Improving Robustness of Question-Answering System Using Domain-adaptive Pretraining, Adversarial Training , Data Augmentation and Finetuning

Stanford CS224N Default Project (Robust QA track)
Mentor: Chris Waites
Late days are shared. Zhengzhi: 4 days left **Ju:** 2 days left

Ju Zhang
juzhang@stanford.edu

Zhengzhi Lou
szlou@stanford.edu

Abstract

Question-Answering (QA) system based on neural language models (NLM) is known to be highly sensitive to the knowledge domain of training data and often has inferior performance when used for out-of-domain QA tasks. In this paper, the authors attempt to combine a few published methods to improve the robustness of the QA system on out-of-domain data. We first introduce methods we have tried, including domain adversarial training, domain adaptive pretraining, finetuning on few samples, and data augmentation. We then apply these methods through experimentation, improving the robustness of our baseline model on out-of-domain test datasets given two groups of training datasets: three large in-domain datasets and three very small out-of-domain datasets. We experimented and evaluated the effects of the above-mentioned methods both individually and combined. We found that while the individual method generates mixed results, the combination of them can improve the robustness of the baseline model in the QA task to the greatest extent on the out-of-domain datasets.

1 Introduction

QA system is an important domain of natural language processing (NLP), and it has a long history of being researched: the earliest prototype of such system can be traced back to 1960s-1970s [1] [2]. With the booming of deep learning in recent years, QA systems constructed on deep neural networks have become the mainstream [3]. Currently, many applications of QA system based on neural networks such as chatbot [4], search engine [5] and medical information assistant [6] has been developed and even achieved success in business.

However, the neural network-based QA system needs to be trained on a large amount of data. Since most data required for training neural network-based QA system need human annotation [7], in real-world situations, researchers often face the problem of building a QA system for knowledge domains that are different from the training data. For example, while there are many matured QA datasets are based on news and Wikipedia content, the objective may be to build a QA system for medical knowledge, where the training data are harder to find to fine-tune the models.

In these cases, the researchers usually have abundant in-domain data but scarce out-of-domain data to train the neural network models. Studies have already shown that neural network models tend to learn superficial correlations in training data that fail to generalize across different distribution [8] [9]. Consequently, it is often observed that these QA systems which have superb performance when completing tasks within the knowledge domain of their training data can have dramatically decreased performance when completing tasks out of the knowledge domain [10].

To solve this problem, in recent years many research has been done to improve the robustness of the QA system so that it can maintain high performance across different domains. These techniques include mix-of-expert system [11], adversarial training [12] [13], better finetuning techniques on few-shot examples [14], task adaptive finetuning [15], data augmentation [16], and meta-learning [17].

In this project, we aimed to selectively incorporate four robustness-improving techniques listed above, including domain-adaptive pretraining, adversarial training, data augmentation, and few sample finetuning into the baseline model training process and evaluate each technique's effect on the model robustness. Finally, we managed to combine all individual techniques and evaluated how collectively these methods could improve the robustness of the QA system.

2 Related Work

Many recent works with different techniques to improve the robustness of QA systems have been published. We selected four methods that are both promising and within the scope of a course project.

2.1 Data Augmentation

Data augmentation has been shown to be a useful tool to enhance the robustness of deep learning models, particularly in computer vision problems. It is also being increasingly used in NLP problems. Since the weak robustness of the QA system on out-of-domain tasks may be partly due to the learning of superficial correlations in the in-domain data which is hard to generalize [8], using data augmentation may help models ignore these superficial correlations and instead learn the invariances in data. There are many works presenting different effective data augmentation techniques. Back translation is one common strategy, in which the texts of questions and passages are translated into a different language and then back to the original one. In this way, paraphrased training data is created in an automatic way [18]. Word substitution [19] and alignment shifting [20] are also two strategies to create augmented data. A recent study also presented the work of using an automatic system to create semantically equivalent adversaries (SEA) to augment the training data [21].

2.2 Adversarial Training

The core idea of adversarial training is to help models learn domain-invariant features rather than features that are specific to one domain [22]. After being proposed in 2014 [23], adversarial training has been widely applied in the computer vision field to help with tasks such as image recognition [24]. It has also be extended to NLP tasks such as text classification [25] and relation extraction [26]. In a recent publication, adversarial training has been successful used in QA tasks and shown to improve the F1 score by up to 2 points compared to the same model without adversarial training [27]. In this study, the authors created one discriminator to classify the question and context into domains and one normal BERT-based [28] QA model. The latter learns domain-invariant features by trying to project texts into domain-invariant embeddings that can confuse the discriminator.

2.3 Domain-adaptive Pretraining

Since the BERT models are usually pretrained on general domain corpora such as web text and news, it is natural to reason that tailoring a pretrained model to the domain of a target task may improve the performance. Studies have reported the second phase of pretraining on data in task domains can improve the performance in classification tasks [15], named entity recognition (NER) and speech rendition tasks [29], and QA tasks [30]. A recent study even proposed that for domains with sufficient data such as the biomedical field, skipping the general domain pretraining process can lead to performance gains over a model pretrained first on general-domain data and subsequent task-domain data [31].

2.4 Finetuning on Few Examples

BERT has been known to have instability in finetuning [32]. Studies have shown that when finetuning on a small dataset, the randomness introduced in BERT training processes such as weight initialization and training data order can cause significant perturbation in results [33]. In one recent publication, the authors demonstrated that some layers of the BERT network may be inferior starting points for fine-tuning and re-initializing these layers may improve the performance. The same study also suggested ignoring gradient bias correction and insufficient training time may also lead to instability [14]. A few studies also proposed a few techniques to improve the finetuning stability, such as finetuning on a large intermediate task before finetuning on a small dataset [34] and using an innovative regularization method [35].

3 Approach

3.1 Baseline Model

We built our QA system based on the DistilBERT model [36]. DistilBERT is a general-purpose language representation model which is derived from BERT [32] but it is smaller and faster while maintaining a similar level of language understanding capabilities. The model used in this project has been pretrained by HuggingFace Inc. [37] and made available to the public. This version of DistilBERT has 6 layers, 768 hidden units, 12 heads, and 66M parameters and was distilled from a BERT model checkpoint trained on lower-cased English text [38]. Our baseline model is created by further finetuning it on the three in-domain datasets described in the experimentation section with default parameters as the project guideline advised.

3.2 Methods to Improve Robustness

In this study, a few different approaches are explored to improve the robustness of the QA system on out-of-domain data, including domain adaptive pretraining, data augmentation, and domain adversarial training and adjusting the finetuning process for few examples.

3.2.1 Data Augmentation

We have explored two different types of data augmentation.

First, the questions in the out-of-domain dataset were augmented through back-translation. This method was inspired by [21]. Specifically, using the established online translation service Google Translate, we translate each of the questions in the three out-of-domain datasets into French and then translate them back into English. This method provides us with a paraphrase of the original question prompt, holding the context and answer constant.

Second, the contexts in the out-of-domain dataset were augmented through alignment shifting [20]. Using this method, we want to make sure that when unrelated contexts are deleted, our model can still accurately find the correct answer from the context. We first went over the questions and answers and find out where the answers are located in the contexts. We then randomly select chunks of tokens that do not appear in answers or between answers and randomly truncate these contexts out.

Since these two methods deal with different parts of data and thus independent of each other, we have crossed applied them to generate the augmented out-of-domain datasets.

3.2.2 Domain Adversarial Training

To adopt the adversarial training, we referred to the research code provided by [27] and modified the adversarial training model part and integrate it into our codebase. As described in the previous section, the discriminator tries to correctly classify the domain based on the joint embedding of context and questions. During training, the objective of the QA model is not only to minimize the negative loglikelihood of the predictions for start position and end position but also to confuse the discriminator model as much as possible. The overall optimizing objective becomes minimizing the sum of the QA loss and a weighted Kullback-Leibler (KL) divergence between uniform distribution over domain classes K and the discriminator's prediction, as in

$$L_{adv} = L_{QA} + \lambda \cdot KL(Uniform(K)||Pred_{discriminator}).$$

3.2.3 Domain Adaptive Pretraining

To accomplish domain adaptive pretraining (DAPT), the DistilBERT model pretrained by HuggingFace was further pretrained on domain-specific MLM task data. Here we create the MLM task training and validation dataset from the three in-domain datasets (SQUAD, NewsQA, and Natural Question).

To create the MLM task datasets, the questions and passages in these QA datasets were concatenated and random masking was then applied to the sequence, with each word having a certain probability p_m of being masked. In the next step, the masked words can either be replaced by a [MASK] token with probability p_1, or a random word from the dictionary with probability p_2, or keep as original with probability p_3. We referred to the masked data generation function in the research code provided by [15] to create the MLM task dataset into our project. After the DAPT process, the obtained model can be saved and further finetuned with QA task-specific data, same as models without this process.

3.2.4 Finetuning on Few Sample

As discussed in Related Work section, many factors can influence the results of finetuning on few samples. As the time is limited, we only experimented on the effects of freezing all layers in the pretrained BERT model except for the output layer.

4 Experiments

4.1 Data

The project intends to improve the model's ability to generalize from in-domain datasets, which are abundant, to out-of-domain datasets, which are scarce. As shown in Table 1, we have three in-domain datasets and three out-of-domain datasets. The number shown in the table is the number of questions in the dataset. Multiple questions can share the same context. Each question is followed by three crowd-sourced answers, each of which have two numerical numbers indicating the start and end position of the extracted answer, respectively.

Dataset	Question Source	Passage Source	Train	Val	Test
in-domain datasets					
SQuAD [39]	Crowdsourced	Wikipedia	50000	10507	-
NewsQA [40]	Crowdsourced	News articles	50000	4212	-
Natural Questions [41]	Search logs	Wikipedia	50000	12836	-
oo-domain datasets					
DuoRC [42]	Crowdsourced	Movie reviews	127	126	1248
RACE [43]	Teachers	Examinations	127	128	419
RelationExtraction [44]	Synthetic	Wikipedia	127	128	2693

Table 1: **Statistics for datasets used in this project, borrowed from [45]**

4.2 Evaluation method

We used two widely used evaluation metrics: Exact Match (EM) and F1 score.

- Exact Match is a binary 0/1 metric that says whether the answer matches exactly to the ground truth. It is a stricter measure of the two.
- F_1 score is the harmonic mean of precision and recall:

$$F_1 = \frac{\texttt{true positive}}{\texttt{true positive} + 1/2(\texttt{false positive} + \texttt{false negative})}.$$

  It is the less strict measure of the two.

4.3 Experimental details

Figure 1 shows the general experiment flow chart. By opting for whether incorporating a specific technique at each decision point, we created different combinations of robustness-improving techniques.

All experiments were conducted on the Pytorch (version 1.7.1) deep learning framework [46] and in Python 3.6.8 environment. We trained the models on NC6 virtual machine (VM) of Microsoft Azure Platform [47] with one single K80 GPU. The AdamW [48] optimizer, which has been shown one of the fastest optimizers in recent years, was used in all model training,

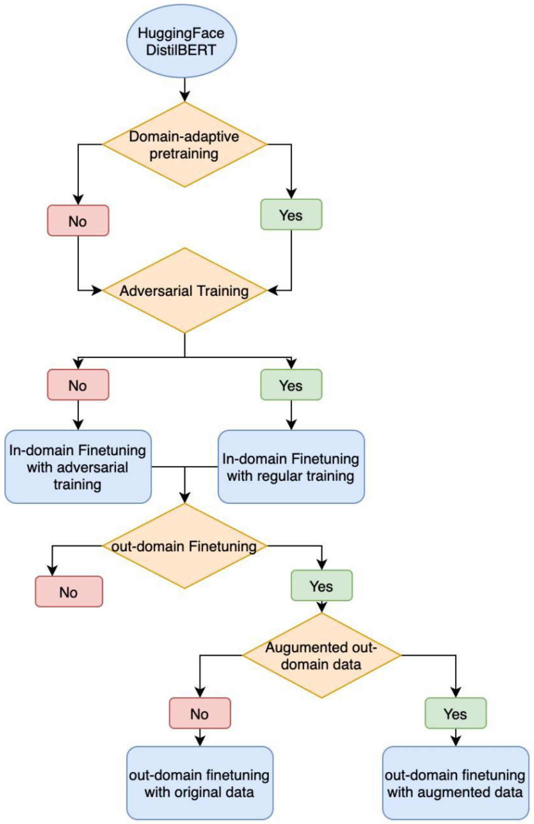
HuggingFace DistilBERT
Domain-adaptive pretraining
No
Yes
Adversarial Training
No
Yes
In-domain Finetuning with adversarial training
In-domain Finetuning with regular training
out-domain Finetuning
No
Yes
Augumented out-domain data
No
Yes
out-domain finetuning with original data
out-domain finetuning with augmented data

Figure 1: **Flowchart of experiments**

For model finetuning on in-domain data, we used the default hyperparameter setting of a batch-size of 16, an epoch number of 3, and a learning rate of 3×10^{-5}. Model performance on validation dataset was evaluated every 5000 gradient steps, and the model with the best performance was saved for further analysis and finetuning.

For finetuning on out-of-domain data, we used a batch-size of 16, an epoch number of 3, and a decreased learning rate of 3×10^{-6}. Since the out-of-domain data is scarce, model performance on the validation dataset was evaluated every 20 gradient steps, and the model with the best performance was saved for further analysis.

To implement adversarial training, a discriminator was added to the same pretrained DistilBERT as in the baseline model. The discriminator is composed of three repeat components that consist of a linear layer of 768 units, one ReLU layer and a dropout layer (dropout rate = 0.1), and one softmax output layer for domain classification. The weight of discriminator loss λ was set as the default of 0.5. We conducted a search on the hyperparameters and structure of the discriminator, including the λ, dropout rate, and the number of hidden units, but did not find a setting yielding better performance than the default proposed by the original research paper (result not shown). For adversarial training, finetuning on the in-domain dataset or out-of-domain dataset is following the same settings as mentioned above.

To achieve domain adaptive learning, the DistrilBERT model pretrained by HuggingFace was further pretrained using masked language modeling (MLM) on the three in-domain datasets before any QA task-specific finetuning. The in-domain QA datasets were wrangled to generate MLM task datasets. The process includes first masking random words with masking probability $p_m = 0.15$, and then replacing the masked words by [MASK] token with $p_1 = 0.8$, by random other words with $p_2 = 0.1$, or keeping them as original tokens with $p_3 = 0.1$. This setting is used in the original research paper by [15]. The MLM model was also trained with AdamW optimizer and a batch-size of 16, an epoch number of 3, and a learning rate of 3×10^{-5}. The model performance on the validation dataset was evaluated every 5000 gradient steps, and the model with the least MLM loss was used for the downstream QA task-specific finetuning as shown above.

4.4 Results

	DAPT-						DAPT+					
	Adv-			Adv+			Adv-			Adv+		
	F-	F+		F-	F+		F-	F+		F-	F+	
		A-	A+		A-	A+		A-	A+		A-	A+
F1	47.72	48.92	48.92	48.51	48.71	48.71	47.09	48.15	48.15	49.09	**50.61**	50.38
EM	30.63	31.68	31.68	33.51	34.29	34.29	30.63	32.46	32.46	33.77	35.08	**35.34**

Table 2: **Experiment results based on F1 and EM scores. A combination of different robustness-improving techniques was tested. The first column is the baseline model trained following the project guideline.** DAPT +/-: whether the model was further pretrained using in-domain data on masked language model; Adv +/-: whether the model with adversarial training; F +/-: whether finetuning was conducted using out-of-domain data; A +/-: finetuning using augmented or original out-of-domain data

Part of the results from the experiments is shown in Table 2. The performance of models, which are trained with or without each robustness-improving technique, are measured based on the F1 and EM scores on out-of-domain validation datasets. Since we discovered that freezing all layers but the output layers consistently produced inferior results, finetuned models shown in Table 2 are without freezing layers during the process.

We observe that each technique may generate a better or worse performance compared to the baseline model (DAPT-, Adv- and F-), which has an F1 score of 47.72 and an EM score of 30.63. For example, only using DAPT causes a decreased F1 score (47.09) and unchanged EM sore. Using adversarial training alone can improve both scores (F1: 48.51; EM: 33.51) and this improvement generally holds when combined with other methods. Interestingly, finetuning on the augmented out-of-domain data produced the same performance score in most experiments when compared to its original out-of-domain counterparts, except when combining with domain adaptive pretraining and adversarial training. But we did observe it produced slightly better performance in the situation of freezing all layers during finetuning (results not shown). We think it suggests a combination of DAPT and adversarial training make the model more prepared to computationally figure out the invariant information in the augmented data.

Overall, our experiment demonstrated a stronger improvement in model robustness when combining individual techniques. The combination of DAPT, adversarial training, and finetuning with augmented out-of-domain data produced the best EM score, 35.34. The best F1 score is achieved when the combination of DAPT and Adversarial training and finetuning with original out-of-domain data, which is 50.61. However, both F1 (50.61 vs 50.38) and EM (35.08 vs 35.34) are pretty close in these two settings.

Finally, we evaluated the performance of the two best models above on the test dataset. Surprisingly, we observed the performance is less desirable. The model combining all robustness-improving methods (DAPT+, Adv+, F+, and A+) generated an EM score of 41.19 and an F1 score of 58.46, and the other model (DAPT+, Adv+, F+ and A-) generated an EM score of 41.12 and an F1 score of 58.50. Both models have decreased F1 scores and only minor improvement in EM score compared to the baseline model (F1: 59.25; EM: 40.07). We think this could be due to the fact that out-of-domain training data size is too small compared to test data size. Such small data size creates large variance in the result. We reason that from the observation that the baseline model also has drastically different F1 and EM scores on the test dataset. Consequently, optimization of the performance on the validation test set can not guarantee the optimized performance on the test dataset.

5 Analysis

We conduct some case studies here to analyze the predictions produced by our model. When people see a correct prediction by models on a QA task, they usually tend to believe that the models have

Question: Due to which disease did Julius Garfinckel die? *Context*: Julius Garfinckel died on his 64th birthday of pneumonia in Washington, D.C. ... *Answer*: pneumonia *Prediction*: died on his 64th birthday of pneumonia
Question: Which musical instrument is connected with Neumeister Collection? *Context*: The Neumeister Collection is a manuscript compilation of chorale preludes for organ ... *Answer*: organ *Prediction*: chorale preludes for organ

Table 3: Wrong predictions that overlap with the correct answer

Question: Who suggests they kidnap the daughter of an executive that fired Ryu? *Context*: ... Ryu and his girlfriend, Yeong-mi conspire to kidnap the daughter of the boss who fired him ... they decide to kidnap Yu-sun, the daughter of the boss's friend, Park Dong-jin ... *Answer*: Yeong-mi *Prediction*: Park Dong-jin
Question: Who is the book, Be Happy at Work, written for? *Context*: ... Joanne Gordon does. She is the author of Be Happy at work ...She wants to help people who do not feel satisfied with their jobs find work that is good for them ... *Answer*: people who do not feel satisfied with their jobs *Prediction*: Joanne Gordon

Table 4: Wrong predictions that are of the right type

human-like capabilities of understanding natural languages. The following two types of errors we observe contradict this idea and illustrate the good and the bad of our models.

First, some wrong predictions have overlap with the correct answer, but they fail to answer the question. Table 3 provides two such examples. Here, our model is able to find the correct sentence where the answer appears. However, it fails to address the exact answer posed by the questions, namely "disease" in the first question and "musical instrument" in the second question. One explanation would be that our model is able to find the approximate place for an answer, but fails to reason and select the precise words that answer the question.

Second, some other wrong predictions are of the correct type, but meaningless in the contexts. Table 4 provides two of such examples. Here, our model predicts the right type of answer in both cases, namely the name of a person, to the question of "Who". Wrong predictions fall into the same section as the correct answers. However, as multiple entities of the correct type appear in the nearby contexts, our models fail to “reason" the relationship between these entities.

Overall, we doubt people's belief that the model has actually the ability to reason. One explanation for such error is that our model is able to locate the piece of possible answer and find out the entity of the correct type to answer the question. It is essentially doing pattern matching.

6 Conclusion

In this project, we successfully implemented a selected number of techniques, including adversarial training, domain adaptive pretraining, data augmentation, and finetuning on out-of-domain data that were demonstrated to improve the robustness of the deep learning QA system. We quantified the effects of each technique and different combinations of them on the performance of the out-of-domain QA task using the F1 score and EM score.

We find that while each method may bring mixed result compared to the baseline model, a combination of them generally lead to a greater performance of the QA system on out-of-domain tasks. The greatest performance on the EM score was achieved when all four techniques were combined and the greatest F1 score was achieved when three methods (adversarial training, finetuning, and DAPT) were combined. We also briefly touched on the importance of finetuning setting when working with few sample out-of-domain data, where we demonstrated that the choice of freezing most layers during finetuning may negatively influence the effect.

We believe this work of horizontal comparison is innovative, as we have not seen a similar analysis elsewhere. This work lays a solid ground for future study of comparing these robustness-improving techniques and helps researchers to develop better strategies to improve the QA system.

One of the limitations is the scope of the project does not allow us to conduct larger and deeper experiments. With each technique we applied here, there could be months of work of adjustment to achieve the best performance. However, time does not allow us to do that. We only attempted to conduct limited hyperparameter and model structure search for adversarial training, but it did yield positive results. We ended up with default parameters as the original research paper suggested most of the time.

A dark cloud here is the observed discrepancy between the performance on the out-of-domain validation and test datasets. Although we reasoned it is probably due to the small sample size of out-of-domain training and validation data causing large variance in the result, a detailed analysis is highly desirable. But this effort also requires some labels from the test dataset, which is not provided here.

As part of the future work, we would like to analyze the inconsistent behavior of the model on the validation and test datasets, if provided needed data. Another direction is to evaluate the individual and synergetic effects of more techniques on QA system robustness. We also would like to dive deeper into optimizing each technique for our task, which includes trying different data augmentation strategies and variations of creating MLM task dataset.

References

[1] B. Green, A. K. Wolf, Carol L. Chomsky, and K. Laughery. Baseball: an automatic question-answerer. In *IRE-AIEE-ACM '61 (Western)*, 1961.

[2] William A. Woods. Lunar rocks in natural English: Explorations in natural language question answering. In Antonio Zampolli, editor, *Linguistic Structures Processing*, pages 521–569. North Holland, Amsterdam, 1977.

[3] Yashvardhan Sharma and Sahil Gupta. Deep learning approaches for question answering system. *Procedia Computer Science*, 132:785–794, 2018. International Conference on Computational Intelligence and Data Science.

[4] Silvia Quarteroni. A chatbot-based interactive question answering system.

[5] A. Kadam, S. Joshi, S. Shinde, and S. P. Medhane. Question answering search engine short review and road-map to future qa search engine. *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, pages 1–8, 2015.

[6] YongGang Cao, Feifan Liu, Pippa Simpson, Lamont Antieau, Andrew Bennett, James J. Cimino, John Ely, and Hong Yu. Askhermes: An online question answering system for complex clinical questions. *Journal of Biomedical Informatics*, 44(2):277–288, 2011.

[7] R. Puri, Ryan Spring, M. Patwary, M. Shoeybi, and Bryan Catanzaro. Training question answering models from synthetic data. *ArXiv*, abs/2002.09599, 2020.

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

[9] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *NAACL-HLT*, 2018.

[10] Danqi Chen, A. Fisch, J. Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *ACL*, 2017.

[11] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87, 03 1991.

[12] Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Y. Matsumoto. Adversarial training for cross-domain universal dependency parsing. In *CoNLL Shared Task*, 2017.

[13] Seanie Lee, Donggyu Kim, and Jangwon Park. Domain-agnostic question-answering with adversarial training. In *MRQA@EMNLP*, 2019.

[14] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning. *ArXiv*, abs/2006.05987, 2020.

[15] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. *ArXiv*, abs/2004.10964, 2020.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *ACL*, 2018.

[17] Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *EMNLP/IJCNLP*, 2019.

[18] Jean-Philippe Corbeil and Hadi Abdi Ghadivel. Bet: A backtranslation approach for easy data augmentation in transformer-based paraphrase identification context. *ArXiv*, abs/2009.12452, 2020.

[19] Jason Wei and K. Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *ArXiv*, abs/1901.11196, 2019.

[20] Zach Ryan and M. Hulden. Data augmentation for transformer-based g2p. In *SIGMORPHON*, 2020.

[21] 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.

[22] Yaroslav Ganin, E. Ustinova, Hana Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *ArXiv*, abs/1505.07818, 2016.

[23] Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[24] Connor Shorten and T. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 2019.

[25] Xilun Chen, Y. Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Q. Weinberger. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570, 2018.

[26] Yi Wu, David Bamman, and S. Russell. Adversarial training for relation extraction. In *EMNLP*, 2017.

[27] Seanie Lee, Donggyu Kim, and Jangwon Park. Domain-agnostic question-answering with adversarial training. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 196–202, Hong Kong, China, November 2019. Association for Computational Linguistics.

[28] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[29] Leonard Konle and Fotis Jannidis. Domain and task adaptive pretraining for language models. In *CHR*, 2020.

[30] Nina Poerner, Ulli Waltinger, and Hinrich Schutze. Inexpensive domain adaptation of pretrained language models: Case studies on biomedical ner and covid-19 qa. In *EMNLP*, 2020.

[31] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ArXiv*, abs/2007.15779, 2020.

[32] 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.

[33] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *ArXiv*, abs/2002.06305, 2020.

[34] Jason Phang, Thibault Févry, and Samuel R. Bowman. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *ArXiv*, abs/1811.01088, 2018.

[35] Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. *ArXiv*, abs/1909.11299, 2020.

[36] 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.

[37] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[38] MS Windows NT kernel description. `https://huggingface.co/transformers/pretrained_models.html`. Accessed: 2010-09-30.

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] Adam Paszke, S. Gross, Francisco Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, B. Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[47] C. Kotas, Thomas Naughton, and N. Imam. A comparison of amazon web services and microsoft azure cloud platforms for high performance computing. *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4, 2018.

[48] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.