Multi-Phase Adaptive Pretraining on DistilBERT for Compact Domain Adaptation

Stanford CS224N Default Project

Anthony Perez
Department of Computer Science
Stanford University
aperez01@stanford.edu

Abstract

While modern natural language (LM) models such as transformers have made significant leaps in performance relative to their predecessors, overfitting to the in-domain training set remains a key problem, especially in low-resource settings. As a result, such models fail to generalize to out-of-domain data, thus hampering performance in real-world cases where data is not independently and identically distributed (IID). According to Gururangan et al. (2020) [1], the use of domain-adaptive pretraining (DAPT), which involves pretraining on unlabeled out-of-domain data, and task-adaptive pretraining (TAPT), which entails pretraining on all of the unlabeled data of a given task, can dramatically improve performance on the large RoBERTa model when the in-domain and out-of-domain data distributions have a small amount of overlap. Consistent with the Robust QA track of the default project, this report aims to investigate and test the hypothesis that TAPT in tandem with DAPT can improve out-of-domain performance on smaller transformers like DistilBERT for the question answering task, especially in the presence of domain shift. The final results suggest that the use of TAPT can lead to a slight increase in Exact Match (EM) performance without DAPT. However, applying DAPT, even with the use of word-substitution data augmentation, significantly degrades the performance of the model on the held-out out-of-domain dataset.

1 Key Information to include

- Mentor: Yuyan Wang
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

In the modern-era of natural language processing (NLP), pretrained transformer models are capable of achieving near-human performance on a variety of tasks and domains even without the need to finetune on additional data. However, the fact that such models train millions of parameters with even the smallest transformers means that overfitting to the original domain is common, which can lead to performance issues after finetuning. According to Jia and Liang (2017) [2], one explanation for this phenomena is that models are sensitive to pertubations within the training set, allowing them to learn extraneous correlations that can hurt performance in the presence of adverserial examples. This implies that such models fail to generalize to out-of-domain data, especially in low-resource settings.

2.1 Multi-Phase Adaptive Pretraining

Given that in-domain and out-of-domain data distributions tend to diverge in practical NLP applications, it is important to find methods that can overcome the overfitting problem with only a very small subset of the entire out-of-domain dataset. One potential solution offered by Gururangan et al. (2020) [1] is known as *adaptive pretraining*. On one hand, domain-adaptive pretraining (DAPT) utilizes a large unlabaled corpus from a new target domain to initiate a second stage of pretraining on an already pretrained transformer model. In return, the model is expected to generalize its predictive behavior to the new domain in order to improve its performance on the chosen downstream task. On the other hand, task-adaptive pretraining (TAPT) initiates further pretraining by utilizing unlabeled data from the task distribution. This enables the model to learn more task-relevant parameterizations in a fairly inexpensive manner, preparing the model to train toward a more optimal representation for the task. When these two techniques are combined such that DAPT is followed by TAPT, this is known as multi-phase adaptive pretraining (MAPT).

Although Gururangan et al. [1] claims that MAPT applies to both high-resource and low-resource settings, the smallest dataset they used had thousands of unique training datapoints. Thus, while the use of MAPT appears to be successful in generalizing LMs to new domains, it is unknown whether this approach will apply to truly low-resource settings (i.e. less than 500 unique examples for the target domain). Moreover, Gururangan et al. applies MAPT to RoBERTa, a large pretrained transformer proposed by Liu et al. (2019) [3] that can already adapt to new domains and tasks quite well on its own. Although the authors claim that applying MAPT should work for all pretrained models, it still remains to be seen whether this claim actually holds, especially with lighter models such as DistilBERT from Sanh et al. (2020) [4]. Ideally, MAPT should lead to a performance increase with DistilBERT in settings where computational efficiency is emphasized.

2.2 Summary of Approach and Results

To investigate whether MAPT can be used to successfully generalize to a low-resource out-of-domain dataset with DistilBERT, this report approaches the problem by tracking MAPT's performance on the question-answering (QA) task given three large in-domain datasets and three small out-of-domain datasets. In order to compensate for the lack of sufficient out-of-domain data, word-substitution data augmentation–as proposed by Wei and Zhou (2019) [5]–is used during the DAPT phase to boost the number of examples that are seen by the model. Despite the implementation of these techniques, TAPT only seems to lead to marginal improvements in performance on the held-out out-of-domain dataset, and this improvement is only seen in the validation set when DAPT is not applied beforehand. On the other hand, regardless of whether TAPT is applied, finetuning the in-domain baseline with or without DAPT degrades performance by a nontrivial amount. This implies that in the presence of domain shift, QA with MAPT is not robust to changes in the data distribution when the model does not have enough unique examples in the training set to generalize to.

3 Related Work

While the analysis provided by Gururangan et al. [1] offers the most complete overview of MAPT, there have been several works prior that focus on various aspects of MAPT, such as DAPT, TAPT, transfer learning for domain adaptation, and how to classify domains.

3.1 DAPT and TAPT

In 2019, Han and Eisenstein [6] first identified DAPT as a means of boosting LM performance when the target domain diverges from the pretraining corpus. In particular, they focus on a scenario in which there are three sources of data: the original pretraining corpus, an unlabeled corpus for the target domain, and a separate labeled dataset for the downstream task. Interestingly, the authors found that this unsupervised form of DAPT substantially improves BERT on the sequence labeling task. Nonetheless, this form of DAPT differs slightly from the DAPT technique mentioned by Gururangan et al. [1] since it utilizes task data that does not come from the target domain. In fact, it shares more in common with TAPT than it does with DAPT.

In addition to the work done by Han and Eisenstein, research conducted by Sun et al. (2019) [7] suggests ways to finetune BERT on text classification tasks, noting that pretraining BERT with target domain data using a masked language model is an effective strategy. They also propose cross-domain pretraining, which occurs when the model is pretrained with the union of target and non-target domain data. This, of course, is similar to the definition of DAPT that is proposed by Gururangan et al. [1].

3.2 Transfer Learning

Aside from various iterations of DAPT and TAPT, another way to generalize models to out-of-domain data is to use transfer learning. For instance, Alsentzer et al. (2019) [8] suggests that starting with domain-specific pretraining models, such as in the clinical domain, leads to state-of-the-art results in domain-specific tasks. Thus, if a pretrained model with a large amount of domain overlap already exists, it is recommended that such a model is used as a baseline instead of a base model like BERT. However, given that a domain-specific model does not already exist for the out-of-domain QA datasets used for this project, the next best option is to start pretraining from the base DistilBERT model.

3.3 In-Domain vs. Out-of-Domain

Finally, given that the term "domain" has been used vaguely in a variety of contexts, some work has been done in narrowly defining what it means for a cluster of data to be "in-domain" or "out-of-domain". In particular, Aharoni and Goldberg (2020) [9] discover that what researchers commonly regard as a domain can be defined using the average-pooled hidden-state sentence representations produced by a transformer model. Thus, the approach that I use in this report assumes that the hidden-state representations of the out-of-domain data provided is clustered closely enough to be treated as a single target domain.

4 Approach

First, note that this project utilizes the default project starter code as a foundation to performing key tasks such as processing data, training the model, and saving model parameters for future use. Thus, the baseline referred to in this report is the same baseline detailed in the default final project handout (e.g. base DistilBERT with finetuning on the in-domain dataset).

As shown in Figure 1, the approach to MAPT can be separated into four main stages: data processing with augmentation, DAPT/TAPT, and QA finetuning. Each one can be described as follows:

4.1 Data Processing with Augmentation

Let q, p, and a denote an arbitrary question, context paragraph, and answer, respectively. Consider an in-domain dataset $\mathcal{D}_{ID}$ and an out-of-domain dataset $\mathcal{D}_{OOD}$ where for $(x, y) \in \mathcal{D}_{ID} \bigcup \mathcal{D}_{OOD}$, $x = (q, p)$ and $y = a$. Since DAPT will be performed on only out-of-domain data, I create more out-of-domain examples to create the augmented out-of-domain dataset $\bar{\mathcal{D}}_{OOD}$. In particular, for any $(x, y) \in \mathcal{D}_{OOD}$ such that $x = (q, p)$ and $y = a$, I augment the data by substituting a word for a synonym in q and p with probability $\mathcal{P}$ for n runs to create n new datapoints $\mathbf{x} = \{(q'_1, p'_1), \ldots, (q'_n, p'_n\})$. Then, given that the context paragraph has been altered, I edit y accordingly to create n new datapoints $\mathbf{y} = \{a'_1, \ldots, a'_n\}$, thus ensuring that $\mathbf{x}, \mathbf{y} \in \bar{\mathcal{D}}_{OOD}$. The technique that is used to perform these augmentations is detailed further by Wei and Zhou [5]. To convert the strings into augmented strings with accurate synonyms, I use the Paraphrase Database (PPDB) as highlighted

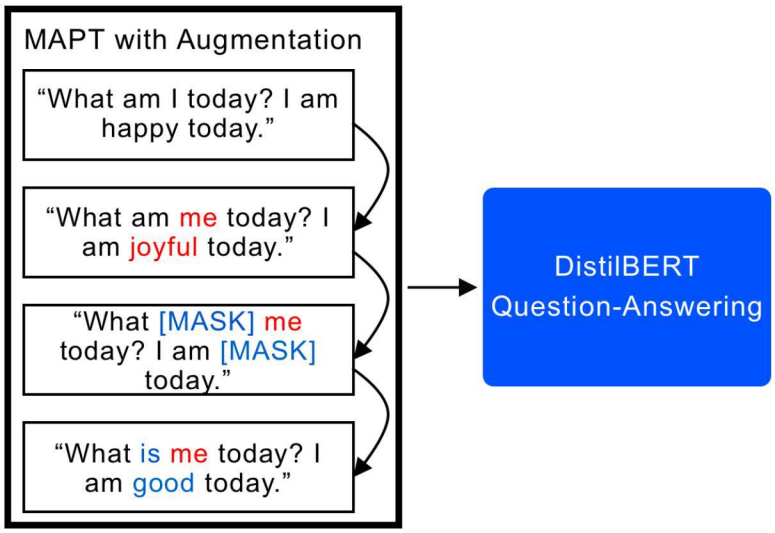

Figure 1: MAPT consists of data augmentation followed by masked language modeling.

in Ganitkevitch et al. (2014) [10] and Pavlick et al. (2015) [11] to fetch the correct synonyms with the help of the NLPAug library created by Edward Ma (2019) [12]. This is done (as opposed to filling words with `[MASK]` tokens) to ensure that the word substitutions are accurate.

4.2 DAPT

Next, I use the union of the original and augmented out-of-domain training datasets $\mathcal{D}_{OOD}^{train*} = \mathcal{D}_{OOD}^{train} \bigcup \tilde{\mathcal{D}}_{OOD}^{train}$ to pretrain the base DistilBERT model towards the target domain. To do so, I first create a new model for masked language modeling (MLM) using the base uncased DistilBERT parameters without a span classification head. Then, given a batch of question-context pairs generated from $\mathcal{D}_{OOD}^{train*}$, I take each (q, p) string in the batch and convert the words to `[MASK]` that are not `[CLS]`, `[SEP]`, and `[UNK]` tokens with probability 0.15, removing `[PAD]` tokens during the process.

Finally, I tokenize the batch of masked strings to create a new set of embeddings with a maximum length of 384 for each sequence, thus re-padding the sequences with the corresponding attention masks. Let $\mathbf{b}_{input}$ be this new batch of input embeddings. The last step is to input this batch into the MLM model with the original unmasked batch of strings as a target $\mathbf{s}_{target}$ to yield the MLM loss $\mathcal{L}_{MLM}(\mathbf{b}_{input}, \mathbf{s}_{target})$, which is then optimized by Adam to update the parameters of DistilBERT. This process is repeated for subsequent batches until some number of epochs is complete. When training is complete, I save the parameters θ for the next step.

4.3 TAPT

To perform TAPT, I use the union of the original in-domain and out-of-domain training sets $\mathcal{D}^{train} = \mathcal{D}_{ID}^{train} \bigcup \mathcal{D}_{OOD}^{train}$ to pretrain the DAPT DistilBERT model towards the desired task distribution. To do so, I use an identical set of steps as DAPT to create an MLM model with the saved parameters θ, optimize the MLM loss $\mathcal{L}_{MLM}$ for some number of epochs using Adam, and then save the updated θ for the next (and final) step.

4.4 QA Finetuning

In order to perform the QA task on the out-of-domain data, I use $\mathcal{D}^{train}$ to finetune the MAPT model. To do so, I create a DistilBERT QA model with a span classification head using the saved parameters θ. Then, as the default project handout suggests, a batch $\mathbf{b}_{input}$ of embeddings is taken from $\mathcal{D}^{train}$ along with a batch of true start and end positions $\mathbf{p}_{target} = (\mathbf{p}_{start}, \mathbf{p}_{end})$ and subsequently passed into the model to compute the QA loss $\mathcal{L}_{QA}(\mathbf{b}_{input}, \mathbf{p}_{target})$, which is the cross-entropy loss for the predicted start and end locations of the answers in the batch. Finally, $\mathcal{L}_{QA}$ is used by an Adam

optimizer to update the parameters θ. This process is repeated for subsequent batches until some number of epochs is complete. Then, θ is saved for evaluation.

Observe that the entire training set (both in-domain and out-of-domain) is trained at once. Experimentally, doing this (as opposed to training them separately) seems to yield almost no difference in performance as long as each domain is trained for the same number of epochs.

5 Experiments

For the configurations below, assume that all aspects of the experiments are identical to the ones listed in the default project handout unless specified otherwise.

5.1 Data

Consistent with the default project handout, the experiments performed use the larger in-domain datasets of SQuAD [13], NewsQA [14], and Natural Questions [15] along with the smaller out-of-domain datasets of DuoRC [16], RACE [17], and RelationExtraction [18]; the out-of-domain datasets are augmented during the experiments. As expected, the data splits between training, validation, and testing are identical to the default project handout as well.

5.2 Evaluation method

The primary metrics I used to evaluate the performance of the experimental models are the Exact Match (EM) score and F1 score of the out-of-domain data, since the goal is to improve performance on the target domain. As an auxiliary evaluation metric, I also analyze the QA training loss over epochs during the finetuning stage to ensure that the models are learning the intended objective (which they all do). For the analysis, I also use the visualized outputs of the QA models to qualitatively evaluate their performance.

5.3 Experimental details

While several experiments have been run of varying quality, the five models that are listed in the results below are `baseline`, `finetune-ood`, `dapt`, `tapt`, and `mapt`. Each of their specifications are listed below:

`baseline`: A base uncased DistilBERT model that is finetuned on the in-domain training set.

`finetune-ood`: A `baseline` model that is finetuned on the out-of-domain training set.

`dapt`: A base uncased DistilBERT model that applies DAPT with augmented out-of-domain training data and subsequently finetuned on the in-domain and out-of-domain training sets.

`tapt`: A base uncased DistilBERT model that applies TAPT with in-domain and out-of-domain training data and subsequently finetuned on the in-domain and out-of-domain training sets.

`mapt`: A based uncased DistilBERT model that applies DAPT followed by TAPT, then subsequently finetuned on the in-domain and out-of-domain training sets.

For the QA finetuning stage of every model, training occurred over 3 epochs. Additionally, all of the models were trained using the same learning rate $\alpha = 0.00003$, random seed 42, batch size 16, and maximum length of a predicted answer $L_{max} = 15$. For `dapt` and `mapt`, the DAPT stage occurred over 20 epochs. For `tapt` and `mapt`, the TAPT stage occurred over 3 epochs. For data augmentation, the probability of a word being replaced was $\mathcal{P} = 0.1$ and the number of examples created per question-context pair (q, p) was $n = 16$, which are hyperparameters that are both consistent with the optimal results found in Wei and Zhou's paper on easy data augmentation (EDA) [5].

In terms of training time, DAPT, TAPT, and QA finetuning ran in about 120, 210, and 170 minutes respectively. All of the experiments were run with an `NC6_v3` virtual machine on Azure.

As a note, all features (i.e. tokenized representations) were recomputed after every experiment to ensure that the correct data was being applied depending on whether data augmentation was used or not.

5.4 Results

The F1 and EM scores for the out-of-domain validation and test sets are listed in Table 1 and Table 2 respectively. *Observe that the F1 score for the test set on the baseline is an estimate since only four submissions are allowed in the RobustQA track:

Table 1: Validation EM and F1 scores for the `baseline`, `finetune-ood`, `dapt`, `tapt`, and `mapt` models.

Model	Baseline	Finetune-OOD	DAPT	TAPT	MAPT
EM	33.25	31.94	30.10	**35.08**	31.68
F1	**48.43**	47.22	45.05	47.72	47.10

Table 2: Test EM and F1 scores for the `baseline`, `finetune-ood`, `dapt`, `tapt`, and `mapt` models.

Model	Baseline*	Finetune-OOD	DAPT	TAPT	MAPT
EM	–	**40.51**	39.75	40.44	39.13
F1	59.19	**59.24**	56.92	58.66	57.32

As the data in Table 1 and Table 2 suggests, there are little to no changes in performance between the `baseline` model and the `finetune-ood` model at validation and test time. This pattern seems to hold for the `tapt` model as well despite the increase in EM score on the validation set. However, this slight uptick in performance can likely be attributed to the stochasticity of the data generated from the out-of-domain distribution. At a minimum, this result shows that using masked-language modeling on the unlabeled training set does not negatively impact the model's ability to predict exact start and end locations for answers within a given context, which is expected behavior.

More surprisingly, however, the `dapt` and `mapt` models seem to perform considerably worse than the `baseline` model in the out-of-domain validation and test sets despite their objective of generalizing to the target domain. The most straightforward explanation for this lack of performance is that DAPT fails to train a model toward the target domain in low-resource settings due to a lack of sufficient examples to learn from. Data augmentation should help with this issue in practice, but the fact that there are so few examples means that the augmented datapoints create noise that does not help the model learn new representations. Another key concern is that unlike the datasets shown in Gururangan et al. [1], the data provided for each domain is a subset of data provided for the task. Typically, DAPT is useful in cases where there are vast amounts of unlabeled data within the domain of a problem but outside the labeled dataset. For the out-of-domain datasets, there is no additional unlabeled training data offered. Finally, another potential reason why MAPT as a whole does not lead to the performance boost that is expected is the presence of domain shift–a phenomenon that occurs when the training, validation, and test data distributions diverge for a given task. This is especially important in the analysis of the test set performance since the DuoRC [16], RACE [17], and RelationExtraction [18] datasets offer an uneven number of contributions at test time. Regardless of the reasoning, these results are generally worse than I expected, even despite the aforementioned dataset issues. Because of this, the answer to whether MAPT on DistilBERT can result in similar performance boosts to large models like RoBERTa is inconclusive.

6 Analysis

To understand why MAPT does not result in a performance increase on the target domain, it is useful to look at individual outputs of the model to see where its behavior diverges from the expected result. As an example, consider the following question-context pair along with the true and predicted answer:

Question: What was the judge's verdict?

Context: NEW YORK (CNN) – Charges have been dropped against four men accused of raping an 18-year-old student at Hofstra University after the woman recanted her allegations, prosecutors said...

...A judge dismissed all charges Wednesday night and ordered the release of the four men – Jesus Ortiz, 19; Stalin Felipe, 19; Kevin Taveras, 20; and Rondell Bedward, 21; all of the New York metropolitan area, according to Nassau County, New York, District Attorney Kathleen Rice...

Answer: dismissed all charges

Prediction: dismissed all charges Wednesday night and ordered the release of the four men

From the prediction above, it is evident that neither of the EM or F1 scores would be improved by this example despite the nearly identical semantic relationship between the true answer and the prediction. Of course, this suggests that quantitative metrics like EM and F1 scores are not always going to be the best indicator of model output quality for every example. Thus, while the quantitative performance decrease of MAPT seems significant, there is a possibility that such results are not capturing the full scope of the MAPT model's performance.

Now, consider an example that is even more difficult to predict correctly:

Question: Who scored the first points for Denver?

Context: Denver took the opening kickoff and started out strong with Peyton Manning completing an 18-yard pass to tight end Owen Daniels and a 22-yard throw to receiver Andre Caldwell. A pair of carries by C. J. Anderson moved the ball up 20 yards to the Panthers 14-yard line, but Carolina's defense dug in over the next three plays. First, linebacker Shaq Thompson tackled Ronnie Hillman for a 3-yard loss. Then after an incompletion, Thomas Davis tackled Anderson for a 1-yard gain on third down, forcing Denver to settle for a 3–0 lead on a Brandon McManus 34-yard field goal. The score marked the first time in the entire postseason that Carolina was facing a deficit.

Answer: Brandon McManus

Prediction: Peyton Manning

While the MAPT model can easily identify names in response to a "Why?" question, it fails to recognize that Brandon McManus scored the first points for Denver due to a lack of semantic understanding. Since there is no variation of the word "points" within the context, the model must understand not only that a field goal results in points scored, but that the field goal kicked by McManus is the first play to result in points scored. If there existed a sufficient amount of football-related domain data, then it is likely that the model would learn these semantic meanings during the DAPT stage. However, because there is an insufficient amount of out-of-domain data related to football (or even other sports), MAPT is unable to help the model generalize to these examples.

7 Conclusion

Ultimately, the key discovery of this project is that MAPT (and the DAPT stage of MAPT in particular) causes the model's performance to suffer relative to the baseline primarily due to the limited number of unique examples that exist in the out-of-domain training set. Additionally, the experiments above have shown that EDA techniques such as word substitution are unlikely to help models adapt to entirely new domains without sufficient overlap with the original domain. Whether this claim can extend to other augmentation techniques like backtranslation is unknown, but this is a possible area of future research.

While there are some positives to the results, there are quite a few limitations that the experiments had in hindsight. Firstly, it would have been useful to treat each of the separate out-of-domain datasets as their own separate domain, training three separate models in the process. While the training sets would have been split even further, it may be possible for DistilBERT to learn representations for a single isolated dataset, even if it cannot do so for the union of all three out-of-domain datasets. Additionally, while the focus of the experiments was on pretraining the models further, there were little to no changes made on the finetuning aspect of the project. For example, despite it being outside the scope of MAPT, it would have been useful to analyze how data augmentation during QA training time influences the performance of the model compared to augmentation during the DAPT stage. While the final results were less successful than desired in achieving the goal of increased out-of-domain performance, highlighting the shortcomings of DAPT and TAPT in low-resource settings will hopefully guide other NLP research focused on building robust models for downstream tasks in the future.

References

[1] 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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[2] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems, 2017.

[3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[4] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[5] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019.

[6] Xiaochuang Han and Jacob Eisenstein. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China, November 2019. Association for Computational Linguistics.

[7] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification?, 2020.

[8] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[9] Roee Aharoni and Yoav Goldberg. Unsupervised domain clusters in pretrained language models, 2020.

[10] Juri Ganitkevitch and Chris Callison-Burch. The multilingual paraphrase database. In *The 9th edition of the Language Resources and Evaluation Conference*, Reykjavik, Iceland, May 2014. European Language Resources Association.

[11] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July 2015. Association for Computational Linguistics.

[12] Edward Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019.

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

[14] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset, 2017.

[15] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, March 2019.

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

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

[18] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics.