Robust Cross-domain Question Answering via Adversarial Training and Document Embedding Clustering

Stanford CS224N Default Project Robust QA track

Anson Wang
Department of Electrical Engineering
Stanford University
sicongw@stanford.edu

Abstract

State-of-the-art question answering models often perform well on in-domain questions, but fail to generalize to out-of-domain questions. In this project, we leverage domain adversarial training [LKP19] to encourage the model to learn domain-agnostic representations, thereby improving generalization performance. We further re-examine the definition of a domain, where we explore generating domain labels using document embeddings and k-means clustering rather than simply equating domains with dataset names. We show that the embedding-based labeling approach results in improvement over the dataset-based labeling approach.

1 Key Information to include

- Mentor: Fenglu Hong
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

Over the past few years, pre-trained transformer models such as BERT have delivered exciting improvements in many natural language understanding (NLU) tasks. These models are pre-trained on massive corpora with the Masked Language Modeling (MLM) and next sentence prediction objectives, then fine-tuned on a downstream task such as question answering with a task-specific dataset [Dev+18]. However, despite these models achieving state-of-the-art F1 scores on various NLU tasks, it remains to be seen whether the models truly *understand* language and semantics, rather than simply generating answers based on some memorized distributions. In fact, [JL17] demonstrates that many state-of-the-art models are not robust against adversarial examples designed to test semantic understanding. This suggests the models may have only learned brittle, superficial correlations. Another manifestation of the lack of robustness is poor cross-domain performance. Models tend to overfit to the training data distribution, and generalize poorly when presented with test data drawn from a different distribution.

In this project, we specifically examine cross-domain robustness in the task of question answering (QA). One way to improve generalization performance, or equivalently to prevent overfitting, is to leverage adversarial training as proposed by [LKP19]. The basic idea is to introduce a discriminator model that uses hidden features generated by the QA model to classify which domain a training example originates from, while simultaneously encouraging the QA model to fool the discriminator as much as possible. Under this adversarial regularization, the QA model tends to generate domain-agnostic hidden representations which prevent overfitting to the training distribution, thus help improve performance on out-of-domain data distributions.

To prepare for adversarial training, each data example needs to have a domain label. [LKP19] adopts a dataset-based approach, where the domain label is determined by the dataset the training example belongs to. For example, if 3 datasets are used for training, then examples belonging to the first dataset will have label 1, examples belonging to the second dataset will have label 2, so on and so forth. This labeling approach is simple, but may not be effective. Instead, a more natural approach is to use linguistic features extracted from the training examples to group these examples into domains. Specifically, [SHW21] proposes using a normalized vector consisted of the Term Frequency-Inverse Document Frequency (TF-IDF) scores of the 300 most common words as the linguistic feature, and then groups these features into K domains via K-means clustering. In this project, we propose an improvement where instead of TF-IDF scores, we extract the *document embeddings* from the training examples and cluster them into K domains. We show that our approach improves both EM and F1 scores on the out-of-domain validation set when compared to the TF-IDF and dataset-based approaches.

3 Related Work

Cross-domain Generalization. Various methods for improving domain generalization exist. In [Jac+91], Jacobs et al. proposes the idea of Mixtures of Experts (MoE), where the training dataset is partitioned into several subsets (domains), and for each subset an expert network is trained. A gating network is simultaneously trained to mix the outputs from individual expert networks to form the final output. When trained properly, MoE is an effective multitask model that could generalize well to out-of-domain data distributions.

In [DYA19], Dou et al. proposes using Meta-Learning to improve few-shot fine-tuning performance on language understanding tasks. The idea of meta-learning is to partition the training dataset into separate domains, and search for "middle ground" model parameters such that when fine-tuned with a few examples drawn from a specific domain, the model can converge with good performance on that particular domain. In the context of robust QA, the model can be meta-trained on all in-domain datasets, and then fine-tuned on the limited out-of-domain training set.

In [LKP19], Lee et al. proposes using domain adversarial training to improve QA domain generalization. A discriminator network is introduced to play an adversarial game against the QA model on classifying domain labels based on hidden features generated by the QA model. This forces the QA model to generate domain-invariant features which improve cross-domain generalization. However, Lee et al. define "domain" in a perhaps overly simplistic way. They treat the dataset source, such as SQuAD or RACE, as a domain. Since a single dataset could contain multiple topics, or multiple datasets could contain instances of the same topic, treating dataset as a domain may hinder and confuse the discriminator, making the adversarial training procedure less effective. [SHW21] addresses this problem by generating domain labels based on clustering TF-IDF features of context paragraphs in the training dataset. We think [SHW21] shows a promising direction, and that better document embedding techniques can be used instead of TF-IDF which is akin to bag-of-words methods that are not very effective at distilling semantic meanings.

Document Embedding. The goal of document embedding is to project documents into a high-dimensional vector space, such that the semantic similarity between documents can be compared using vector metrics such as cosine similarity or Euclidean distance. A variety of methods have been proposed for generating document embeddings. For example, one simple method converts each word of a document into GloVe word embedding, and then average all word embeddings to represent the document embedding. Another common method feeds sentences into BERT and takes the average of all output hidden vectors to represent the document embedding. [RG19] takes the BERT-based approach to the next level, where they fine-tune BERT so that it outputs semantically meaningful hidden vectors that support cosine-similarity comparison. The fine-tuned model is called Sentence-BERT (SBERT), which achieved state-of-the-art performance on semantic textual similarity tasks. Since domain clustering is essentially an unsupervised semantic similarity comparison task, we think SBERT is a suitable choice for the job.

4 Approach

4.1 Domain Label Generation

We explore two ways to generate domain labels.

Dataset-based labeling. Suppose our training data are taken from K dataset sources, such as SQuAD, DuoRC, etc. Then each dataset source is considered as a domain. That is, examples originating from dataset source i are labeled with domain i, where $i \in \{1, \ldots, K\}$. This is the labeling method used by [LKP19].

Embedding-based labeling. In the QA task, training data consist of paragraph-question pairs. For each training example, we take the context paragraph and feed it into a pre-trained Sentence-BERT [RG19] to produce a document embedding vector. Once we have the embeddings from the entire training set, we perform K-means clustering on them to extract K domains. Examples belonging to cluster i are labeled with domain i, where $i \in \{1, \ldots, K\}$. Here K is a hyper-parameter that needs to be pre-selected. In terms of implementation, we use the `sentence-transformers` library developed by [RG19] for SBERT, and `scikit-learn` for k-means clustering.

4.2 Adversarial Training

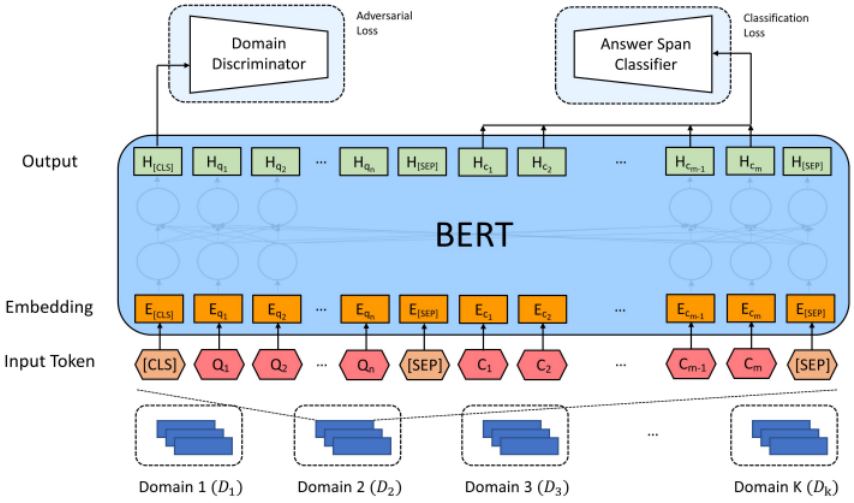

Figure 1: Domain adversarial architecture from [LKP19]

We adopt a similar model architecture as [LKP19], consisting of a QA network and a discriminator network. We use the Huggingface [Wol+20] `DistilBertForQuestionAnswering` as our QA model. The hidden state output corresponding to the `[CLS]` token, h_{cls}, is fed into the discriminator that tries to predict which domain the training example belongs to. The QA model's loss function is modified to encourage the model to produce a domain-agnostic h_{cls} that maximally confuses the discriminator.

Specifically, a new adversarial loss term $\mathcal{L}_{adv}$ is introduced:

$$\mathcal{L}_{adv} = \frac{1}{N} \sum_{i=1}^{N} KL(\mathcal{U}_K \mid\mid P_{dis}(l \mid h_{cls}^{(i)}))$$

where N is the batch size, K is the number of domains, $\mathcal{U}_K$ is a K-dimensional uniform distribution, $P_{dis}(l \mid h_{cls}^{(i)})$ is the domain distribution predicted by the discriminator. Note that $\mathcal{L}_{adv}$ is minimized when the distribution $P_{dis}(l \mid h_{cls}^{(i)})$ is identical to $\mathcal{U}_K$.

Let the QA answer loss be $\mathcal{L}_{QA}$, which is based on the correctness of the predicted answer span. Then, the combined Domain-adversarial QA loss is defined as:

$$\mathcal{L}_{DQA} = \mathcal{L}_{QA} + \lambda_{adv}\mathcal{L}_{adv}$$

where λ_{adv} is a hyper-parameter that controls the importance of the adversarial term.

As for the discriminator, its loss is defined as

$$\mathcal{L}_{dis} = -\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N_k}\log P_{dis}(l = k \mid h_{cls}^{(k,i)})$$

where N_k is the number of examples within a minibatch that belongs to domain k, and $P_{dis}(l = k \mid h_{cls}^{(k,i)})$ is the predicted probability that example (k, i) belongs to domain category k.

For each training step, we perform gradient descent on both $\mathcal{L}_{DQA}$ and $\mathcal{L}_{dis}$.

4.3 Baseline

The Huggingface `DistilBertForQuestionAnswering` model is used as a baseline. This is a pre-trained DistilBERT model with a single dense layer QA top that predicts answer span. The baseline model is trained on in-domain data as well as a tiny amount of out-of-domain data.

5 Experiments

5.1 Data

We use SQuAD [Raj+16], NewsQA [Tri+16], Natural Questions [Kwi+19] as in-domain datasets, and DuoRC [Sah+18], RACE [Lai+17], Relation Extraction [Lev+17] as out-of-domain datasets. These datasets are splitted into train, validation, and test subsets. The split is detailed in Table 1.

Category	Dataset	Train	Validation	Test
in-domain	SQuAD	50537	10784	-
	NewsQA	126287	10567	-
	Natural Questions	65480	17537	-
out-of-domain	DuoRC	435	411	4240
	RACE	196	182	633
	Relation Extraction	127	128	2694

Table 1: Sample count in each dataset split. This is a count of the number of *questions*, not context paragraphs.

All of our models, including the baseline, are trained on the *combined* in-domain and out-of-domain training sets. That is, we group `indomain_train` and `oodomain_train` into a single training set. We do not separately fine-tune on `oodomain_train`.

Model performance is evaluated on the in-domain and out-of-domain validation sets separately. The out-of-domain test set has answers held-out, so it is only used for the final leaderboard submission.

5.2 Evaluation method

The primary metric we use for evaluating model performance is the average F1 score across all examples in the validation or test set. Compared to Exact Match (EM), the F1 metric is more lenient and closer to how humans would evaluate answers.

5.3 Experimental details

We train three models: (1) Baseline, (2) Domain adversarial with dataset-based labeling (DA-dataset), (3) Domain adversarial with embedding-based labeling (DA-embedding). All models are trained on

the combined in-domain and out-of-domain training sets. Details of hyper-parameters are listed in Table 2.

Model	**QA_lr**	**QA_weight_decay**	**epochs**	λ_{adv}	**dis_lr**	**num_domains** (K)
Baseline	3e-5	0	3	-	-	-
DA-dataset	3e-5	0	3	1e-2	3e-5	6
DA-embedding	2.08e-5	1.1e-2	3	8e-2	1.04e-5	40

Table 2: Model training hyper-parameters

Hyper-parameters for the baseline model are taken from the RobustQA project handout. For the DA-dataset model, hyper-parameters are taken from [LKP19].

The hyper-parameters for the DA-embedding model are chosen empirically as follows: We first fix $K = 20$ and `num_epochs`=3. Then we use the Weights & Biases library [Bie20] to run a Bayes hyper-parameter search on {QA_lr, QA_weight_decay, λ_{adv}, dis_lr}. The set of hyper-parameters achieving the highest F1 score on `oodomain_val` is selected. Finally, we train with $K = 20, 30, 40, 50$ while fixing all other hyper-parameters, and pick the K with the highest F1 score.

5.4 Results

Our best model, DA-embedding, achieved an **F1 score of 60.372** and an **EM score of 42.477** on the out-of-domain test set (`oodomain_test`). It ranks 3rd on the RobustQA test leaderboard (team named "bulgogi") at the time of this writing.

Performance of the baseline, DA-dataset, and DA-embedding models on in-domain and out-of-domain validation sets are reported in Table 3 and Table 4 respectively.

Model	**EM**	**F1**
Baseline	54.81	70.92
DA-dataset	54.50	70.68
DA-embedding	54.74	70.84

Table 3: In-domain validation set validation results

Model	**EM**	**F1**	**% over baseline F1**
Baseline	36.13	51.19	-
DA-dataset	34.03	50.13	-2.1%
DA-embedding	**37.43**	**52.52**	+2.6%

Table 4: Out-of-domain validation results

As shown in Table 4, our DA-embedding model generalizes better on out-of-domain datasets, outperforming both the DA-dataset and baseline models. In addition, our embedding-based labeling approach shows a 3.6% improvement over [SHW21] which uses a TF-IDF based approach for labeling domains ([SHW21] achieved an F1 score of 50.706).

However, aside from the satisfactory performance of our model, we are actually surprised by the overall results on two aspects. Firstly, the DA-dataset model, which is an almost exact reproduction of [LKP19] other than swapping BERT for DistilBERT, performs significantly worse than the baseline. We tried different hyper-parameters as well as various forms of learning rate annealing, but still could not reproduce the authors' claims. Potential reason for underperformance will be discussed in Section 6.1. Secondly, the baseline is surprisingly strong due to the fact that we mixed in `oodoamin_train` in the training data. This will be analyzed in more details in Section 6.3.

6 Analysis

6.1 Underperformance of the Dataset-based Domain Labeling Approach

Despite numerous effort, we were unable to make the DA-dataset model outperform the baseline. We believe the bottleneck lies in the dataset-based domain labeling which generates overly broad and semantically-inconsistent labels. These labels mislead the adversarial training process, causing the model to learn hidden features that deteriorate overall language understanding performance. This hypothesis is supported by the DA-dataset model's weak in-domain performance as well as the performance increase once we switch to the embedding-based labeling approach.

6.2 Domain Clustering

We use Sentence-BERT to convert context paragraphs into document embeddings, and then cluster these embeddings into K semantic domains. To qualitatively evaluate whether this labeling approach is effective, we randomly select a few paragraphs from a domain and see if they are semantically relevant. Some examples are shown in Table 5 and 6. We can see that paragraphs within a certain cluster discuss similar topics.

Index	Excerpt
1	Several molecular mechanisms of **antibacterial resistance** exist. Intrinsic antibacterial resistance may be part of the genetic makeup of bacterial strains ...
2	**Antibacterial antibiotics** are commonly classified based on their mechanism of action, chemical structure, or spectrum of activity. Most target bacterial functions or growth processes ...
3	Phage therapy is another option that is being looked into for treating resistant strains of **bacteria**. The way that researchers are doing this is by infecting pathogenic bacteria with their own viruses, more specifically, bacteriophages ...

Table 5: Excerpts of random paragraphs from Cluster 9 which appears to be biology-related. Topic keywords are manually bolded.

Index	Excerpt
1	Mendes revealed that production would begin on 8 December 2014 at Pinewood Studios, with **filming** taking seven months. Mendes also confirmed several filming locations, including London, Mexico City and Rome ...
2	...This was also the first season without **executive producer** Nigel Lythgoe who left to focus on the international versions of his **show** So You Think You Can Dance ...
3	Season six began on Tuesday, January 16, 2007. The **premiere** drew a massive audience of 37.3 million viewers, peaking in the last half hour with more than 41 million viewers ...

Table 6: Excerpts of random paragraphs from Cluster 16 which appears to be TV and movie related. Topic keywords are manually bolded.

6.3 Surprising Effect of Mixing in Small Amount of Out-of-domain Training Data

Not necessarily related to adversarial training, but one surprising result we notice is that including a tiny amount of out-of-domain training examples (0.31% of total examples) significantly boosts generalization performance. Specifically, we experiment with training the Baseline, DA-dataset, DA-embedding models on in-domain-only train set (`indomain_train`), and compare their performance

against models trained on a mix of in- and out-of-domain datasets (Section 5.4). Model performance on the out-of-domain validation set (`oodomain_val`) is shown in Table 7.

Training data: 100% in-domain		
Model	**EM**	**F1**
Baseline	33.25	48.14
DA-dataset	32.98	48.64
DA-embedding	33.25	49.38
Training data: 99.69% in-domain + 0.31% out-of-domain		
Model	**EM**	**F1**
Baseline	36.13 (+8.7%)	51.19 (+6.3%)
DA-dataset	34.03 (+3.2%)	50.13 (+3.1%)
DA-embedding	37.43 (**+12.6%**)	52.52 (+6.4%)

Table 7: Performance comparison between in-domain training vs mixing in small amount of out-of-domain data. Percentage in parenthesis denotes relative improvement compared to in-domain only training.

Given the tiny number of out-of-domain training examples, one would not expect them to make a big difference. But quite the opposite – a mere 0.31% mix-in of out-of-domain data boosts both F1 and EM scores on all 3 models by as much as 12.6%. That is a huge gain in generalization performance for such a small, simple change!

For comparison, we also experimented with training the baseline model on `indomain_train` only, and then fine-tune on `oodomain_train`. The fine-tuning only resulted in a minor increase in performance: EM score 33.25 $\rightarrow$ 33.77 (+1.6%), F1 score 48.14 $\rightarrow$ 49.59 (3%). The mix-in approach is clearly superior over the fine-tuning approach.

Due to limited time with this project and the fact that we only noticed this result towards the end, we are unable to pursue a deeper analysis of this phenomenon. Before making guesses why the mix-in approach works so well, we think it is critical to conduct more experiments to see whether this phenomenon generalizes to other tasks and datasets, or whether it is a sporadic one-off coincidence associated with our particular dataset distribution. This could be an investigative topic for future work.

7 Conclusion

In this project, we explored the use of Domain Adversarial training to improve robustness of a DistilBERT based question answering system. We demonstrated the importance of domain label generation, where labels based on semantic similarity result in better model performance than labels generated based on dataset source. We explored using Sentence-BERT to convert training paragraphs into document embeddings, followed by k-means clustering to generate K labels. Using these semantically-generated labels, our model was able to outperform the DistilBERT baseline.

Towards the end of our project, we also discovered a surprising effect, where mixing in tiny amount of out-of-domain data into training can significantly boost model generalization performance. We conducted experiments to verify that this phenomenon holds for all 3 models we trained in this project. The mix-in approach is extremely simple yet seems to be very effective, and we feel this could be a venue for future CS224N projects.

References

[Bie20] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: `https://www.wandb.com/`.

[Dev+18] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[DYA19] Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. "Investigating Meta-Learning Algorithms for Low-Resource Natural Language Understanding Tasks". In: *CoRR* abs/1908.10423 (2019). arXiv: 1908.10423. URL: http://arxiv.org/abs/1908.10423.

[Jac+91] Robert A. Jacobs et al. "Adaptive Mixtures of Local Experts". In: *Neural Computation* 3.1 (1991), pp. 79–87. DOI: 10.1162/neco.1991.3.1.79.

[JL17] Robin Jia and Percy Liang. "Adversarial Examples for Evaluating Reading Comprehension Systems". In: *CoRR* abs/1707.07328 (2017). arXiv: 1707.07328. URL: http://arxiv.org/abs/1707.07328.

[Kwi+19] Tom Kwiatkowski et al. "Natural Questions: a Benchmark for Question Answering Research". In: *Transactions of the Association of Computational Linguistics* (2019).

[Lai+17] Guokun Lai et al. "RACE: Large-scale ReAding Comprehension Dataset From Examinations". In: *CoRR* abs/1704.04683 (2017). arXiv: 1704.04683. URL: http://arxiv.org/abs/1704.04683.

[Lev+17] Omer Levy et al. "Zero-Shot Relation Extraction via Reading Comprehension". In: *CoRR* abs/1706.04115 (2017). arXiv: 1706.04115. URL: http://arxiv.org/abs/1706.04115.

[LKP19] Seanie Lee, Donggyu Kim, and Jangwon Park. "Domain-agnostic Question-Answering with Adversarial Training". In: *CoRR* abs/1910.09342 (2019). arXiv: 1910.09342. URL: http://arxiv.org/abs/1910.09342.

[Raj+16] Pranav Rajpurkar et al. "SQuAD: 100, 000+ Questions for Machine Comprehension of Text". In: *CoRR* abs/1606.05250 (2016). arXiv: 1606.05250. URL: http://arxiv.org/abs/1606.05250.

[RG19] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *CoRR* abs/1908.10084 (2019). arXiv: 1908.10084. URL: http://arxiv.org/abs/1908.10084.

[Sah+18] Amrita Saha et al. "DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension". In: *CoRR* abs/1804.07927 (2018). arXiv: 1804.07927. URL: http://arxiv.org/abs/1804.07927.

[SHW21] Danny Schwartz, Brynne Hurst, and Grace Wang. "Domain Adversarial Training for QA Systems". In: *CS224N Final Project* (2021). URL: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/reports/final_reports/report117.pdf.

[Tri+16] Adam Trischler et al. "NewsQA: A Machine Comprehension Dataset". In: *CoRR* abs/1611.09830 (2016). arXiv: 1611.09830. URL: http://arxiv.org/abs/1611.09830.

[Wol+20] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.