Robust Question Answering using Domain Adversarial Training

Stanford CS224N Default Project

Bryan Zhu
Department of Computer Science
Stanford University
bwzhu@stanford.edu

Abstract

While recent developments in deep learning and natural language understanding have produced models that perform very well on question answering tasks, they often learn superficial correlations specific to their training data and fail to generalize to unseen domains. We aim to create a more robust, generalized model by forcing it to create domain-invariant representations of the input using an adversarial discriminator system that attempts to classify the outputs of the QA model by domain. Our results show improvements over the baseline on average, although the model performed worse on certain datasets. We hypothesize that this is caused by differences in the kind of reasoning required for those datasets, differences which actually end up being erased by the discriminator.

1 Key Information to include

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

2 Introduction

Much progress has been made in the past few years in building large language models to perform natural language understanding tasks, and in some areas, including reading comprehension question answering (QA), these models can reach or surpass human levels of accuracy. However, it is still unclear whether those models truly have a good understanding of human language, and many examples show that they often learn brittle connections that fail to generalize beyond the scope of their training data. For example, McCoy, Pavlick, and Linzen [1] show that these models can learn simple heuristics about language that work in common cases but don't capture the underlying meaning of sentences and fail in more complex cases, and Jia and Liang [2] show that just by adversarially adding an irrelevant sentence to question-answering passages we can reduce the accuracy of a QA model by over half. Along these lines, we demonstrate that a QA model trained on given datasets of questions has much reduced performance on outside datasets, and we attempt to build a robust model which can perform more effectively on these out-of-domain questions. Such a model would certainly be more useful in the real world, where the questions which are being asked of it will rarely always be of the same kind that it has been trained on.

The approach that we take is using adversarial training. Originating in the field of image processing with generative adversarial networks (GANs), the main idea of adversarial training is to simultaneously train two neural networks with opposing goals with the hope that, when faced with an adversarial opponent, they will be able to optimize in a way to overcome that challenge. In domain adversarial training, one model is a language model which is trained on inputs from various domains.

The other is a discriminator which takes in representations output by the first model and attempts to classify which domain they came from. The discriminator's task is to classify these representations as accurately as possible, while the first model's task is, in addition to solving its original language problem, to make it as hard as possible for the discriminator to classify its representations. In doing so, we hope that the model will produce representations which are void of domain-specific features that will generalize better to unseen data.

We find that while our question answering model has mixed success on confusing its discriminator, it does manage to perform better on out-of-domain data on average, scoring about 2 points higher on both EM and F1 metrics on an out-of-domain validation set compared to a baseline that does not use adversarial training.

3 Related Work

Ganin and Lempitsky [3] in 2015 introduce the idea of domain adaptation through adversarial classification for general backpropagation neural networks, and they demonstrate its effectiveness on the MNIST digit classification dataset and other problems involving reading text from images. Sato et al. [4] bring this methodology to the field of natural language understanding, using adversarial training to build a model that performs dependency parsing and scores 3 to 6 points higher on UAS and LAS scores compared to their baseline. Lee, Kim, and Park [5] also tackle the reading comprehension QA problem using a domain classifying discriminator with a pretrained BERT model, and their results show scores of 1.5 to 2 points higher on EM and F1 metrics. Our work builds on theirs by reproducing their results on a different set of data and by experimenting with the parameters of the discriminator.

There has also been much work outside of adversarial training on improving domain generalization. One approach is a mixture-of-experts technique [6], where a separate model is trained for each in-domain dataset along with an adaptive gating function. When presented with out-of-domain data, the gating function chooses some combination of the outputs of each of the "expert" models, allowing each model to focus on learning its own dataset and delegating the adaptation to the gating function. More recently, an approach that has become popular is meta-learning [7], where the model is pre-trained on a set of multiple tasks in a way such that it can learn new tasks with much fewer examples. Li et al. [8] use ideas from meta-learning to develop a method of training any model to be more robust to domain changes, rather than just creating a specific model. However, Lee et al. tested this method on transformer QA models and found that they were both slower to train and did not result in increased performance, so we did not follow this path.

4 Approach

Our baseline QA model is a pretrained DistilBERT model [9] which is finetuned on the question answering task using data from three different domains (described in detail in Section 5). This model was provided to us as part of the starter code for the Default Project. The discriminator model is a 3-layer MLP which takes as input $\mathbf{h}$, the QA model's representation of its input question and answer, and outputs a probability distribution specifying which one of the domains it believes its input came from. In our case, $\mathbf{h}$ is the representation of the `[CLS]` token at the last layer of the QA model. This discriminator model follows ideas from [5] but was completely implemented myself. The structure of the full model is presented in Figure 1.

The main modifications to the baseline that were made are changes to the loss functions. Since the QA model wants to successfully answer questions but also confuse the discriminator, we add a term $\mathcal{L}_D$ to the loss of the original DistilBERT model $\mathcal{L}_{QA}$ for a total loss of

$$\mathcal{L} = \mathcal{L}_{QA} + \lambda \mathcal{L}_D$$

where λ is a hyperparameter to be tuned. The goal is for the discriminator to be completely unable to tell which domain its input came from; i.e. for the discriminator to output the uniform distribution over its classes, so $\mathcal{L}_D$ is calculated as the Kullback-Leibler (KL) divergence between the discriminator's prediction and the uniform distribution.

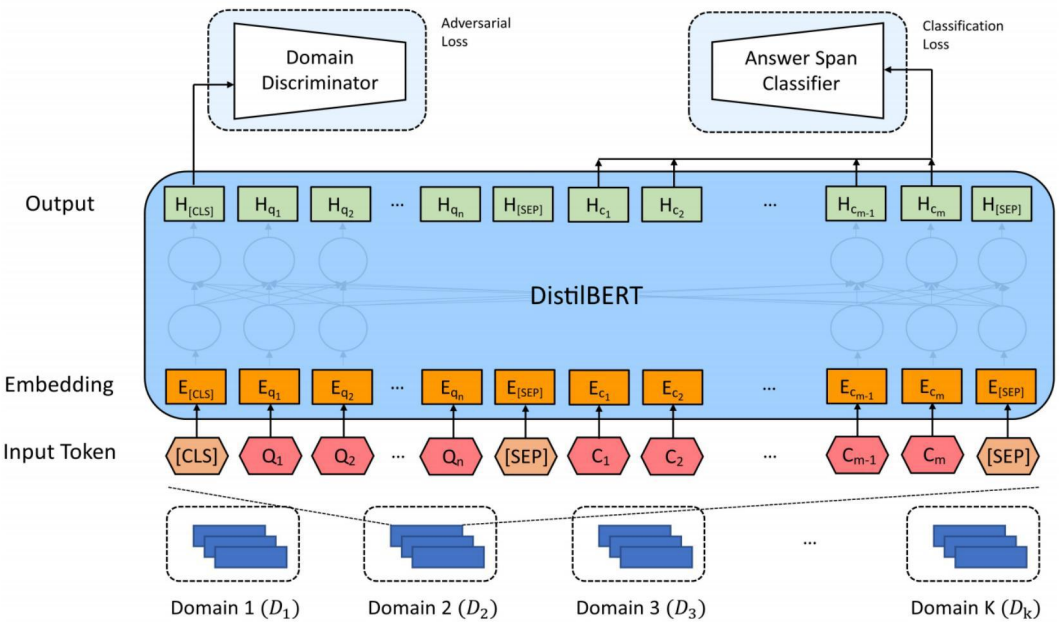

Figure 1: Structure of the QA model and discriminator, taken from [5] with modifications

On the other hand, the discriminator's goal is to predict the correct domain for its input, so its loss function is the cross-entropy loss between its output and the true domain ℓ:

$$\mathcal{L} = -\log \mathbf{p}_{disc}(\ell)$$

For each step of training, we alternate between training the QA model on an input and then the discriminator on the model's embeddings of that input. In both models, loss functions are averaged across the batch and optimized using AdamW [10] with a learning rate of 3×10^{-5}.

5 Experiments

5.1 Data and evaluation

The model was trained on questions from three datasets of different domains: SQuAD [11], Natural Questions [12], and NewsQA [13]. These will be referred to as the IID datasets. Each dataset in the training data contained 50000 questions along with the passages containing the answers to those questions, and the model was tasked to perform reading comprehension on that data. It was then evaluated on questions from three separate domain datasets: DuoRC [14], RACE [15], and RelationExtraction [16], referred to as the OoD datasets. The model was scored on EM (Exact Match) and F1 metrics.

5.2 Improving the baseline

For our first experiment, we simply added the discriminator to the QA model while leaving everything else the same. The hyperparameters relating to the discriminator follow those used in [5]: the model itself is a 3-layer MLP with a hidden layer size of 768, and we use $\lambda = 0.01$ for the weighting parameter in the QA loss.

The second experiment we performed added small samples (127 questions each) from the three out-of-domain datasets to the training data. Though the small number of samples were probably not enough for the model to learn too much directly, the discriminator was forced to differentiate between six domains instead of three, and likewise the QA model had to create representations that made the discriminator predict all 6 classes as equally likely. We hoped that in doing so, the QA model could

learn something about all the classes even when processing training samples in the IID datasets. The results are shown in Table 1:

Model	Train EM	Train F1	Eval EM	Eval F1
Baseline	54.77	70.51	31.68	47.10
Adversarial IID	54.42	70.58	32.72	49.39
Adversarial Combined	54.15	70.10	33.51	49.16

Table 1: Model performance with and without the discriminator

The model achieved comparable performance on the in-domain datasets and managed to outperform the baseline on the out-of-domain datasets on both EM and F1 metrics, both when including and excluding samples of OoD data. The model that included OoD data performed slightly better in EM score and the same in F1.

5.3 Tuning loss weighting

In the Adversarial IID case without OoD data, the discriminator accuracy remained high throughout most of the training process, so our next experiment was testing how increasing λ would affect model performance. Recall that λ determines how the QA model weights its own loss and the loss arising from the discriminator, so increasing λ tells the model to put more importance on confusing the discriminator. We tested values of $\lambda = 0.05$ and $\lambda = 0.1$, and the model performance and discriminator performance are shown in Table 2 and Figure 2 respectively. All of our experiments attained similar performance on the in-domain datasets, so we omit the train scores from here on out.

Lambda	Eval EM	Eval F1
(original) 0.01	32.72	49.39
0.05	31.15	47.25
0.1	31.15	46.19

Table 2: Model performance for different values of λ

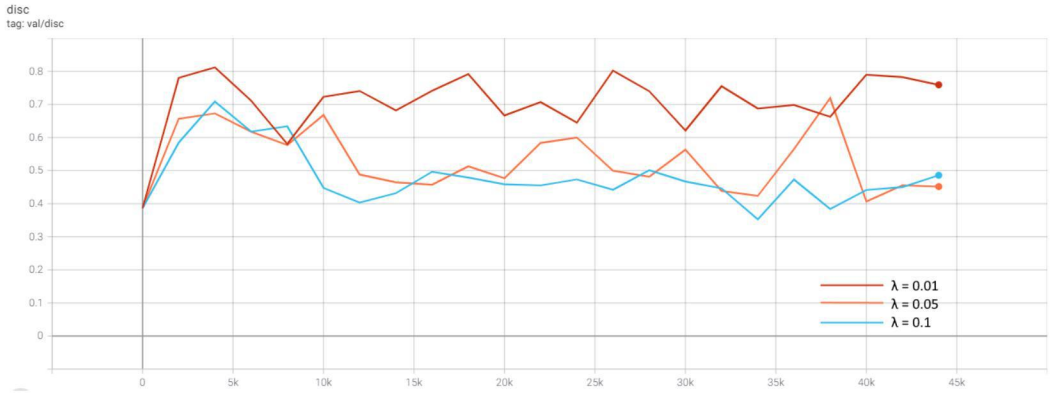

Figure 2: Discriminator accuracy during training for different values of λ

The results seem to suggest that while increasing lambda was successful in producing representations that were better at confusing the discriminator, they were also worse at performing the original task of question answering.

5.4 Tuning the discriminator size

In the Adversarial IID + OoD case, the discriminator started off strong but gradually became less successful as training continued, suggesting that it had more trouble when asked to perform classification over six domains as opposed to three. We decided to experiment with increasing the

Hidden size	Eval EM	Eval F1
(original) 768	33.51	49.16
1536	33.51	49.21

Table 3: Model performance for different discriminator hidden layer sizes

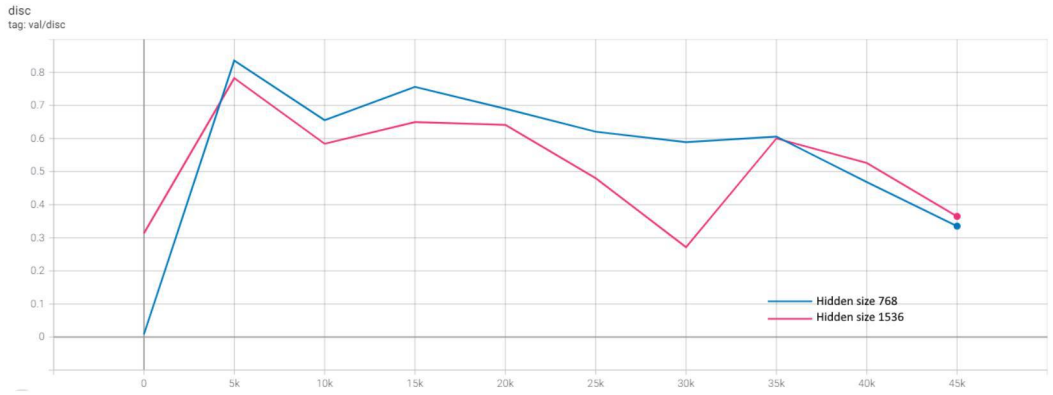

Figure 3: Discriminator accuracy during training for different hidden layer sizes

hidden size of the discriminator when using all six training datasets, hopefully giving it more power. The results are shown in Table 3 and Figure 3.

There seemed to be no significant difference in model performance or discriminator performance when increasing the hidden layer size, suggesting that this was not a particularly important factor.

5.5 Individual dataset performance

Finally, we investigated the performance of some of our original models from Section 5.2 on the six individual datasets. The results are shown in Table 4.

	SQuAD	Nat. Questions	NewsQA	DuoRC	RACE	RelationExtraction
Baseline EM	63.33	52.80	39.27	33.33	23.44	38.28
Baseline F1	77.01	69.43	57.51	40.31	36.76	64.12
Adv-IID EM	62.17	52.55	40.72	33.33	22.66	42.19
Adv-IID F1	76.51	69.50	58.97	43.41	37.32	67.34
Adv-Comb EM	62.47	51.94	39.53	34.92	13.28	52.34
Adv-Comb F1	77.32	69.02	58.05	43.12	29.00	75.25

Table 4: Model performance on individual datasets

The adversarial model trained on just IID train data performed neutral or slightly better on all of the OoD datasets, while interestingly, the model trained with the combined train data performed significantly worse on RACE but significantly better on RelationExtraction. The large degradation on RACE is not something we expected, and we will discuss this further in section 6. However, because the Adv-Combined model still had the best average EM and F1 score, and because the distribution of questions in the test set favored questions from RelationExtraction, we decided to submit that one to the test leaderboard.

Our final test submission is: **EM = 42.706, F1 = 60.202**.

6 Analysis

6.1 Dataset analysis

The key question to discuss with regard to our results is why our final model performs so well on RelationExtraction but loses performance so sharply on RACE. To do this, we took a deep dive into

the kinds of passages and questions that showed up in our training and evaluation datasets. What we found was that in our IID datasets, the vast majority of questions focused on extracting details within the passage: the formats of the passages varied but the answer could be found just by looking at one or two lines of the passage. The model's job would be to dig through the passage and correctly identify which lines to extract. However, many of the questions from RACE required a higher-level understanding of the passage, often requiring us to consider the passage as a whole, summarize parts of the passage, or compare or count up various different parts of the passage. These RACE questions require a slightly different kind of thinking than simple detail extraction.

We hypothesize that this is why all of our models exhibit reduced performance on RACE, but especially the discriminator model which had samples of RACE data during training time. When presented with these different kinds of questions, it is unclear what the Adv-IID model should try to do, and indeed it doesn't get great performance. However, the Adv-Combined model had already seen these kinds of questions during training, and more importantly needed to convert them into representations which were as indistinguishable as possible from the usual detail extraction questions seen in other datasets. The issue is that these representations may not have been the best way to approach this kind of problem. When faced with further RACE questions, it would also produce these less effective representations.

We can see this occurring in various concrete instances. For example, given a passage about a boy's trip to the ocean, the question asks "Where did the story take place?" The IID model answers the correct answer of "by the sea." However, the Adv-Combined model focuses in on one part of the story and answers "beach." Other examples are shown in Appendix A.

On the other hand, we believe that the addition of the discriminator does help the model avoid learning specific features of the different kinds of passages particular to each dataset, such as the way news articles are generally structured for NewsQA. This helps it attain a higher success rate on DuoRC and especially RelationExtraction, whose questions are generally detail extraction problems but from shorter and differently structured passages.

6.2 Loss weighting analysis

We also found it interesting that the model that performed best on question answering had a discriminator accuracy averaging around 70%, meaning the representations that performed the best on OoD data were not particularly domain-independent on IID data. This could suggest that there are domain-specific features that the network should learn in order to answer questions well, but that these features can transfer over to other datasets. It could also mean that the model simply added noise to confuse the discriminator which did not actually help with question answering in any way, and with higher λ the model just added more noise. It would be interesting to try to discern if either of these is the case, though we are unclear on how to do so.

7 Conclusion

In this project, we implemented and evaluated an adversarial training system for robust reading comprehension question answering on multiple domains. This system improves upon the baseline DistilBERT model on both EM and F1 scores, and achieves a final test set score of EM = 42.706 and F1 = 60.202 which is 5th place on the leaderboard at the time of this writing. We experimented with different hyperparameters of the adversarial model, including weighting between the original loss and adversarial loss and tuning hidden layer size, to find those that gave the best performance. However, we found that the system was not perfect, and while it improved performance on average, it increased performance sharply in some areas while actually regressing in others.

Given the results which seem to suggest that perfectly domain-independent representations do not perform best, one future avenue of work to look at is partial domain independence. We could experiment with only using a part of the QA model's representation in the discriminator, forcing certain features to be independent while allowing it to learn other domain-specific features, or using multiple models where one has a discriminator and the other does not, and concatenating their outputs to use in question answering.