More Explorations with Adversarial Training in Building Robust QA System

Stanford CS224N Default Project

Xiyu Wang
Department of Computer Science
Stanford University
wxy@stanford.edu

Abstract

In real world Question Answering (QA) applications, a model is usually required to generalize to unseen domains. It was found that an Adversarial Training framework where a conventional QA model trained to deceive a domain predicting discriminator can help learn domain-invariant features that generalize better. In this work we explored more discriminator architectures. We showed that by using a single layer Transformer encoder as the discriminator and taking the whole last layer hidden states from the QA model, the system performs better than the originally proposed simple Multilayer Perceptron (MLP) discriminator taking only the hidden state at the [CLS] token of the BERT QA model.

1 Key Information to include

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

2 Introduction

Recent advances in NLP research such as Transformer[1] language model architecture and BERT[2] pre-training technique have pushed the state-of-the-art of various NLP tasks further by a large margin. Question Answering (QA) is one of these tasks and it was shown that models can outperform human on certain QA datasets such as SQuAD[3].

Question Answering or Reading Comprehension is a task where a model is given a paragraph of text and a short question about the paragraph, the model is expected to find the answer within the paragraph. Many practical applications can be abstracted as Question Answering task such as search engine, customer service chatbot etc.

However, it was observed that models outperforming human on a single dataset usually overfits to the dataset and cannot generalize well to datasets from a different domain without additional training [3]. This limitation hinders real world QA applications because there is usually a domain shift between the training data and the serving data.

Many approaches have been proposed to improve model performance on out-of-domain data, such as Mixture of Experts[4] and few-sample fine-tuning[5]. However these approaches either drastically increase the model size or still require out-of-domain training data for fine-tuning. Adversarial Training[6] does not have these shortcomings. Adversarial Training forces the QA model to learn domain-invariant features from training datasets across a few different domains. Specifically a domain discriminator is trained using the hidden states of the QA model to predict the domain of the examples. The QA model is trained to minimize both the QA loss and the adversarial loss that tries to prevent the discriminator from predicting the domains correctly.

Lee et al. [6]'s work that proposed Adversarial Training in building robust QA system only experimented with one discriminator architecture using the last layer hidden state at the [CLS] token. In this paper, we explored using full hidden states of different layers of the QA model and a single layer Transformer encoder as the discriminator. We also explored using different numbers of hidden layers in a Multilayer Perceptron (MLP) as the discriminator on the [CLS] token. We found a discriminator taking the whole last layer hidden states from the QA model performs better than the original proposed Adversarial Training architecture in our experimentation setup. We trained our models using 3 in-domain training datasets (SQuAD [7], NewsQA [8] and Natural Questions [9]). We evaluated our models on the held out test datasets from 3 out-of-domain datasets (DuoRC [10], RACE [11] and RelationExtraction [12]).

3 Related Work

Question Answering There are many types of QA system. Some are knowledge-based using structured data. Some are information retrieval-based using unstructured (text) data. Some find the answers explicitly as a span of words in the text. Some generate the answers word by word using a language model. Some systems like search engines need to first retrieve relevant documents from a large corpus of documents and then try to find answers from the documents.

In this work we only discuss one specific type of QA system - reading comprehension, where the system is given a paragraph of text and a question text at the same time. The answers are explicit spans of consecutive words in the paragraph. The system is expected to understand both the paragraph and what the question asks for and returns the correct span of text. In Machine Learning language, this system can be formulated as a classification system. The system outputs a probability of whether each word is the start of the answer or end of the answer.

BERT[2] first exceeded human performance in reading comprehension in 2018. BERT takes in a pair of paragraph and question as one sequence separated by special characters, i.e. [CLS]question[SEP]paragraph[SEP], and outputs probabilities of start and end tokens. The QA model in this paper finetunes a more cost effective version of BERT called DistilBERT[13].

Adversarial Training was originally proposed in a very popular image generation method known as Generative Adversarial Network (GAN) [14]. In GAN training, there are often two models, a generator that generates images and a discriminator that predicts whether an image is real or generated. The generator and the discriminator are trained simultaneously. While the discriminator gets better at telling apart the generated images from the real images, the generator gets better at generating realistic images that can fool the discriminator.

Lee et al. [6] first proposed using Adversarial Training for building robust QA systems. Their approach uses the hidden state embedding at the [CLS] token of a BERT QA model and a simple MLP as the discriminator. The problem with this approach is although the [CLS] token hidden state can be used to predict the domain, the opposite is not necessarily true. We cannot claim the model is not learning domain specific features just because the [CLS] hidden state is not revealing any. All the other hidden states could still have domain specific features encoded into their embeddings and used in predicting the answer spans.

4 Approach

Adversarial Training can be understood as a regularization technique. We assume the conventional QA loss under-defines what language feature representations to use. In other words, a conventional QA model will treat domain-specific and domain-invariant features as equally good and choose randomly. Adding a discriminator and an adversarial loss regularizes the model to choose domain-invariant features that perform better in out-of-domain data.

The discriminator is loosely coupled with the QA model. It can be added to any QA models as long as it has access to the QA model's hidden states. The discriminator can also be implemented using different structures such as MLP, Transformer layers with a classification head etc. We tried using MLP with one hidden layer and two hidden layers. We also tried using one Transformer layer with one hidden layer MLP classification head on the [CLS] token as shown in Figure 1.

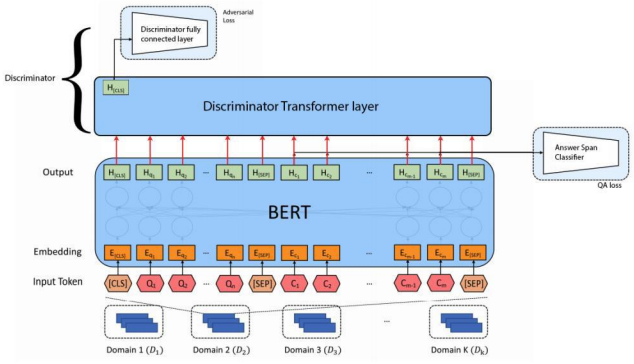

Figure 1: Adversarial training architecture of using one Transformer layer with one hidden layer MLP classification head on the [CLS] token as the discriminator. The discriminator takes the whole last layer hidden states of the QA model as input. Figure adapted from [6].

4.1 Problem Definition

Formally, given K in-domain datasets $D^{(k)}, k = 1...K$, each has $N^{(k)}$ examples. We assume each dataset covers only one domain with label $l^{(k)}$. Extending the conventional QA setup, each training example is a tuple of 4 values: $D_i^{(k)} = \{c_i^{(k)}, q_i^{(k)}, y_i^{(k)}, l_i^{(k)}\}_{i=1}^{N^{(k)}}$, where c represents the paragraphs, q represents the questions, y represents the answer spans the QA model tries to predict, $l_i^{(k)} = l^{(k)}$ for every i represents the domain labels the discriminator tries to predict.

4.2 QA Model

A conventional QA model parameterized by θ is trained to minimize the negative log-likelihood of the answer y_i where $y_{i,s}$ and $y_{i,e}$ are respectively the start position and the end position.

$$\mathcal{L}_{QA} = -\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N_k} \Big[logP_\theta(y_{i,s}^{(k)}|c_i^{(k)}, q_i^{(k)}) + logP_\theta(y_{i,e}^{(k)}|c_i^{(k)}, q_i^{(k)})\Big]$$

4.3 Adversarial Training

A discriminator D parameterized by ϕ is trained to minimize the cross-entropy loss between the predicted class probability and the true domain class label $l_i^{(k)}$.

$$\mathcal{L}_D = -\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N_k} logP_\phi(l_i^{(k)}|\mathbf{h}_i^{(k)})$$

where $\mathbf{h}$ is the hidden state embedding.

An adversarial loss is added in the QA model optimization to maximize the entropy of $P_\phi(l_i|\mathbf{h}_i^{(k)})$. In other words, it tries to fool the discriminator to output uniform distribution over the K domain classes. This is the same as minimizing the Kullback-Leibler (KL) divergence between the uniform distribution denoted as $\mathcal{U}(l)$ and the discriminator's prediction. It is critical to note that we only update the QA model's parameters θ to minimize this loss, not the discriminator's parameters ϕ.

$$\mathcal{L}_{adv} = \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N_k} KL(\mathcal{U}(l)||P_\phi(l_i^{(k)}|\mathbf{h}_i^{(k)}))$$

The final loss for the QA model is

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

where λ is a hyper-parameter for controlling the weight of the adversarial loss. During training, at each step, we first optimize the QA model with the discriminator's parameters ϕ frozen, then optimize the discriminator with the QA model's parameters θ frozen.

5 Experiments

5.1 Data

We trained our models on 3 in-domain training datasets: SQuAD [7], NewsQA [8] and Natural Questions [9]. 50k paragraphs and their associated questions from each dataset are provided by the robust QA track default project. Because Natural Questions dataset has more questions per paragraph and effect of chunking, the number of training examples (i.e. paragraph, question pairs) are not the same. We 2x upsampled SQuAD and NewsQA in our training to make the number roughly equal. This is important for the discriminator to learn the probability of domain classification correctly.

Dataset	Question Source	Passage Source	Passages	Training Examples
SQuAD	Crowdsourced	Wikipedia	50000	50537
NewsQA	Crowdsourced	News articles	50000	65480
Natural Questions	Search logs	Wikipedia	50000	126287

Table 1: The training datasets.

For dev, eval and testing, we used 3 out-of-domain datasets: DuoRC [10], RACE [11] and RelationExtraction [12]. We used F1 score on the dev set in our training loops to determine the best checkpoint. We reported the per dataset eval set and combined test set F1 and EM metrics below.

Dataset	Question Source	Passage Source	Dev	Eval	Test
DuoRC	Crowdsourced	Movie reviews	127	126	1248
RACE	Teachers	Examinations	127	128	419
RelationExtraction	Synthetic	Wikipedia	127	128	2693

Table 2: The dev, eval and test datasets.

5.2 Experimental details

DistilBERT As a baseline, we used a pre-trained DistilBERT model from the Hugging Face "distilbert-base-uncased" model distribution. The model is finetuned using the training dataset.

Original Adv We implemented the Adversarial Training framework replicating [6] on top of the DistilBERT model. The original paper used a two hidden layer MLP as the discriminator. The discriminator used the last layer hidden state at the [CLS] token. The MLP hidden layers have a dimension of 768, the same as the DistilBERT hidden layer dimension. A 0.1 dropout layer is used after each hidden layer. The output layer has 3 nodes used to predict the 3 possible in-domain classes in our datasets. We used AdamW optimizer with the same learning rate 3e-5 for both the DistilBERT model and the discriminator. We used 1e-2 for λ in the loss function.

Simpler Adv This model used an even simpler discriminator than the original paper. It used an one hidden layer MLP. A 0.5 dropout was used to give a stronger regularization. Other hyper parameters are the same as the Original Adv.

Last Layer Adv This model (shown in Figure 1) used the Hugging Face DistilBertForSequenceClassification model as the discriminator. It is simply a DistilBERT model with a classification head. It was configured to have one Transformer layer. The Transformer layer hyper parameters were the same as the DistilBERT model. The parameters were initialized randomly. The classification head was a single hidden layer MLP with 768 hidden dimension and 0.1 dropout. The discriminator took the whole last hidden layer of the QA DistilBERT as input.

Middle Layer Adv This model is almost the same as the Last Layer Adv. The only difference is the discriminator took the whole 4th hidden layer of the QA DistilBERT as input. The hypothesis is that it might be inevitable to reveal domain specific information in the last layer because the answer distribution might be different across the domains. But the middle layer should be able to learn domain-invariant features.

5.3 Results

	DuoRC		**RACE**		**RE**		**Test**	
	F1	EM	F1	EM	F1	EM	F1	EM
DistilBERT	38.59	29.37	**40.04**	**28.12**	66.51	42.19	59.19	40.28
Original Adv	39.49	30.16	36.00	19.53	65.95	42.97	57.23	38.10
Simpler Adv	**43.04**	**32.54**	32.82	19.53	**68.77**	**45.31**	58.88	40.48
Last Layer Adv	42.01	**32.54**	32.22	17.97	67.78	**45.31**	**59.66**	**40.99**
Middle Layer Adv	37.05	28.57	31.92	18.75	67.71	41.41	-	-

Table 3: Robust QA track - eval results on each individual out-of-domain dataset and test results on the combined out-of-domain dataset.

From Table 3 we can see that the Original Adv model does not achieve better F1 score on RACE and RelationExtraction eval dataset compared to the DistilBERT baseline. The same was also observed in the original paper. The combined test score is also weaker.

It is a little bit surprising that the Simpler Adv model achieves better F1 scores than the Original Adv model on two eval datasets. It also achieves a better test score. This might be because a simpler model architecture and a stronger dropout regularization makes the discriminator less likely to overfit and provides more stable gradient to the QA model to learn domain-invariant features better.

The Last Layer Adv model is the best in terms of the test scores. This justifies our hypothesis that using the whole layer hidden states, the QA model eliminates domain specific features more comprehensively and adapts better to out-of-domain data. With a Transformer layer, the discriminator is also able to pick up cross token domain specific information.

The Middle Layer Adv model does not achieve better result than the Last Layer Adv model on the eval datasets. We didn't generate a test dataset score because we were limited to only evaluating on the test dataset for 4 times.

6 Analysis

The models perform the best on the RelationExtraction dataset. This is expected because this dataset is sourced from Wikipedia and two training datasets are also sourced from Wikipedia. DuoRC and RACE is hard for our models. RACE is particularly hard because it is from examinations. It has tricky examples like paragraph "...but it's a fact of daily life that 1.6 billion people around would have no electricity in their homes...by 2030, when the Earth's population will be likely to top 8billion, 1.3 billion people will still lack electricity..." and question "How many people still lack electricity in the world now?". The model needs to figure out the answer is "1.6 billion" not "1.3 billion".

We compared the answers for the RACE dataset to see why Adversarial Training is not helping. One thing we noticed was within the 85 answers that were different between the Last Layer Adv model and the DistilBERT model, Last Layer Adv's answer was a subset of DistilBERT for 11 times and DistilBERT was a subset of Last Layer Adv for 21 times. In other words, Last Layer Adv was two times more likely to predict a long and imprecise answer. For example, when DistilBERT predicted "one week", Last Layer Adv predicted "one week, it is important to note that one week in the life of a mouse is the same as about nine months". One possible explanation is since Adversarial Training is a regularization technique, it makes the model less opinionated. In other words, while reducing variants from the model, bias were increased in certain cases.

We also investigated more on why the Middle Layer Adv model did not achieve better result than the Last Layer Adv. This could be because the latter layers are very important in generating domain-invariant features. Because in this setup, gradients from the discriminator only flowed downward from the middle layer, the latter layers were not regularized. Figure 2 shows the discriminator accuracy

and loss $\mathcal{L}_D$ during training. We can see that the discriminator looking at the middle layer of the QA model achieved much higher accuracy than that looking at the last layer. This suggests that the middle layer is still handling lower level language details (like words and phrases) that's more domain specific. And the last layer is more responsible for higher level language structures (like grammars) that can be domain-invariant.

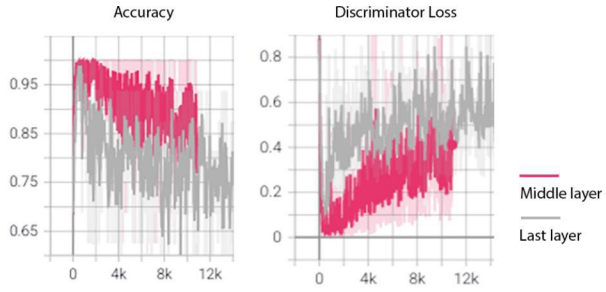

Figure 2: Discriminator accuracy and training loss from the Last Layer Adv model and the Middle Layer Adv model.

7 Conclusion

Within all the discriminator architectures we explored, the best one used a single layer Transformer encoder taking the whole last layer hidden states from the QA model as the discriminator. It achieved 2.43 F1 points on the test dataset better than the originally proposed Adversarial Training architecture. It was also found that a discriminator using the last layer hidden states performs better than using the middle layer hidden states. However, Adversarial Training might increase models' bias and make the model perform worse on certain datasets like RACE.

In terms of future work directions, we found during training that dev set metrics usually peaked around 1.5-2 epochs while training loss kept decreasing. This means despite the domain Adversarial Training regularization, the model still overfits to the training dataset. We could potentially find other losses besides the domain prediction loss that can further regularizes the model.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[2] 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.

[3] Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*, 2019.

[4] 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.

[5] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*, 2020.

[6] Seanie Lee, Donggyu Kim, and Jangwon Park. Domain-agnostic question-answering with adversarial training. *CoRR*, abs/1910.09342, 2019.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.