Robust Question-Answering using Adversarial Learning

Stanford CS224N Default Project

Aasavari Kakne
ICME
Stanford University
adkakne@stanford.edu

Samarpreet Singh Pandher
Civil & Environmental Engg.
Stanford University
samar89@stanford.edu

Vivek Kumar
SCPD
Stanford University
vivkumar@stanford.edu

Abstract

Question Answering (QA) is a critical task for NLP applications such as conversational agents and search engines in which generalization to new domains is highly desirable. Despite outperforming human benchmarks, state-of-the-art QA models often fail to generalize to new domains without significant fine-tuning. To address this challenge, Lee et al. [1] couples a pre-trained language model such as BERT [2] and a discriminator model (which predicts domain labels) so that the language model learns to predict features that are indistinguishable within the in-domain datasets. We aim to build a robust question-answering model by applying above-mentioned adversarial learning approach with pre-trained distilBERT [3] generator with a simple 3-layer discriminator. Our best model outperforms baseline and attains F-1 score of 58.63 and EM score of 40.14 on test leaderboard of RobustQA track. Alongside, we performed extensive experiments to determine impact of hyper-parameters on F-1 score and EM metrics which can be seen in results section.

1 Key Information to include

- Mentor: Kaili Huang
- External Collaborators (if you have any): None
- Sharing project: None

2 Introduction

Question Answering (QA) task is one of the highly sought-after areas in NLP research. Ability to generalize on new datasets a.k.a. robustness of QA models is desirable for numerous applications in wide range of domains such as Information Retrieval (identifying accurate answer given a search query, e-commerce chatbot which resolve customer questions) to Education (virtual teaching assistant that provides answers for students in remote areas).

Although QA models outperform human benchmarks in challenges such as SQuAD leaderboard, they often over-fitted and fail to generalize on new datasets by a huge margin. One might suggest the use of number of closed-domain QA models to resolve this issue. But, this solution is computationally expensive and it is difficult to identify and gather training data for all possible domains at development time. Thus, we must move towards open-domain Question-Answering. A step in that direction is construction of robust QA models. One intriguing approach of creating robust QA models is Adversarial Learning for Domain-Agnostic Question Answering [1].

Adversarial training technique has helped solve multiple challenges in deep learning networks in the computer vision and it intrigued us to learn about its application in the NLP domain. Previously, Lee et. al [1] explored using adversarial training framework for domain generalization in Question

Answering (QA) task. Their model consists of a BERT [2] as conventional QA model and a 3 layer MLP as a discriminator. The training is performed in the adversarial manner, where the two models constantly compete with each other, so that QA model can learn domain-invariant features.

For our default project, we are required to use pre-trained distilBERT as our QA model. Hence we employ distilBERT architecture as the Generator model within the adversarial architecture [3] with a simple 3-layer MLP as discriminator model.

3 Related Work

3.1 DistilBERT

There is a growing trend of use of transfer learning with large-scale pre-trained language models in Natural Language Processing. This has helped to significantly improve performance. However, there are several downsides to this trend, including environmental cost of training these large models and associated computational and memory requirements. In this context, DistilBERT was explored as a smaller and faster alternative to the massive BERT model. DistilBERT, a general-purpose pre-trained version of BERT, is 40% smaller, 60% faster, and retains 97% of the language understanding capabilities of BERT [3].

DistilBERT leverages the Knowledge Distillation compression technique where the DistilBERT as a student architecture is trained with a distillation loss over the soft target probabilities of the teacher, BERT. For the DistilBERT, the number of layers is reduced by a factor of 2 compared to BERT.

The authors of DistilBERT opted for a general-purpose pre-training distillation rather than a task-specific distillation. They proposed using a triple loss for training combining language modeling, distillation and cosine-distance losses aimed at leveraging the inductive biases learned by larger models during pre-training.

3.2 Adversarial Learning

In the adversarial approach for Domain-agnostic QA system, during training the QA model tries to fool the discriminator so that the hidden representation becomes indistinguishable to the discriminator. On the other hand, the discriminator is trained to classify the joint embedding of question and passage from QA model into the given known domains. If the QA model can project question and passage into an embedding space where the discriminator cannot tell the difference between embeddings from different known domains, we assume the QA model learns domain-invariant feature representation.

This approach is applied on MRQA Shared Task 2019 and has shown better performance compared to the baseline model. This technique helped the new model to outperform the baseline on DROP(DP), DuoRC(DR), RelationExtraction(RE), and RACE(RA) dataset by large margin. The model shows better performance in terms of EM (over 1.5 points) and F1 (over 2 points) on most of the test datasets except for ST.

3.3 Meta Learning

Authors of Lee et. al [1] also experimented with meta learning, however BERT being a very large model with millions of parameters, application of meta learning to do domain generalization did not work well. In order to maximize meta objective in both train and test domain, it needed to compute Hessian-vector products which slowed down training. Authors also tried to use Span refinement, to find the most plausible answer span in a sentence which is similar to the question in terms of cosine similarity, as golden span. The question and sentences in the passage are encoded into fixed-size vectors with universal sentence encoder. This approach boosts up the performance of some datasets but degrades the performance a lot in the other datasets.

4 Approach

4.1 Baseline

For baseline, we will fine-tune pre-trained distilBERT model [3]. More details are found in cs224n project handout [4].

4.2 Adversarial approach

For adversarial training approach, we select distilBERT as our pre-trained QA generator which learns to predict domain-agnostic features which are then fed into a discriminator. The discriminator accepts CLS features predicted by generator and learns to predict domain labels from $0, 1, ...K-1$ where K is the number of in-domains. In our case, the discriminator is a MLP model with 3 hidden layers, input size 768 and output layer is classification for K=3 with in-domains as SQuAD, NewsQA and Natural Questions.

The discriminator tries to predict domain labels and the generator tries to fool the discriminator by learning domain-agnostic features. Thus, they constantly compete with each other [5]. This can be seen in 2. More concretely, generator loss (equation 4) is composed of two terms - QA loss (where we maximize $p_{start}(i)p_{end}(i)$) (equation 1) and adversarial loss (where we minimize KL divergence between uniform probability distribution and softmax probability output from discriminator (equation 2). Thus, the generator learns to predict domain-agnostic features. On the other hand, discriminator loss (equation 5) is simply cross entropy loss of predicting the correct domain label.

$$\mathcal{L}_{QA} = -\frac{1}{N}\sum_{i=1}^{N} \log p_{start}(i) + \log p_{end}(i) \tag{1}$$

$$\mathcal{L}_{adv} = \frac{1}{N}\sum_{i=1}^{N} KL(U(l)||p_{\phi}(l(i)|h(i))) \tag{2}$$

Where KL divergence between probability distributions P and Q is defined as -

$$KL(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{3}$$

$$\mathcal{L}_G = \mathcal{L}_{QA} + \lambda \mathcal{L}_{adv} \tag{4}$$

$$\mathcal{L}_D = -\frac{1}{N}\sum_{i=1}^{N} \log p_{\phi}(l(i)) \tag{5}$$

5 Experiments

This section contains the following:

5.1 Data

In this project, we have been provided with three in-domain reading comprehension datasets (Natural Questions, NewsQA and SQuAD) for training a QA system which will be evaluated on test examples from three different out-of-domain datasets (RelationExtraction, DuoRC, RACE).

Each data point in above-mentioned datasets can be represented as (q, c) for question q and context c with label (s, e) such that answer $a = c[s : e]$ i.e. text in context starting at s and ending at e.

Statistical details of these datasets can be found in table below:

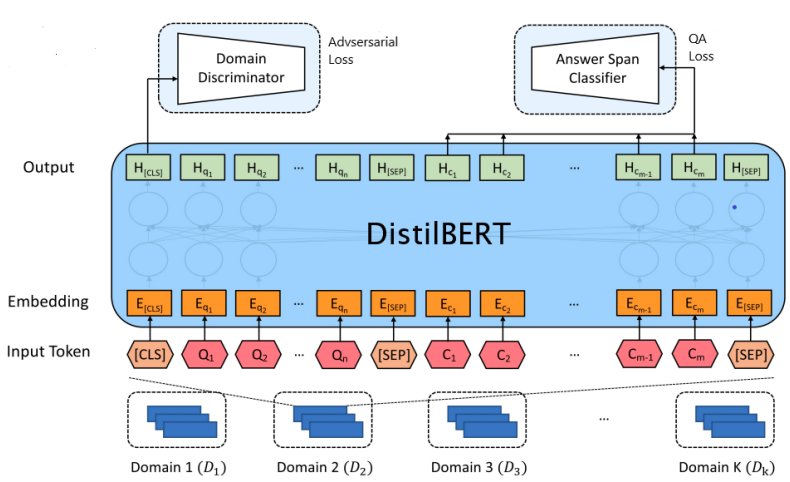

Figure 1: Overall training procedure for learning domain-invariant feature representation. QA Model learns to predict start and end position in the passage and fool discriminator for domain-invariant representation [1]

Dataset	Question Source	Passage Source	Train	Dev	Test
in-domain Datasets					
SQuAD	Crowdsourced	Wikipedia	50,000	10,507	-
NewsQA	Crowdsourced	News Articles	50,000	4,212	-
Natural Questions	Search Logs	Wikipedia	50,000	12,836	-
oo-domain Datasets					
DuoRC	Crowdsourced	Movie Reviews	127	126	1,248
RACE	Teachers	Examinations	127	128	419
RelationExtraction	Synthetic	Wikipedia	127	128	2,693

Table 1: Statistics for datasets used for building the QA system for this project. Question Source and Passage Source refer to data sources from which the questions and passages were obtained

5.2 Evaluation method

We used following metrics of success:

1. F1 score : the primary performance metric that will be used to rank submissions. F1 is the harmonic mean of precision and recall. It is a less strict metric compared to the other metric below.

2. Exact Match : a binary measure (i.e. true/false) of whether the system output matches the ground truth answer exactly.

5.3 Experimental details

We trained our baseline on indomain_train (training dataset composed of all three in-domain datasets) with learning rate of 3e-5, batch size of 16 for 2 epochs. We also tried baseline for 3, 5 epochs but best score found at 2 epochs.

For our adversarial experiments, we tuned dis-lambda (i.e. weight of adversarial loss), dropout and hidden_size of discriminator. At a time, we fixed two of them and changed only one of them in order to draw inference about impact of that hyper-parameter on performance. All of these experiments used learning rate of 3e-5, batch size of 16 for 5 epochs. The adversarial model is trained on V100 GPU for about 6 GPU hours (we can observe combined training loss of generator and discriminator in 2). All of these experiments are trained on indomain_train, validated on indomain_val and tested on oodomain_val. Additionally, we submit our predictions for oodomain_test to test leader-board on RobustQA track.

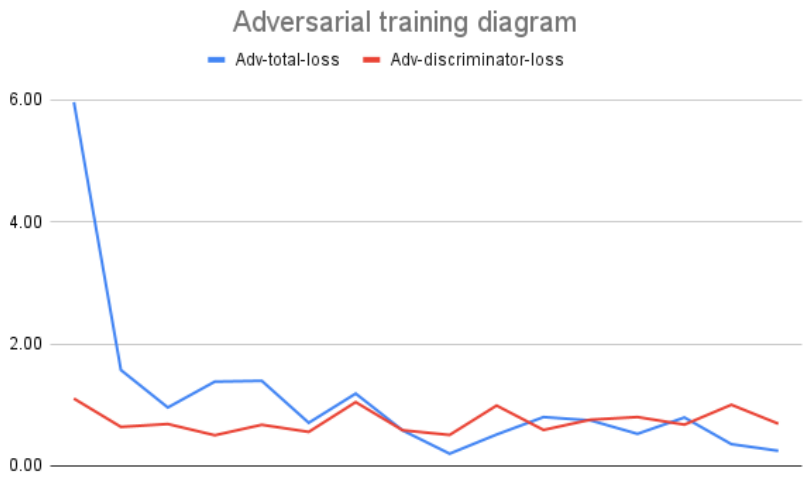

Figure 2: Here we see that after initial epochs, we can see that generator loss and discriminator loss compete with each other i.e. if one increases other decreases and vice-versa. [1]

5.4 Results

Params / Datasets	dis-lambda=0.5		dis-lambda=0.1		dis-lambda=0.01	
	F1	**EM**	**F1**	**EM**	**F1**	**EM**
indomain-val	70.23	53.87	70.61	**54.26**	**70.68**	54.25
oodomain-val-all	48.51	32.72	49.21	**34.55**	**49.75**	32.98
Race	34.37	19.53	34.2	**21.88**	**35.07**	21.09
DuoRC	41.15	33.33	**47.33**	**38.1**	46.96	34.92
RelationExtraction	65.55	**43.75**	66.06	**43.75**	**67.19**	42.97

Table 2: Discriminator lambda vs Score (F1, EM) on different datasets (fixed hyper-parameters are hidden_size = 768, dropout=2e-1)

As seem from table 2, F-1 score for oodomain_val steadily improves with decrease in λ. We believe that too large value of λ (when $\lambda = 0.5, 0.1$) introduces more noise in generator loss. Thus, by lowering λ (i.e. 0.01), generator is able to learn from QA loss as well as adversarial loss. This set of experiments yield our best model which achieves 49.75 on F-1 score for oodomain_val dataset.

Params / Datasets	dropout=0.2		dropout=0.1	
	F1	**EM**	**F1**	**EM**
indomain-val	**70.68**	**54.25**	70.29	54.07
oodomain-val-all	**49.75**	**32.98**	48.41	32.2
Race	35.07	21.09	**35.98**	**22.66**
DuoRC	**46.96**	**34.92**	41.31	31.75
RelationExtraction	67.19	**42.97**	**67.84**	42.19

Table 3: Dropout vs Score (F1, EM) on different datasets (fixed hyper-parameters are hidden_size=768, dis-lambda=1e-2)

From table 3, we observe that F-1 score for oodomain_val increases as we increase dropout from 0.1 to 0.2. We hypothesize that high dropout leads the discriminator learn domain labels using lesser information. So, it will not over-fit on training data. As a result the generator also improves by competing with the discriminator.

Params / Datasets	hidden size = 768		hidden size = 512		hidden size = 256	
	F1	EM	F1	EM	F1	EM
indomain-val	70.68	54.25	**70.90**	**54.85**	70.85	54.69
oodomain-val-all	**49.75**	32.98	49.47	32.46	47.89	**33.51**
Race	35.07	21.09	**37.62**	22.66	36.53	**23.44**
DuoRC	**46.96**	**34.92**	38.38	24.6	41.49	33.33
RelationExtraction	67.19	42.97	**72.25**	**50**	65.55	43.75

Table 4: Hidden Layer dimension size vs Score (F1, EM) on different datasets (fixed hyper-parameters are dropout=2e-1, dis-lambda=1e-2)

From table 4, we observe that F-1 score for out of domain validation set steadily increases with Hidden size of the Discriminator. We hypothesize that - smaller hidden size (i.e. 512 and 256) leads the model to under-fit the intricate boundaries between in-domains where larger hidden size (i.e. 768) is able to fit the boundaries better.

Model / Datasets	Our best model		baseline	
	F1	EM	F1	EM
indomain-val	**70.68**	54.25	70.43	**54.46**
oodomain-val-all	**49.75**	32.98	49.0	**34.82**
Race	35.07	21.09	**35.44**	**21.88**
DuoRC	**46.96**	**34.92**	41.69	33.33
RelationExtraction	67.19	42.97	**69.76**	**49.22**

Table 5: Score (F1, EM) of our model(hidden_size=768, dis-lambda=1e-2, dropout=2e-1) vs baseline on different datasets

As seen from table 5, our best model achieves F-1 score of 49.75 on oodomain_val i.e. 0.75 point improvement over the baseline. Additionally, our oodomain_val F-1 score improves by **5.27** points for DuoRC which means the model indeed generalizes to new domains. Further, our oodomain_val F-1 score declines by **2.57** for Relation Extraction (which is very similar to training datasets). Thereby, proving that our model is not over-fitting (as the baseline did) and rather generalizing to new domains. We saw a small drop of 0.37 points in oodomain_val F-1 score for RACE which was unexpected. But this drop is too small to draw inference from.

Lastly, our best model attains F1 score of **58.634** and EM score of **40.138** on test leader-board of RobustQA track.

6 Analysis

We analyze our approach on the robustness of adversarial training for QA model by evaluating F1 and EM score of both indomain and ooomain validation dataset. As we can see in the Table 5 above, we could improve F1 score on both indomain and oodomain datasets by 0.15 and 0.75 respectively using adversarial training approach.

question	gold truth	baseline_predictions	our_model_predictions
How often do doctors suggest teens to have an eye test?	once a year	once a year	about once a year
Where did the story take place?	sea	by the sea	by the sea
Where do people usually meet their friends in England?	in a pub	in a pub	a pub
What city is Karnaphuli Paper Mills located in?	Chittagong	Chittagong	Chittagong, Bangladesh

Figure 3: Examples for EM score comparison from gold truth, baseline and best model prediction

At the same time, EM score for our model dropped down compared to the baseline. To understand fall in EM score, we analyzed predicted and gold answers for both oodomain and indomain dataset as EM is very sensitive to the exact prediction. We can observe from figure above as how an insignificant but small difference in predictions may affect overall EM score.

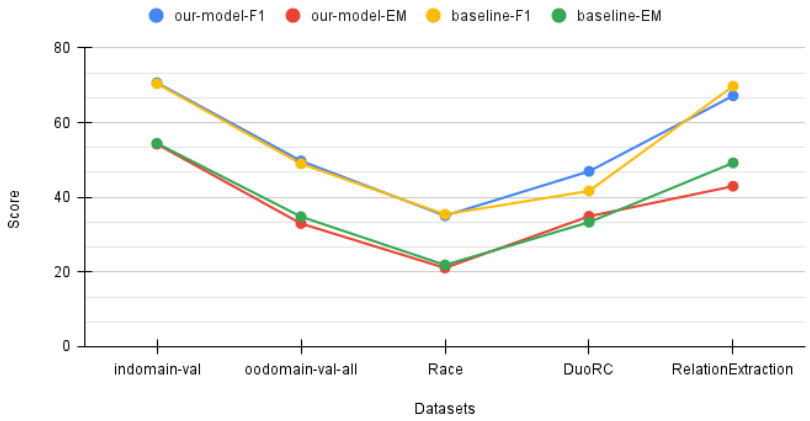

Figure 4: Plot of F1, EM score for our best adversarial model vs baseline for all datasets

To draw more intuition and generate more insights, we compared scores of individual out-of-domain datasets for both baseline and our best model, as shown in figure 4 above. We improved F-1 score over the baseline for dataset DuoRC by more than 5 points and EM score by more than 1.5 points. For RACE, We observe small decrease for our model in both F-1 and EM scores over baseline. We do see drop of almost 2 points for RelationExtraction with our best model, which is understandable as all indomain datasets are similar to RelationExtraction dataset. From these trends, we can conclude that the baseline model seems to be overfitted to in-domain datasets, where as our model performance generalizes better for out-of-domain dataset.

We experimented with batch size 16 and 32, this experiment did yield small improvement in EM but negatively impacted the F-1 score. We also trained our adversarial model for 20 epochs in order to draw insights about the training process. We found that the performance for adversarial model only improved till epoch 5, and worsened for later epochs.

We started with initial size of 768 for all 3 hidden layers on our discriminator model. In our quest to improve performance, We tried to boost the strength of discriminator by reducing size of 2nd and 3rd layer of the discriminator model to 512 and 256 respectively. However this did not improve performance for out-of-domain datasets in comparison to our best model. Score on RACE dataset improved in this experiment by 2 points but there was a negative impact on score of other two datasets. This suggests that 768 is best-suited size for hidden layers in discriminator architecture as it increases the ability of the discriminator to distinguish features from in-domain datasets.

7 Conclusion

We employ adversarial learning on Question-Answering task to learn domain-invariant features. On test leaderboard, our best model attains F-1 score of **58.63** and EM score of **40.14**. On validation leaderboard, our best model achieves F-1 score of **49.75** and EM score of **32.98**. Additionally, we performed extensive experiments to determine relationship of hyper-parameters with model's performance. Our main findings include -

- large values of λ hurts model performance by introducing too much noise in generator loss
- too small value of λ also hurts model performance because it reduces the importance of adversarial learning (setting $\lambda = 0$ means essentially reverting back to baseline)
- larger dropout improves model performance by preventing overfitting of the discriminator
- larger hidden_size in discriminator model architecture improves performance by better fitting the intricate boundaries between in-domains

We acknowledge that our model does not improve on RACE as it is very different from in-domains. To resolve this issue in future, we will work towards building a more robust discriminator so that generator would be able to generalize to out-domains that are significantly different from in-domain datasets. To leverage out-of-domain information during training, we can use limited amount of out-of-domain data points to improve QA model robustness as demonstrated in [6].

The code is available on github: https://github.com/aasavari-kakne/robustqa

References

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

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[4] 2022 CS224N teaching staff. Cs224n default project handout - robustqa track, February 2022.

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

[6] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.