Importance Weighting for Robust QA

Stanford CS224N {Default} Project

Elinor Brondwine
Department of Computer Science
Stanford University
elinorb@stanford.edu

Abstract

Machine Reading Comprehension (MRC) Questions Answering (QA) systems are commonly used within conversational agents and search engines to support users information needs while saving users the effort of navigation in documents, when the information need is a question for which the user seeks an answer. While state of the art approaches have shown to be successful for QA on a general domain, enterprise retrieval problems where the information need for QA exists in domains that are specialized and have limited or none annotated data remain open. In this work we address adaptation to new specialized domains with very little training data for MRC-QA , focusing on importance weighting. We propose two features for importance weighting that are applicable for an unsupervised setting, and present preliminary results comparing importance weighting with transfer learning.

1 Key Information to include

- Mentor: Chris Waites

2 Introduction

Machine Reading Comprehension (MRC) Questions Answering (QA) is a task in which a model receives a text segment (context) and a question about that context, and infers an answer to the question in the form of a text span from the context. MRC-QA systems are commonly used within conversational agents and search engines to support users information needs while saving them the effort of navigation in documents, when the information need is a question for which the user seeks an answer.

The SQuAD[1] competition drove important advances in MRC-based QA and led to rapid improvements. However, while successful in-domain, models trained for the MRC-QA task have been failing to generalize beyond the training distribution, learning superficial correlations[2].

This paper is focused on MRC-based QA in new specialized domains, using domain adaptation with very little data. This problem is highly motivated by practice via enterprise retrieval problems, where the information need for QA exists in specialized domains that have limited or none annotated data.

In our experiments we take an off-the-shelf pre-trained transformer model (a QA variant of DistilBERT [3, 4]) that was trained on a general domain dataset. We then utilize a general QA dataset together with a few examples from the specialized domain, with the goal of adapting this model for the specialized domain. Specifically, we use the specialized domain examples to modify the training by employing importance weighting[5], a technique in which we apply a weight of importance to each sample in the training set, based on some feature f that is a distributional feature which ideally captures similarity to the specialized domain data. We propose two feature for the importance weighting: (1) the question length measured by unique tokens, and (2) vocabulary similarity measured by the number of similar tokens in the question and context. We compare the effectiveness of importance weighting in comparison to the base-model with and without additional fine-tuning on very little specialized domain data. Our features utilize the question and the context, resulting in a setup that is applicable where no supervised data is available for the specialized domain.

The general QA dataset used is a combination of SQuAD[1], NewsQA[6], and Natural Questions[7], and we refer to it as (in-domain data). The specialized domain QA dataset used is a combination of DuoRC[8], RACE[9], and Relation Extraction[10], and we refer to it as (out-of-domain data).

Our results are in line with the conclusion of previous work [5], showing that importance weighting can be effective in comparison to the base model, but not in comparison to additional fine-tuning. While previous work[5] showed that using importance sampling did not hurt model effectiveness, in our experiments fine-tuning with the base-model outperforms configurations that employ importance weighting.

3 Related Work

While data availability has led to improvements in the MRC-QA task when targeting general domains [1], this task remains open for specialized domain. Most work on MRC-based QA has been focused on datasets which are characterised by short factoid-style answers in a general domain, including SQuAD[1], NewsQa[6], SearchQA[11], Trivia QA[12], and MS-MACRO[13]; specialized domain datasets for this task are more rare, and are mainly in the medical domain (i.e., emrQA[14] and MEDIQA[15]). In our experiments we train on a general domain and focus on adapting to specialized domains from various topics: movie reviews[8], examination data[9], and wikipedia-synthetic[10].

Multiple solutions have been proposed for adaptation of deep learning models in the context or MRC-QA , including mixture of expert[16], data augmentation[17, 18], adversarial training [19, 20] and meta learning[21, 22]. Another common approach is employing transfer learning by training first on a general domain dataset then fine-tuning on a small set of examples from a specialized domain data[5, 23, 24]. We use similar approach in some of our experiments.

Importance weighting for MRC-QA using answer length has shown to be effective in case none or very little data is available from the specialized domain, otherwise the added value in comparison to transfer learning was negligible [5]. In our work we explore different features for importance weighting, and use different datasets. We propose and implement two features for importance weighting: question length and vocabulary similarity. Our features do not require having the answer, and so our basemodels are applicable where no supervised data is available from the specialized domain. Similar to previous results [5] importance weighting configurations outperform the base model in our experiments. In contrast, when used with additional fine-tuning, configurations utilizing importance weighting show inferior results in comparison to using fine-tuning with no importance weighting.

4 Approach

In this section we are going to present the methodology we used in our experiments.

4.1 Model Framework

Baselines We leverage a pre-trained transformer language model called DistilBERT [4] as our base model. For our baselines we trained that model on an additional QA data set. We denote this model $\mathrm{DistilBERT}(D)$, where D denotes the QA specific dataset used to train the weights of the model. We define two configurations for the base model: one for the in-domain dataset and one for the out-of-domain sdataet. DistilBERT is 40% smaller than the original BERT[25], while retaining 97% language understanding and improving speed by 60%[3]. The QA implementation leverages an additional classification head that classifies the probability of a span in the context to be the answer to the question.

Importance Weighting In this configuration we adapt the base model by applying importance weighting to the in-domain training data used. We use the distribution of feature from the in-domain and out-of-domain datasets to estimate the resemblance of each training example in the in-domain to the out-of-domain dataset. We then weight the loss assigned to our training examples accordingly, emphasizing learning from examples that are likely to represent the out-of-domain when learning the model parameters by assigning such examples higher weights. This configuration is denoted $\mathrm{DistilBERT}_w$, where w denotes the weighting configuration used to weight the in-domain training dataset. We present the specific features used in Section 4.2 and the implementation details in Section 5.3.

Fine-tuning for Domain Adaptation We implemented a fine-tuning step for additional domain adaptation using the out-of-domain data. This is done as a second step, on top of an existing model from one of the configuration described above. We introduce additional configuration, in which we perform this step on top of the in-domain baseline as well as the importance weighting models. This models are denoted $\mathrm{DistilBERT}^f$ and $\mathrm{DistilBERT}^f_w$ respectively.

In Table 1 we provide a list of the resulting models and respective notation. We dedicate the rest of this section to elaborate on the features used for importance weighting.

Notation	Training data	Fine-tuning
Baselines		
$\mathrm{DistilBERT}(in)$[1]	in-domain	-
$\mathrm{DistilBERT}(out)$	out-of-domain	-
Importance weighting		
$\mathrm{DistilBERT}_{vocab}$	weighted in-domain - Shared Vocabulary	-
$\mathrm{DistilBERT}_{len}$	weighted in-domain - Question Length	-
Additional domain adaptation		
$\mathrm{DistilBERT}^f$	in-domain	out-of-domain
$\mathrm{DistilBERT}^f_{vocab}$	weighted in-domain - Shared Vocabulary	out-of-domain
$\mathrm{DistilBERT}^f_{len}$	weighted in-domain - Question Length	out-of-domain

Table 1: List of the models we used in our experiments: baseline models, importance weighting configurations, and configurations for which we performed additional fine-tuning with the out-of-domain examples. The training data used for the base model is listed in the middle column.

4.2 Importance Weighting Configurations

Similar to previous work[5], we define the weight given to sample s as

$$w(s) = \frac{p_{\text{out}}(s)}{p_{\text{in}}(s)} \tag{1}$$

where p_{out}, p_{in} denote the likelihood of observing a sample in the out-of-domain dataset and the in-domain dataset, respectively. The likelihood is estimated with respect to a given feature. We propose and implement two features for importance weighting: question length and vocabulary similarity. Our features do not utilize the answer, resulting in base models configurations that are applicable to unsupervised settings.

4.2.1 Question Length

For this feature we estimate the likelihood of a sample according to the distribution of question lengths. Specifically we look at the number of unique tokens that appeared in the question. We denote models using this feature by setting $w = len$. To avoid zero probabilities we smooth the distributions by adding 1 to each length category in the distribution.

4.2.2 Shared Vocabulary

Let t be a token in the out-of-domain and in-domain dataset. We calculate the maximum likelihood estimator for token t for each dataset D, assuming a unigram language model. We define $p(t)$ as the probability of the unigram language model estimated to generate token t. In this configuration, we calculate a weight w' for each individual token t by dividing the probability of the estimated out-of-domain language model to generate token t with the probability of the estimated in-domain language model to generate token t. We define the weight of sample s as the sum over all token weights in the question and context of that sample:

$$w(s) = \sum_{t \in s} w'(t) \tag{2}$$

For the estimation of the language models we use all tokens in the question and context. If a token is out of vocabulary for the out-of-domain datasets, we assign it a low probability to avoid zero probabilities. We denote models using this feature by setting $w = vocab$.

5 Experiments

5.1 Data

Data	Question source	Context Source	Train	Dev	Test
		in-domain datasets			
SQuAD[1]	Crowdsourced	Wikipedia	50,000	10,507	-
NewsQA[6]	Crowdsourced	News articles	50,000	4,212	-
Natural Questions[7]	Search logs	Wikipedia	50,000	12,836	-
		out-of-domain datasets			
DuoRC[8]	Crowdsourced	Movie reviews	127	126	1,248
RACE[9]	Teachers	Examinations	127	128	419
RelationExtraction[10]	Synthetic	Wikipedia	127	128	2,693

Table 2: Statistics for datasets used for building the QA system for this project. Question Source and Context Source refer to data sources from which the questions and context were obtained. Source: CS224n handout and Fisch et al.[26]

In Table 2 we preset the data sets used in the training and in our experiments. The input data that is used for MRC-QA is a question and a respective context - a passage of text that includes the answer to the question. The output is the answer to the question - a span of text from the context that has the answer. The datasets that were used for training are Wikipedia/news datasets together with a small size of out of domain datasets from various topics: movie reviews, examination, and Wikipedia-synthetic.

5.2 Evaluation Method

We report the EM (exact match) and F1 measures on the out-of-domain validation and test set.

5.3 Experimental Details

We utilize a pre-trained off-the-shelf QA for our baselines and for the fine-tuning, DistilBertForQuestionAnswering [2]. We use the parameter values provided in the robustQA project default baseline [3] and run the experiments with a fix random seed of 42. A partial list of the parameter values is reported in Table 3. For the best performing model, $\mathrm{DistilBERT}^f$, we experimented with additional epoch sizes. We report the results from these experiments in Tables 5 and 6. In Table 4 and Figure 1 we report the best result of the model (#epochs may vary).

For the importance weighting, we pre-computed the feature weights based on the training and validation data and implemented an extension of DistilBertForQuestionAnswering that computes the weighted loss function used in training, using a weight vector and an input.

5.4 Results

5.4.1 Test Leader Board - RobustQA

We report the test results in Table 4. We had a quota of 4 submissions to the RobustQA Test Leader Board. We selected the models that showed improvements on the validation set. As a comparison we

[2] https://huggingface.co/transformers/model_doc/distilbert.html

[3] https://github.com/MurtyShikhar/robustqa

Learning rate	3e-05
Batch size	16
# Epochs	3

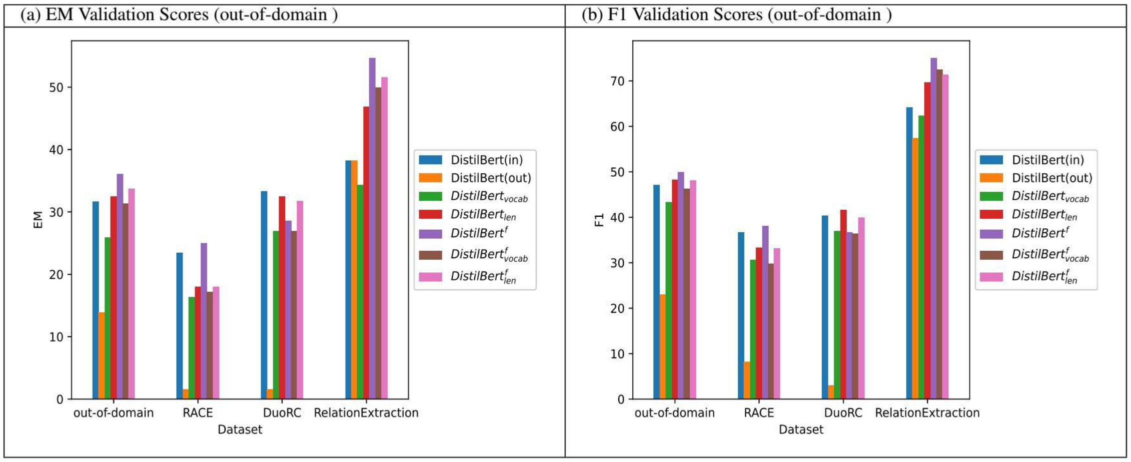
Table 3: Parameter values used[4].

also used an importance weighting configuration without fine-tuning. The results are consistent with what we have seen on the validations data, showing that fine-tuning with the base-model outperforms configurations that employ importance weighting. In the next section we will present the validation results.

Model	EM	F1
$\mathrm{DistilBERT}^{f}$	**41.628**	**59.141**
$\mathrm{DistilBERT}^{f}_{len}$	41.078	58.409
$\mathrm{DistilBERT}^{f}_{vocab}$	39.725	57.579
$\mathrm{DistilBERT}_{vocab}$	38.073	56.798

Table 4: Test set results: EM and F1 scores from the Test Leader Board - RobustQA. The line separates fine-tuned model from the weighted model that uses Shared Vocabulary without additional fine-tuning. Best results are boldfaced.

5.4.2 Validation Results (out-of-domain)



(a) EM Validation Scores (out-of-domain)

(b) F1 Validation Scores (out-of-domain)

Figure 1: Reporting the evaluation measures for all models. In cases of multiple options (different epochs number) we report the best performing configuration.

In Figure 1 we report the EM and F1 scores for prediction on the validation out-of-domain datasets (both separately and pooled). Where multiple results were available per model we report the results of the best performing model (based on the all out-of-domain validation data). All configurations perform better on the Relation-Extraction dataset in comparison to the other datasets. The best performing model for both evaluations measure and in most datasets is $\mathrm{DistilBERT}^{f}$ (blue), a fine-tune only configuration (no importance weighting). The lowest performing model is $\mathrm{DistilBERT}(out)$ (orange), that is trained on the very small out-of-domain training data.

When looking at Figure 1 at the baselines models, $\mathrm{DistilBERT}(in)$ (blue) and $\mathrm{DistilBERT}(out)$ (orange), we see that training with in-domain data outperforms training with out-of-domain data, demonstrating that training on a large size general QA dataset can be more beneficial than training on a very small out-of-domain data. The amount of out-of-domain data is probably too small to properly train the model parameters. An exception for this is the EM scores received for the Relation-Extraction dataset. A possible reason for that could be the synthetic nature of the dataset.

Model	Description	Fine-tuning # epochs	All	RACE	DuoRC	RelationExtraction
DistilBERT(in)	in-domain		31.67	23.44	33.33	38.28
DistilBERT(out)	out-of-domain		13.87	1.56	1.59	38.28
DistilBERT$_{vocab}$	Shared Vocabulary		25.92	16.41	26.98	34.38
DistilBERT$_{len}$	Question Length		32.46	17.97	32.54	46.88
DistilBERT^f	in-domain	2	31.68	23.44	**33.33**	38.28
DistilBERT^f	in-domain	3	35.602	**25**	29.37	52.34
DistilBERT^f	in-domain	4	**36.13**	**25**	28.57	**54.69**
DistilBERT^f	in-domain	7	**36.13**	**25**	28.57	**54.69**
DistilBERT$^f_{vocab}$	Shared Vocabulary	3	31.41	17.19	26.98	50
DistilBERT$^f_{len}$	Question Length	3	33.7	17.97	31.75	51.56
DistilBERT$^f_{len}$	Question Length	4	33.77	17.97	31.75	51.56

Table 5: EM validation set results: EM scores received for the different configurations calculated on the validation set of the out-of-domain data. Best results are boldfaced.

Model	Base model training data	Fine-tuning # epochs	out-of-domain	RACE	DuoRC	RelationExtraction
DistilBERT(in)	in-domain		47.1	36.76	40.31	64.12
DistilBERT(out)	out-of-domain		23.01	8.22	3.03	57.47
DistilBERT$_{vocab}$	Shared Vocabulary		43.4	30.67	37.04	62.39
DistilBERT$_{len}$	Question Length		48.27	33.31	**41.67**	69.73
DistilBERT^f	in-domain	2	47.21	36.5	41.24	63.79
DistilBERT^f	in-domain	3	**50.26**	**38.22**	38.56	73.83
DistilBERT^f	in-domain	4	50.01	38.09	36.7	**75.05**
DistilBERT^f	in-domain	7	50.01	38.09	36.7	75.05
DistilBERT$^f_{vocab}$	Shared Vocabulary	3	46.24	29.74	36.38	72.46
DistilBERT$^f_{len}$	Question Length	3	48.16	33.16	39.89	71.31
DistilBERT$^f_{len}$	Question Length	4	48.16	33.16	39.89	71.31

Table 6: F1 validation set results: F1 scores received for the different configurations calculated on the validation set of the out-of-domain data. Best results are boldfaced.

Additionally, Figure 1 shows that DistilBERT(in) is out-performing DistilBERT$_{vocab}$ (green) consistently. In contrast, DistilBERT$_{len}$ (red) outperforms DistilBERT(in) on average - in line with previous work[5], however when looking at the datasets separately we see high variability in the effectiveness of this model in comparison to the baseline. Using importance weighting with the features we proposed did not consistently improve the model effectiveness over the baseline. When looking at configurations that employ importance weighting in Figure 1, we see that Question Length based configurations outperform Shared Vocabulary based configurations, across all datasets and independent of additional fine-tuning. This consistency suggests that better features may lead to better results in importance weighting without trade-offs. Given the small amount of samples, vocabulary features are perhaps too granular and do not allow generalization for the rest of the out-of-domain data.

Fine-tuning over the baseline model, DistilBERT^f (purple in Figure 1) outperforms all other configurations on the out-of-domain dataset. Specifically, in most datasets importance weighting configurations result in inferior effectiveness in comparison to using fine-tuning with out-of-domain data alone. This result is inconsistent with previous work where importance weighting (however, with answer based features) resulted in similar effectiveness when used in addition to fine-tuning with out-of-domain data[5].

Tables 5 and 6 complete the information we presented in Figure 1. Similarly, we report the EM and F1 scores for prediction on the validation data for the different baselines, respectively, for all of our model configurations: baselines, importance weighting only, fine-tuning only, and the combination of importance weighting with additional fine-tuning. Fine-tuning is always performed with the out-of-domain training data, and the base-model training data is specified per model. In contrast to Figure 1, Tables 5 and 6 report multiple results from explorations with different epochs sizes.

6 Analysis

Figure 2 shows the smoothed distribution of the number of unique tokens per question in the in-domain and out-of-domain datasets and the relative importance weights (a sum normalization over the weight assigned). Looking at the Question Length based weights, we see that we are mostly learning from examples with very short or very long questions. We see that samples with 23 tokens are assigned a high weight, since 52 sample out of 117k are of that length in in-domain data in comparison to 1 sample out of 763 of out-of-domain data. Since the difference in size between the datasets is so big, it is possible that the smoothing is leading to some undesired effects.

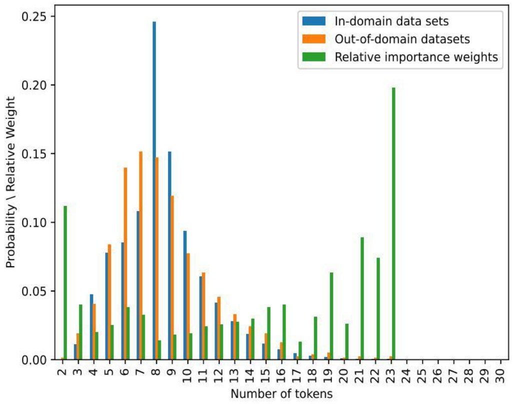


Figure 2: The smoothed distribution of the number of unique tokens per question in the in-domain and out-of-domain datasetes, and the relative importance weights (a sum normalization over the weight assigned).

Surprisingly, even with importance weighting for longer examples, the weighted configurations are failing to predict the correct answer for the long questions that was in the validation set: "Which is the best ticket to buy if you live in London and want to go to a small town 80miles away for four days?". The desired answer, "Monthly Returns", is correctly predicted by the $\mathrm{DistilBERT}^{f}$ model, but configurations that employ Question Length importance weighting get it wrong, with "up to 45% on the standard fare ." and "up to 45%" as predicted answers for configurations without and with additional fine-tuning, respectively.

7 Conclusion

This project addresses adaptation to new specialized domains with very little training data, focusing on importance weighting. Our experiments show that importance weighting is inferior to fine-tuning on the specialized domain data. We proposed distributional features that do not make use of the answer, opening the door to unsupervised importance weighting. We were able to demonstrate improvements when using importance weighing over the baseline using the number of unique tokens in the query, however this improvement was not consistent across all domains.

This work is preliminary - to better understand the potential of importance weighting for QA domain adaptation (in English), additional exploration is required. Specifically, a wider verity of features and domains, as well as testing different parameter configurations. Comparing to importance weighting with answer length is lacking due to time limitations.

References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[2] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[3] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[4] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[5] Timothy J Hazen, Shehzaad Dhuliawala, and Daniel Boies. Towards domain adaptation from limited data for question answering using deep neural networks. *arXiv preprint arXiv:1911.02655*, 2019.

[6] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*, 2016.

[7] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[8] Amrita Saha, Rahul Aralikatte, Mitesh M Khapra, and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. *arXiv preprint arXiv:1804.07927*, 2018.

[9] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[10] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*, 2017.

[11] Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

[12] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[13] 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 *CoCo@ NIPS*, 2016.

[14] Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. emrqa: A large corpus for question answering on electronic medical records. *arXiv preprint arXiv:1809.00732*, 2018.

[15] Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 370–379, 2019.

[16] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[17] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. An exploration of data augmentation and sampling techniques for domain-agnostic question answering. *arXiv preprint arXiv:1912.02145*, 2019.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, 2018.

[19] Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. Adversarial training for cross-domain universal dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 71–79, 2017.

[20] Yu Cao, Meng Fang, Baosheng Yu, and Joey Tianyi Zhou. Unsupervised domain adaptation on reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7480–7487, 2020.

[21] Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. Investigating meta-learning algorithms for low-resource natural language understanding tasks. *arXiv preprint arXiv:1908.10423*, 2019.

[22] Trapit Bansal, Rishikesh Jha, and Andrew McCallum. Learning to few-shot learn across diverse natural language classification tasks. *arXiv preprint arXiv:1911.03863*, 2019.

[23] Georg Wiese, Dirk Weissenborn, and Mariana Neves. Neural domain adaptation for biomedical question answering. *arXiv preprint arXiv:1706.03610*, 2017.

[24] Yu-An Chung, Hung-Yi Lee, and James Glass. Supervised and unsupervised transfer learning for question answering. *arXiv preprint arXiv:1711.05345*, 2017.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[26] Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. *arXiv preprint arXiv:1910.09753*, 2019.