Towards a Robust Question Answering System through Domain-adaptive Pretraining and Data Augmentation

Stanford CS224N Default Project

Feng Chen
Department of Applied Physics
Stanford University
fengc@stanford.edu

Abstract

Large language models (LMs) have shown great success over a bunch of tasks in the past few years. These large LMs are trained on enormous corpus, and it now becomes a question whether they are robust to domain shift. We find in this paper that the domain of question answering (QA) problems has significant impact on the performance of these fine-tuned LMs and these fine-tuned QA models are still sensitive to domain shift during test time. This potentially causes problems in many real-word applications where broad or evolving domains are involved. In this paper, we offer two potential solutions towards building a more robust QA system with domain shift. First, we propose to continue pretraining the LMs on the objective domains. We find that domain-adaptive pretraining helps improve out-of-domain test performance. In some cases, we might have additional small amount of training data on the test domain. We propose to use data augmentation tricks to maximally utilize these data for domain adaptation purpose. We find that data augmentation tricks, including synonym replacement, random insertion and random deletion, can further improve the performance on out-of-domain test samples. Our work shows that the improvements in performance from domain-adaptive pretraining and data augmentation are additive. With both methods applied, our model achieves a test performance of 60.731 in F1 score and 42.248 in EM score. The experiments and methods discussed in this paper will contribute to a deeper understanding of LMs and efforts towards building a more robust QA system.

1 Key Information to include

- Mentor: Elissa Li

2 Introduction

Today in the natural language processing (NLP) community, large pretrained language models (LMs) are usually used as base model to be fine-tuned on the objective tasks. These large LMs such as RoBERTa [1] and GPT-3 [2] have shown great success over a bunch of tasks. On question answering tasks specifically, fine-tuned ALBERT model, for example, achieves an average F1 score of 91.286 [3], surpassing human performance [4]. Because these large LMs are trained on enormous corpus, it now becomes a question whether they are robust to domain shift. Recent study suggests that those LMs are still not robust to out-of-domain test samples [5]. This might cause problems in many real-world applications where broad or evolving domains are involved. We confirm in this paper that fine-tuned LMs on the task of question answering (QA) are still sensitive to domain shift during test time. NLP problems are hard for computer because the meaning of the language can depend on the domain. In different contexts, the same word or similar structures can have different meanings.

It is important to research the robustness of the LMs and find ways to adapt pretrained models to specific domain or task. To alleviate the robustness problem, previous work suggests using a second phase of pretraining [5] and data augmentation tricks [6]. However, these methods have not been validated on LMs for the specific task of QA. As is pointed out in [6], it is still unknown whether data augmentation tricks could yield the same substantial improvements when using pretrained models. More specifically, does it still help to adapt these large LMs to a specific domain? This is particularly important if we want to further improve the performance as well as the robustness for real-word applications. From a practical perspective, robustness is critical to many application since test and training data are rarely independent and identically distributed. In this paper, we systematically study the methods of domain-adaptive pretraining and data augmentation. We confirm that both methods are still helpful to the robustness of LMs on the problem of QA. More importantly, we demonstrate that these two methods are additive. Applying both methods leads to the state-of-the-art results.

3 Related Work

Model robustness Model robustness remains a key challenge in many neural network NLP models. Previous work has shown that these models are vulnerable to adversarial attack (e.g., [7]). Some work also shows that these models are sensitive to domain shift during test time (e.g., [5]). These problems hinder the applications since test data distribution is rarely the same as training data distribution. Thus, It is important to research the robustness of the NLP models and find ways to enhance the robustness. In this paper, we will examine two specific ways to build a more robust question answering system.

Data augmentation One way to build a robust model is through data augmentation. Data augmentation has already shown great success in a bunch of fields such as computer vision (e.g., [8]). In NLP problems, a recent paper [6] proposes four easy ways to augment training data that lead to an improvement on a bunch of models and tasks. Specifically, it proposes to slightly modify the words in the context (by insertion, deletion, swapping or replacement) and assumes that these small modifications won't change the general meaning of the context thus preserving the label information. Although data augmentation methods in NLP have been explored for a long time (e.g., [9]), there are concerns whether these methods will still be useful with large pretrained LMs [6]. These data augmentation tricks are yet to be tested on pretrained LMs on QA problem.

Domain and task adaptation The usage of language depends on the domain. Humans are very good at domain adaptation on NLP tasks, but generalization of NLP models across domains is still an open question. One way to tackle the domain shift problem is through adaptation. Recent work [5] suggests that adaption to the test domain or the task helps increase generalizability during test time. The authors proposed to use LM tasks to adapt the model to objective domain or task. Because the LM tasks are similar to those used in pretraining, the paper refers to the methods as domain-adaptive pretraining and task-adaptive pretraining. Meantime, there have been a few similar works looking into the domain-adaptive pretraining (e.g., [9]) and task-adaptive pretraining (e.g., [10]). Although these works show that adaptive pretraining boosts robustness, the adaptation method hasn't been studied systematically on the problem of QA. It also remains unknown whether adaptive pretraining is additive to other methods such as data augmentation.

4 Approach

We formalize the problem of robust QA system as follows: given training data $\mathcal{D}_{Train}$ in domain $\mathcal{C}_{Train}$, we aim to learn a function $\mathcal{F}$ to answer the questions in domain $\mathcal{C}_{Test}$. We here use DistilBERT [11] as our base model. The baseline system fine-tunes DistilBERT on $\mathcal{D}_{Train}$ and directly uses the fine-tuned DistilBERT for predictions in test domain. Although the task is the same during training and testing, the data distributions, or more specifically, the domains of the data are different. Therefore, the model generally achieves lower performance on test domain than on training domain (see Table 2 and 3). And even the training and test domains stay the same, performance for the model on QA problem in different domains is different. In this paper, we mainly investigate two approaches to enhance the robustness of the baseline model: domain adaptive pretraining (DAPT) [5] and a variant of data augmentation techniques originally proposed in [6].

DAPT assumes large unlabeled corpus ($\mathcal{D}_{DAPT}$) available in the test domain, which by assumption should have similar distribution as the objective domain. In DAPT, we hope to adapt the model to the

test domain by continuously pretraining the large pretrained LM on additional domain-specific data. The continuously pretraining procedure follows the original training settings for BERT [12]. The unsupervised learning is composed of two tasks: the masked LM and next sentence prediction. In masked LM task, 15% of the input tokens are randomly selected to be replaced. Of these selected tokens, 80% will be replaced by [MASK] token, 10% will stay unchanged and 10% will be replaced by random vocabulary tokens. The objective is to predict the next word based on the context seen so far. For next sentence prediction task, the objective is to predict whether two sentences follow each other in the original text, which helps learning the long-term dependency between sentences and should help on downstream tasks such as QA. Codes for pretraining are adapted from the Transformer package[1]. After this second phase of domain-specific pretraining, the model is fine-tuned on QA problem with training data $\mathcal{D}_{Train}$. Codes for fine-tuning are adapted from the provided starting codes[2].

Data augmentation tricks assume additional small amount of training data in the test domain available. Through augmentation, we aim to most efficiently utilize the additional small amount of the training data and train the model to fit the test domain better. Different from DAPT, data augmentation tricks still use the label information. We here explore variants of three data augmentation tricks originally proposed in [6]: synonym replacement, random insertion and random deletion. In the original methods, a random set of words (α percent of the total words) in the paragraph will be chosen. In synonym replacement and random deletion, the selected words will either be replaced by synonyms or be deleted. In random insertion, the selected words will be used to choose synonymous words which will then be randomly inserted into the paragraph. We modify these three methods to fit the QA problem. Since we are still using the labels, we don't want to change, insert into or delete the words corresponding to the truth answer. So, phrases that are referred to as final answers will be excluded from these modifications. In other words, the answer should be consistent among these augmented paragraphs. Still, we need to take into account the change in the answer indices since insertion and deletion would change the index of the answer in the paragraph. An example of the data augmentation tricks used in this paper is shown in Table 1. We implemented these variants of data augmentation tricks with reference to [6]. We use WordNet [13] to find synonyms.

Question: The cause of death of Don Knotts is what?
Answer: Lung cancer

Method	**Paragraph**	**Answer start**
None	Don Knotts died at the age of 81 on February 24, 2006, at the Cedars-Sinai Medical Center in Los Angeles, California from pulmonary and respiratory complications to pneumonia related to lung cancer.	31
Synonym Replacement	Don Knotts died at the age of **eighty one** on February 24, 2006, at the Cedars-Sinai Medical Center in Los Angeles, California from pulmonary and respiratory **knottiness** to pneumonia **refer** to lung cancer.	32
Random Insertion	Don Knotts died at **along** the age **Sinai desert** of 81 on February 24, 2006, at the Cedars-Sinai Medical Center in Los Angeles, California from pulmonary and respiratory **link up** complications to pneumonia related to lung cancer.	36
Random Deletion	Don Knotts died at the age of 81 on February 24, 2006, at the Cedars-Sinai in Los Angeles, California from pulmonary and respiratory complications to pneumonia related to lung cancer.	29

Table 1: Example of data augmentation tricks used in the paper. Punctuation is replaced with space in preprocessing, so the word Cedars-Sinai is counted as two words. This example is taken from the out-of-domain validation set.

[1] `https://huggingface.co/transformers/`
[2] `https://github.com/MurtyShikhar/robustqa.git`

5 Experiments

5.1 Data

The in-domain training data ($\mathcal{D}^{ind}_{Train}$) for QA in the experiments are composed of three datasets: SQuAD [14], NewsQA [15] and Natural Question s[16]. We use 50,000 samples from each dataset for training and the rest for in-domain validation. A small number of additional training samples ($\mathcal{D}^{ood}_{Train}$) are provided for all the out-of-domain datasets. The out-of-domain validation data are composed of three datasets: DuoRC [17], RACE [18] and RelationExtraction [19]. Details about data statistics can be found in the project handout [20]. DuoRC uses passages from movie reviews, RACE uses passages and questions from English examinations, and RelationExtraction uses data from Wikipedia. Therefore, we choose the domain-specific datasets ($\mathcal{D}_{DAPT}$) as IMDB [21], DuoRC, RACE, SQuAD and RelationExtraction. The label information is never used in DAPT. For the method of data augmentation, we only augment the small amount of the out-of-domain training samples $\mathcal{D}^{ood}_{Train}$. Here, we denote the augmented out-of-domain training dataset as $\mathcal{D}^{ood}_{Aug}$. Details about datasets used in different model configurations can be found in A.2.

5.2 Evaluation method

We use Exact Match (EM) and F1 score to evaluate the performance. EM strictly measures whether the output matches the ground truth answer or not. On the other hand, F1 score is less strict, which is defined as the harmonic mean of precision and recall. We evaluate the metrics on each dataset as well as the overall in- and out-of- domain data.

5.3 Experimental details

We here systematically test the methods of DAPT and data augmentation. We investigate three model configurations besides the baseline DistilBERT model: the DistilBERT model with DAPT, the DistilBERT model with data augmentation, and the DistilBERT model with both DAPT and data augmentation. All the experiments are run in parallel on four NVIDIA Tesla K80/K100 GPUs. For DAPT, we continue pretraining on domain-specific corpus $\mathcal{D}_{DAPT}$ for a total number 190,000 steps. 1% of the entire dataset is held out for validation purpose. For the first 127,000 steps, we use a linearly decaying learning rate starting from 5e-5 with a batch size of 32. Then, we use a linearly decaying learning rate starting from 2e-5 for another 63,000 steps with a batch size of 64. The perplexity of randomly held-out documents drops from 12.04 to 4.76 (Fig. 1).

In data augmentation, we randomly generate 18 new paragraphs for each paragraph in $\mathcal{D}^{ood}_{Train}$ (6 new paragraphs from each method: synonym replacement, random deletion and random insertion). We then randomly choose 16 paragraphs and preserve the original labels. The augmented data is 16 times larger than the original dataset $\mathcal{D}^{ood}_{Train}$. In modification process, We choose the hyperparameter $\alpha = 0.1$, which corresponds to about 10% of the words being modified. We add these augmented out-of-domain training data $\mathcal{D}^{ood}_{Aug}$ to the original in- and out-of-domain training data, and fine-tune the DistilBERT model on the union of original and augmented data. We find that training on both in- and out-of- domain datasets simultaneously performs slightly better than training on them separately. When utilizing both DAPT and data augmentation, we first apply DAPT for a second-phase of pretraining and then fine-tune the pretrained model on the labeled training data together with the augmented data.

The fine-tuning setups are chosen to be the same for all model configurations. We run a small range of hyperparameter search and decide to use learning rate of 3e-5 and a batch size of 16. We continuously evaluate the performance every 5,000 steps and the model with the best performance on overall out-of-domain validation set is saved.

5.4 Results

In Table 2 and 3, we show validation results on both in- and out-of- domain data. We find that DistilBERT with both DAPT and data augmentation achieves best performance on the out-of-domain validation data in terms of the overall F1 and EM score. This same model configuration achieves an F1 score of 60.731 and an EM score of 42.428 on the test set.

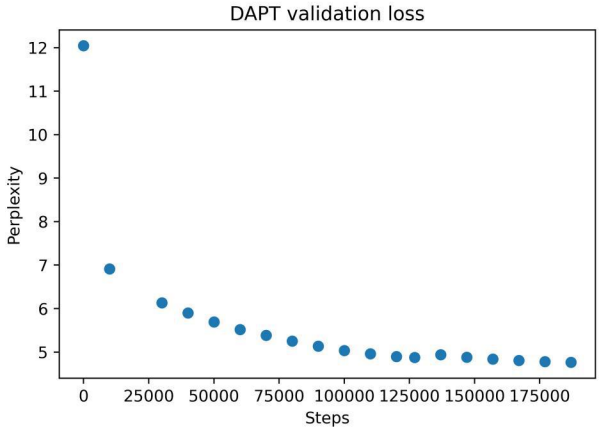

Figure 1: DAPT Training Curve.

The model configurations with only DAPT or data augmentation can serve as ablation tests. From Table 3, we confirm that both DAPT and data augmentation help to enhance the robustness of the model that is fine-tuned from large pretrained LM. If only applying one method, data augmentation results in more improvements in the overall out-of-domain performance. We also find that the enhancement in performance from both methods are additive, though the overall improvement is smaller than the sum of the improvement from each method (ΔF1(both)< ΔF1(DAPT)+ΔF1(Data Aug.), ΔEM(both)< ΔEM(DAPT)+ΔEM(Data Aug.).

Model	SQuAD	NewsQA	Natural Questions	Overall
DistilBERT	F1: 77.52 EM: 62.72	F1: **58.54** EM: **40.55**	F1: **69.15** EM: **52.35**	F1: **71.50** EM: **55.10**
DistilBERT + DAPT	F1: 76.11 EM: 62.79	F1: 57.62 EM: 39.74	F1: 68.44 EM: 52.05	F1: 70.48 (-1.02) EM: 54.86 (-0.24)
DistilBERT + Data Aug.	F1: 77.44 EM: **62.92**	F1: 56.76 EM: 38.53	F1: 67.99 EM: 50.89	F1: 70.64 (-0.86) EM: 54.17 (-0.93)
DistilBERT + DAPT + Data Aug.	F1: **77.76** EM: 62.87	F1: 58.41 EM: 39.93	F1: 68.75 EM: 51.78	F1: 71.38 (-0.12) EM: 54.79 (-0.31)

Table 2: In-domain validation performance.

Model	DuoRC	RACE	RelationExtraction	Overall
DistilBERT	F1: 44.24 EM: **34.92**	F1: 33.40 EM: 18.75	F1: 68.80 EM: 45.31	F1: 58.37 EM: 39.79
DistilBERT + DAPT	F1: 43.73 EM: 31.75	F1: 34.11 EM: **20.31**	F1: 72.42 EM: 50.78	F1: 60.52 (+2.15) EM: 42.40 (+2.61)
DistilBERT + Data Aug.	F1: 43.76 EM: 31.75	F1: **35.25** EM: **20.31**	F1: 73.67 EM: **53.12**	F1: 61.42 (+3.05) EM: 43.85 (+4.06)
DistilBERT + DAPT + Data Aug.	F1: **44.83** EM: **34.92**	F1: 34.57 EM: 19.53	F1: **74.57** EM: 52.34	F1: **62.21 (+3.84)** EM: **44.20 (+4.41)**

Table 3: Out-of-domain validation performance. The overall performance is the weighted sum of the performance of the three out-of-domain datasets. The weights are set in accordance with the test distribution in the project handout [20].

6 Analysis

In terms of the effects of DAPT and data augmentation on in-domain validation performance, we find that models with DAPT and/or data augmentation slightly hurt the overall in-domain performance. This is not surprising. DAPT adapts the model to the test domain, which is different from the training domain. On the other hand, the additional augmented out-of-domain training samples modify the training data distribution away from the in-domain validation data distribution. Therefore, both methods enhance the robustness of the model at the cost of in-domain performance. Strikingly, when we combine both methods, the in-domain performance doesn't worsen but instead becomes better. One possible explanation is that with both methods applied, the model learns more features related to QA that are invariant under domain shift. The model might learn from augmented training data to use its knowledge from domain-adaptive pretraining to solve QA tasks instead of barely learning superficial correlations specific to the domain when only one method is applied. This interesting finding suggests that it is possible to increase robustness without sacrificing the performance. Actually, as shown in Table 3, the overall in-domain validation performance remains almost unchanged when both DAPT and data augmentation are applied. Also, we see that the performance on SQuAD dataset even slightly improved with both methods. This might be a trivial result since we include SQuAD as the pretraining dataset in DAPT, but it also suggests that by including in-domain corpus into the DAPT training data, the adverse effect can be further diminished and even turned into improvement.

Interestingly, we find that the improvement of out-of-domain validation performance depends on the domain. We find that the performance on RelationExtraction dataset improves most, while the performance on DuoRC dataset almost does not change. When using only one method, the F1 and EM scores even drop on DuoRC dataset. These three out-of-domain datasets are different in many ways but one remarkable difference is the length of the paragraph. Based on provided out-of-domain training data, we calculate the mean paragraph length for each dataset. On average, DuoRC dataset has a context length of 621 ± 79 words, RACE has a context length of 297 ± 79 words, and RelationExtraction has a context length of f 25 ± 11 words. The paragraph length is negatively correlated with the improvement in the performance. One possible explanation for the observed difference is that longer paragraph has richer information more complicated context and thus more sensitive to domain shift. And because of the complex semantic information, QA datasets with longer paragraph tend to benefit less from DAPT and data augmentation methods. We leave it for future work to study how dataset properties affect model robustness and out-of-domain performance.

7 Conclusion

We find that even using large pretrained LM such as DistilBERT, the model is still sensitive to domain shift during test time after being fine-tuned on QA problem. We propose to use DAPT and data augmentation to enhance model robustness and we confirm that both methods are effective and compatible with pretrained LM. These two methods are based on different assumptions. DAPT is suitable when additional domain-specific corpus is available, while data augmentation requires additional training examples in the objective domain. Importantly, we find that the improvement from both methods are additive and applying both methods leads to the best performance on test data with an F1 score of 42.248 and an EM score of 60.731. Although the enhancement in out-of-domain test performance is at the cost of in-domain performance when DAPT or data augmentation is applied, we find that the cost is diminished when using both methods. One limitation of the proposed methods is that the improvement in robustness depends on the domain. We speculate that QA problems with longer paragraph length will benefit less from these adaptation methods. In the future, it might be interesting to look into how both training and test domain properties affect the performance and robustness.

References

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

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[3] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

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

[5] 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 preprint arXiv:2004.10964*, 2020.

[6] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[7] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*, 2019.

[8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[9] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[10] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[11] 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.

[12] 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.

[13] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] `http://web.stanford.edu/class/cs224n/project/project-proposal-instructions-2021.pdf`.

[21] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

A Model configuration and training details

A.1 Dataset size

Dataset	Size
$\mathcal{D}^{ind}_{Train}$	150,000
D^{ood}_{Train}	381
$\mathcal{D}^{ood}_{Aug}$	6,096
$\mathcal{D}^{ood}_{Aug}$	1,053,769

Table 4: Dataset size

A.2 Training dataset

Configuration	Training Data
DistilBERT	$\mathcal{D}^{ind}_{Train}$
DistilBERT + DAPT	$\mathcal{D}^{ind}_{Train} \cup \mathcal{D}^{ood}_{Train} \cup \mathcal{D}_{DAPT}$
DistilBERT + Data Aug.	$\mathcal{D}^{ind}_{Train} \cup \mathcal{D}^{ood}_{Train} \cup \mathcal{D}^{ood}_{Aug}$
DistilBERT + DAPT + Data Aug.	$\mathcal{D}^{ind}_{Train} \cup \mathcal{D}^{ood}_{Train} \cup \mathcal{D}_{DAPT} \cup \mathcal{D}^{ood}_{Aug}$

Table 5: Training datasets for different model configurations