Robust Question Answering on Out-of-Domain Data

Stanford CS224N Default Project

Ranchao Yang
Institute of Computational & Mathematical Engineering
Stanford University
audreyrc@stanford.edu

Abstract

Generalizing question answering (QA) models to new domains is a challenging task, as models tend to learn superficial correlations from the training data and require additional fine-tuning to adapt to new domains. This paper attempts to investigate several proposed methods to improve the robustness of question answering model on out-of-domain data. We first introduce and discuss approaches including domain adversarial training, task-adaptive fine-tuning, and data augmentation. Next, we experiment with these methods, applying them to our model individually and collectively. We conclude that the combination of the three methods leads to significant improvement in model performance on out-of-domain data.

1 Key Information to include

- Mentor: Elaine Sui

2 Introduction

Question answering (QA) is a critical natural language processing (NLP) problem. QA systems allow people to ask a question in natural language and get an immediate and brief response. They are widely used in many applications such as search engines [1] and chatbots [2]. With the introduction of powerful neural network models such as BERT [3], excellent performance can be achieved for a wide range of NLP tasks including question answering.

Good performance often highly depends on the size of training data, and it is achieved with the assumption that train data and test data come from the same distribution. Previous work has shown that models tend to learn superficial correlations that fail to generalize beyond the training distribution [4]. However, real-world applications generally do not satisfy the "independently and identically distributed" assumption. For example, in the field of question answering, train data and test data often come from distinct user interactions.

A lot of research has been conducted to solve this problem, and people have experimented with various methods to improve the robustness of models so that models generalize well to new domains: for example, adversarial training [5], task-adaptive fine-tuning [6], data augmentation [7][8], mix-of-expert systems[9], and meta-learning[10].

Our study aims to improve the robustness of our baseline question answering model, which simply fine-tunes on the in-domain datasets, so that it can generalize to out-of-domain data. We experiment with several published methods including adversarial training, task-adaptive fine-tuning, and data augmentation. We incorporate these methods into our training and fine-tuning processes, investigating how they individually and collectively affect our model robustness.

3 Related Work

3.1 BERT and DistilBERT

The Bidirectional Encoder Representations from Transformers (BERT) model pre-trains deep bidirectional representations on a large corpus through masked language modeling and next sentence prediction [3]. It can then be fine-tuned with an additional output layer to create models for a wide range of specific NLP tasks, such as text classification and question answering. Without complicated adaptations to the model architecture, BERT is easy to use but can achieve powerful results for various tasks. After being developed and published by Google in 2019, it has become a widely used baseline in NLP experiments [11].

DistilBERT was later proposed as a smaller, faster, and cheaper version of BERT, but it is able to achieve a similar performance [12]. Unlike prior work, it features the use of distillation during the pre-training step, therefore achieving 97% of understanding capabilities and being 60% faster and 40% smaller.

3.2 Adversarial Training

Adversarial training was originally proposed in 2014 in the field of computer vision, specifically for image generation [13]. Then this concept was further applied to NLP tasks such as text classification [14] [15] and relation extraction [16]. It has also been used to train language-invariant features [17]. In a recent study [5], adversarial training framework is used to train a question answering model with domain-agnostic representation. The model consists of two parts: a QA model and a discriminator. During the training process, the QA model trains domain-invariant features that can hide domain label from the discriminator, and at the same time, the discriminator attempts to identify the correct domain label. As a result, it is shown in the study that QA model learns features that are generalizable to other domains, and the F1 score is improved by over 2 points with adversarial training than without adversarial training.

3.3 Task-Adaptive Fine-Tuning

Task-adaptive fine-tuning refers to further pre-training (masked language modeling and next sentence prediction) on an unlabeled task-specific dataset. Previous studies have shown that this method can help models understand task-specific language better, thus significantly improving the performance on tasks such as text classification [18] [6].

A recent study [19] even shows that for domains with abundant unlabeled text, such as biomedicine, pre-training language models only on specific data results in substantial gains over pre-training on both general and specific data.

3.4 Data Augmentation

Data augmentation has been commonly used in speech [20] and computer vision [21] and has improved model robustness in those areas, especially when the dataset is too small. Since the weak performance of QA models on out-of-domain data may be due to the fact that they tend to learn the superficial correlations in the in-domain data, data augmentation may bring noise to the data and help models ignore these superficial correlations and instead learn domain-invariant features. Previous work has proposed multiple data augmentation techniques in NLP. One strategy is back-translation [7]: translating the original paragraph into a target language and then translating it back into the original language to produce a paraphrase. Paper [8] also proposes multiple other methods that bring noise to text data, including synonym replacement, random deletion, random swap, and random insertion.

4 Approach

4.1 Baseline Model

Our baseline model is based on DistilBERT model, which has 6-layer, 768-hidden, 12-heads, 66M parameters. It pretrained on the same data as BERT, which is BookCorpus, consisting of 11,038 unpublished books and English Wikipedia [12], distilled from the BERT model bert-base-uncased checkpoint. Our baseline model is then constructed by fine-tuning DistilBERT on the three in-domain train datasets (SQuAD, NewsQA, Natural Questions).

4.2 Methods

In this study, we apply published methods to improve the robustness of the baseline, including domain adversarial training, task-adaptive fine-tuning, and data augmentation.

4.2.1 Domain Adversarial Training

Adversarial training aims to learn domain-invariant features rather than ones that are specific to certain domains. Our approach is adapted from paper [5], which introduces a model with domain-agnostic representation for question answering task. As discussed in the last section, the model includes two parts: a QA model and a domain discriminator. During training, the QA model tries to minimize negative log-likelihood of predictions for start and end positions, as for a standard QA task, but it also tries to deceive the discriminator. Meanwhile, the discriminator tries to predict the domain label correctly. So the loss for the QA model is a combination of the standard QA loss and the Kullback-Leibler (KL) divergence between uniform distribution over K classes denoted as U and the discriminator's prediction denoted as P.

$$\mathcal{L} = \mathcal{L}_{QA} + \lambda \cdot \mathcal{L}_{adv} = \mathcal{L}_{QA} + \lambda \cdot KL(U||P)$$

We adapt the code provided by [5] to incorporate adversarial training into our project.

4.2.2 Task-Adaptive Fine-Tuning

Task-adaptive fine-tuning refers to further pre-training on the unlabeled datasets for a specific task, and studies have shown that it can significantly improve the performance on text classification tasks [18][6]. We first use the original train and validation datasets to generate masked language modeling (MLM) datasets for pretraining, following the masked data generation code from [6]. We pre-train MLM on the datasets and then further train this model on QA data to build a QA model.

4.2.3 Data Augmentation

Two sets of data augmentation are applied to the out-of-domain datasets because there are only a limited number of out-of-domain samples.

First, the questions in the train datasets are augmented via back translation. We use Googletrans, which is a published python library that implemented Google Translate API, to translate each question in the three out-of-domain train datasets into french and then back into English.

Second, the questions and the context paragraphs are augmented using methods including word substitution, random swap, and random deletion, following the idea in [8]. We apply these methods to each question in the out-of-domain train datasets. For random words in a sentence, we use NLTK python library to pick a synonym and replace the word with the synonym. We also attempt to swap random words or remove random words from each sentence in order to increase noise. Also, for each context paragraph, we apply the above methods only to the part that does not contain the answer so as to keep the ground-truth answer unchanged.

5 Experiments

5.1 Data

As shown in the Table 1, we have three in-domain reading comprehension datasets and three out-of-domain datasets, and the numbers correspond to the sizes of the datasets. As the project aims to improve the robustness of a QA system, we have a very large number of in-domain data and a small number of out-of-domain data. For each question in the train and validation datasets, there are three human-provided ground truth answers, and our evaluation is based on all three answers.

Dataset	Question Source	Passage Source	Train	Dev	Test
		in-domain datasets			
SQuAD	Crowdsourced	Wikipedia	50000	10,507	-
NewsQA	Crowdsourced	News articles	50000	4,212	-
Natural Questions	Search logs	Wikipedia	50000	12,836	-
		out-of-domain datasets			
DuoRC	Crowdsourced	Movie reviews	127	126	1248
RACE	Teachers	Examinations	127	128	419
RelationExtraction	Synthetic	Wikipedia	127	128	2693

Table 1: Dataset Statistics. Table borrowed from [22].

5.2 Evaluation method

We use two metrics to evaluate the performance: Exact Match (EM) score and F1 score. During evaluation, for each question, we calculate EM and F1 scores for each of the three ground-truth answers and take the maximum of the three.

- Exact match (EM) is a binary measure that evaluates whether the model output matches the ground truth answer exactly.
- F1 is the harmonic mean of the precision and recall and is less strict than EM score.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

5.3 Experimental details

Our baseline model simply fine-tunes DistilBERT on in-domain train data. For fine-tuning on in-domain train data, we used the default hyperparameters suggested in the project guideline: batch size of 16, learning rate of 3e-5, and a total of 3 epochs. We used AdamW as our optimizer. Model was evaluated on the validation dataset every 5000 steps.

Next, we incorporated a domain adversarial training (DAT) framework with both a QA model and a discriminator. During training, we used three layers with a dropout layer in each layer and a drop-out rate of 0.1. We set the weight λ for adversarial loss to be 0.5. We used default hyperparameters for the QA model.

We also attempted to incorporate both task-adaptive fine-tuning (TAFT) and domain adversarial training (DAT) into our model. We built a `DistilBertForMaskedLM` with default hyperparameters to perform pre-training, with batch size of 16, learning rate of 3e-5 and a total of 3 epochs. Then the best pre-training MLM model was used for downstream adversarial training framework, as discussed above.

Next, we applied data augmentation and further fine-tuned our model on augmented out-of-domain train data. For fine-tuning on out-of-domain data, we used the same hyperparameters as above. Since we had a relatively smaller out-of-domain dataset even after performing data augmentation, we evaluated the model on validation dataset every 50 steps. We experimented with (1) the combination of TAFT and fine-tuning on augmented out-of-domain data and (2) the combination of all three methods.

Table 2 shows a list of models that we have experimented with and their performance on out-of-domain validation datasets. It took around 3 hours to train the baseline model, and with adversarial training, it took a little longer than the baseline. Task-adaptive pre-training took extra time, so the last four models took around 4-5 hours.

Method	EM	F1
Baseline	30.63	47.72
+DAT -TAFT -finetune	32.20	47.19
-DAT +TAFT -finetune	30.89	48.24
+DAT +TAFT -finetune	31.68	48.33
-DAT +TAFT +finetune	34.82	50.9
+DAT +TAFT +finetune	35.60	51.75

Table 2: Results on validation datasets. EM and F1 are evaluation metrics for our models. +/-DAT represents with/without domain adversarial training on in-domain train data; +/-TAFT represents with/without task-adaptvie fine-tuning on in-domain data; +/-finetune refers to with/without fine-tuning on augmented out-of-domain data.

Question	What work of fiction is Jack Harkness located in?
Context paragraph	Captain Jack Harkness is a fictional character played by John Barrowman in Doctor Who and its spin-off series, Torchwood.
Answer	Torchwood
Prediction	Doctor Who
Question	Due to which disease did Julius Garfinckel die?
Context paragraph	Julius Garfinckel died on his 64th birthday of pneumonia in Washington, D.C. His funeral was held two days later at All Souls Unitarian Church.
Answer	pneumonia
Prediction	Julius Garfinckel died on his 64th birthday of pneumonia
Question	On what instrument is Hungarian Rhapsodies played?
Context paragraph	The Hungarian Rhapsodies, S.244, R.106 (..., Hungarian: Magyar rapszódiák), is a set of 19 piano pieces based on Hungarian folk themes, ...
Answer	piano
Prediction	"Magyar rapszódiák), is a set of 19 piano"

Table 3: Examples of incorrect predictions.

5.4 Results

Table 2 shows a list of models that we have trained and their evaluation scores on out-of-domain validation datasets. Surprisingly, some techniques fail to improve the baseline model, which has an EM score of 30.63 and an F1 score of 47.72. For example, with only domain adversarial training, the EM score increases to 32.20 but F1 score drops to 47.19.

We also observe that further fine-tuning on out-of-domain data contributes the most to model improvement. With both domain adversarial training and task-adaptive fine-tuning on in-domain data, the EM score has a 1-point improvement and the F1 score has a half-point improvement. On the other hand, with task-adaptive fine-tuning and out-of-domain fine-tuning, the model achieves an EM score of 34.82 and an F1 score of 50.9 on out-of-domain validation data. We think this is probably due to the fact that, through further fine-tuning, the model manages to learn more language correlations in the new domain and becomes capable of understanding the language of new domains better.

After we incorporate all three methods into our model, our model achieves the best performance on out-of-domain validation data: an EM score of 35.60 and an F1 score of 51.75. We then evaluate this model and the second best one using the out-of-domain test datas. On the out-of-domain test datasets, the baseline model achieves an EM score of 40.16 and an F1 score of 59.31. The second best model achieves an EM score of 40.55 and an F1 score of 59.62, which shows limited improvement. The best model shows greater improvement as expected, reaching an EM score of 41.28 and an F1 score of 60.11.

6 Analysis

In this section, we conduct error analysis and further analyze our results. Table 3 provides a list of examples of the incorrect predictions that our model produces during evaluation.

In the first example, the actual answer is a work of fiction "Torchwood" that appears in the context, but the predicted answer is another work of fiction "Doctor Who" which also appears in the context. One possible reason our model fails is that it is able to identify the correct entity type, but since multiple things with the same entity type occur in the context, the model fails to distinguish which one is correct.

In each of the next two examples, the actual answer is a single word, but the predicted answer is a long word span that contains the single word. One possibility is that our model is able to find an approximate place of the answer but cannot select the exact word that answers the question.

7 Conclusion

In this study, we investigate methods to improve the robustness of a question answering model, including domain adversarial training, task-adaptive fine-tuning, and data augmentation. We implement these methods and evaluate their contribution to model improvement using EM score and F1 score as evaluation metrics. We find that our baseline model is most greatly improved when we first perform task-adaptive fine-tuning and domain adversarial training on in-domain data and then further fine-tune on augmented out-of-domain data.

One limitation of this project is that, since time was limited, there was not enough time to tune the hyper-parameters for each model. Hence the results we discussed in previous sections might be sub-optimal. Another limitation arises from the finding that applying domain adversarial training individually barely improved the model robustness, which is unexpected due to its ability to train domain-invariant features.

In the future, we would like to analyze the domain adversarial training framework and optimize it for question answering model. We would also like to optimize the methods we have experimented with and explore other methods that would be helpful for model robustness.

References

[1] Aniket D. Kadam, Shashank Joshi, Sachin V. Shinde, and Sampat 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.

[2] Devanshi Singh, K.Rebecca Suraksha, and S.Jaya Nirmala. Question answering chatbot using deep learning with nlp. In *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6, 2021.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

[4] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *ArXiv*, abs/1707.07328, 2017.

[5] Seanie Lee, Donggyu Kim, and Jangwon Park. Domain-agnostic question-answering with adversarial training, 2019.

[6] 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, 2020.

[7] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *ArXiv*, abs/1511.06709, 2016.

[8] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. A survey of data augmentation approaches for nlp. *ArXiv*, abs/2105.03075, 2021.

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

[10] Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. Investigating meta-learning algorithms for low-resource natural language understanding tasks. *ArXiv*, abs/1908.10423, 2019.

[11] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

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

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

[14] Xilun Chen, Yu 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.

[15] Xilun Chen and Claire Cardie. Multinomial adversarial networks for multi-domain text classification. In *NAACL*, 2018.

[16] Yanhua Yu, Kanghao He, and Jie Li. Adversarial training for supervised relation extraction. *Tsinghua Science and Technology*, 27(3):610–618, 2022.

[17] Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial training for unsupervised bilingual lexicon induction. In *ACL*, 2017.

[18] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[19] Yuxian Gu, Robert Tinn, Hao Cheng, Michael R. Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3:1 – 23, 2022.

[20] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(9):1469–1477, 2015.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

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