoobzadeh, and Timothy J Hazen. 2019. Unsupervised domain adaptation of contextual embeddings for low-resource duplicate question detection. *arXiv preprint arXiv:1911.02645*.

Andreas Rücklé and Iryna Gurevych. 2017. Representation learning for answer selection with LSTM-based importance weighting. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS 2017)*. Association for Computational Linguistics.

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych. 2019a. 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)*, pages 6932–6939. Association for the Advancement of Artificial Intelligence.

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych. 2019b. Neural duplicate question detection without labeled training data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP 2019)*, pages 1607–1617, Hong Kong, China. Association for Computational Linguistics.

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 1056–1063, Brussels, Belgium. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2019. 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)*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 464–473, Berlin, Germany. Association for Computational Linguistics.

Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui. 2017. 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)*, page 695–704, Shinjuku, Tokyo, Japan. Association for Computing Machinery.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2010. What Is Not in the Bag of Words for Why-QA? *Computational Linguistics*, 36(2):229–245.

Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1288–1297, Berlin, Germany. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2017. A Compare-Aggregate Model for Matching Text Sequences. *International Conference on Learning Representations (ICLR 2017)*.

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. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

Kaitao Zhang, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2020. Selective weak supervision for neural information retrieval. In *Proceedings of The Web Conference 2020 (WWW 2020)*, page 474–485, Taipei, Taiwan. Association for Computing Machinery.

A Appendices

A.1 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 very 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 are 43.25, 40.00, and 39.25 (accuracy), respectively. The development scores on WikiPassageQA are 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., 2020a), which are 0.0001 and 5×10^{-5}, respectively.

A.2 Computing Infrastructure

We used a heterogenous 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).

A.3 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. (2020a), 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 4 and in Figure 5 we plot the activatations 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 §5.2). 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 6. 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.

A.4 Individual Transfer: Best Models and Most Similar Domains

We show the best models and most similar domains for all benchmark in Table 6 (we have provided an excerpt of that in Table 2 of the paper). In particular, we see that the best source domains vary across the different benchmarks. Often, the best models are not from intuitively close domains nor from domains with large training sets.

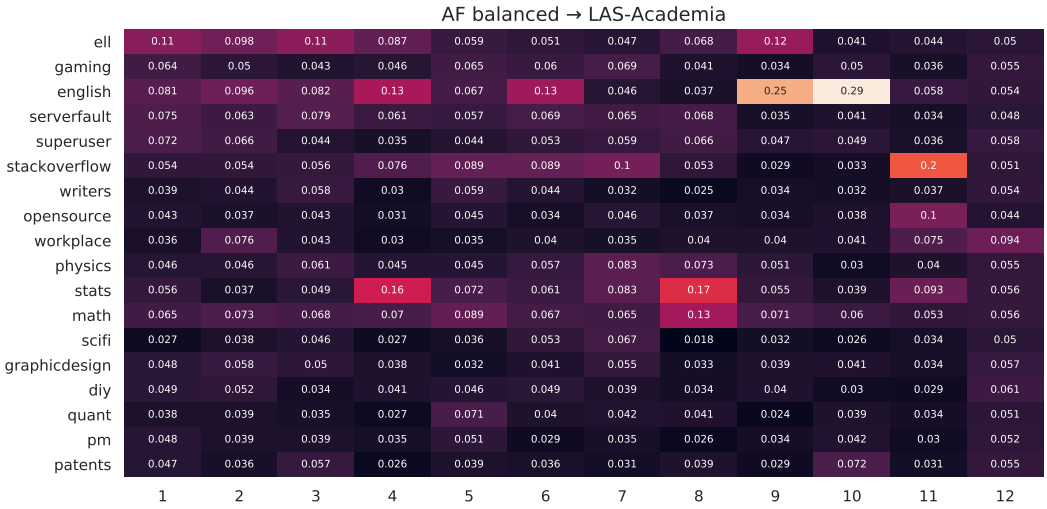

Figure 4: Adapter activations in individual BERT layers for AF balanced when transferring to LAS-Academia.

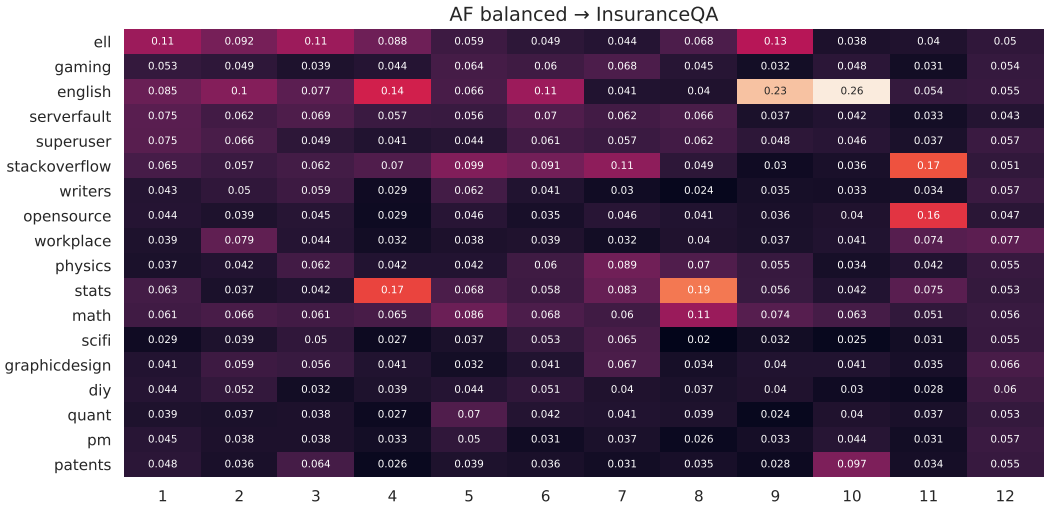

Figure 5: Adapter activations in individual BERT layers for AF balanced when transferring to InsuranceQA.

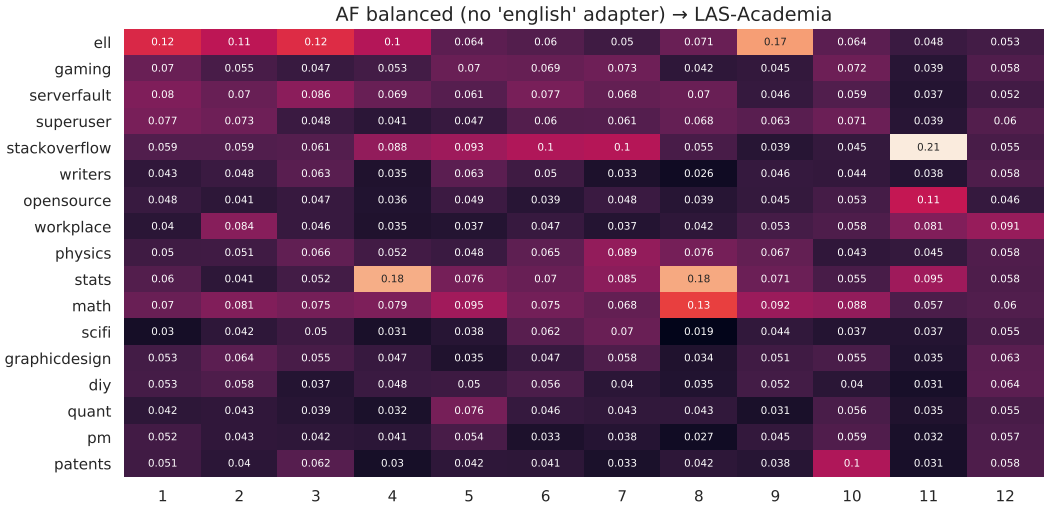

Figure 6: Adapter activations in individual BERT layers for AF balanced (excluding the adapter from the 'English' domain) when transferring to LAS-Academia.

	Best	Most similar
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);
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);
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);
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);
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-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-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);

Table 6: The best models and the most similar domains for all 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.

A.5 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 §5 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). The 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

A.6 Examples of Wrong Predictions on InsuranceQA and AskUbuntu

We provide additional examples of mistakes made by MultiCQA$^{\text{RBa-lg}}$ (zero-shot transfer) on AskUbuntu and InsuranceQA to complement our brief analysis in Section 6.

Table 7 shows an additional example for 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 possible number of CPUs. Tables 8 and 9 show examples of similar problems in InsuranceQA.

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: AskUbuntu example (shortened). This shows that our model mostly focuses on *number of CPUs* and *kernel* information instead of recognizing that the crucial information is the *maximum number*. We underline important aspects that differ.

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 8: InsuranceQA example 1 (shortened). 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 9: InsuranceQA example 2 (shortened). 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.