ALP-Net: Robust few-shot Question-Answering with Adversarial Training, Meta Learning, Data Augmentation and Answer Length Penalty

Stanford CS224N Default Project

Zhen Li, Ran Duan, Yijing Bai
{lizhenpi, duanr, byj}@stanford.edu

Abstract

In this project, we built a robust question answering system called ALP-Net to do few-shot learning on out-of-domain data by adapting 4 different approaches, including domain-agnostic adversarial training, meta learning, data augmentation, and a new introduced answer length penalty. In our experiments, we found that adversarial training, data augmentation, and answer length penalty effectively improve the model performance on the out-of-domain datasets. Our best model achieved 60.962 F1 and 43.005 EM score on out-of-domain datasets test data.

1 Introduction

Deep learning has been very successful in natural language processing tasks in recent years. Among all the NLP tasks, Question Answering (QA) is especially challenging because it requires understanding on the relations between contexts and questions. It's more challenging for modern QA systems to generalize well on out-of-domain data. Many QA models could outperform human on a specific dataset [1], but fail on other unseen dataset [2]. This limits the application of the QA systems.

Thus, we built a robust question-answering system ALP-Net that can generalize well to out-of-domain data with few training examples. Our approaches include Domain-agnostic Adversarial Training, which helps model learn domain-invariant features; A new proposed Answer Length Penalty, which controls the answer length generated from the model and can greatly improve the answer exact match; Data Augmentation, which generated more augmented examples for oo-domain data; and lastly Meta Learning, which is experimented but no competitive results thus not included in the final model. Our experiments shows that with the combination of adversarial learning, answer length penalty, and data augmentation, we can achieve a competitive results of 60.962 F1 and 43.005 EM on out-of-domain test set.

2 Related Work

Pretrained Language Model has been proved to be very successful in various NLP tasks, including question answering. Models like GPT [3], BERT [4] and recent GPT-3 [5] are all large-scale pretrained language models, and they achieved state-of-the-art results on various NLP tasks including question answering. Among these pretrained models, DistilBERT [6] is a small, fast, cheap, and light Transformer model based on the BERT architecture, which is trained on large corpora by predicting the randomly masked tokens. Knowledge distillation is performed during the pre-training phase to reduce the size of a BERT model. It is a lightweight model that has great performance on various NLP tasks including question answering.

Question Answering is an important and difficult NLP task from both research perspective and practical perspective. There are several possible sources of the data including Wikipedia, QA communities, knowledge bases. There are also multiple types of tasks, including machine reading comprehension [7], answer selection [8], knowledge base [9].

Data Augmentation prevents the model to learn brittle correlation from the dataset by generating new training data from the existing training data. Approaches include words synonym replacement [10], BAE[11] (i.e. replace tokens by masking a portion of the text and using BERT to generate alternatives for the masked tokens), and SEAs [12] which are semantic-preserving perturbations that induce changes of model prediction to augment the training data.

Meta Learning mitigates the issue of generalizing a model that favors high-resource tasks and can be used as a good initialization for training and fine-tuning on low-resource datasets (i.e. fewshot learning) [13]. Recent works have attempted to apply meta learning to NLP tasks such that the model can achieve good results across domains with few-shot learning. Qian and Yu applied model-agnostic meta-learning (MAML) that generates dialogs in different domains with a few samples [14]; Bansal et al. applied optimization-based meta-learning that adapts to unseen natural language classification tasks with a few examples [15].

Adversarial Training is originally proposed in image generation field with Generative Adversarial Network (GAN) [16]. Then the concept of adversarial learning was adapted for domain generalization. Specifically adversarial training aims to train a domain discriminator and use the discriminator to encourage model to learn domain-invariant hidden features. Domain-Adversarial Neural Network [**?**] first applied adversarial training for domain generalization. Domain-agnostic Question-Answering [17] applied this approach on Question Answering and achieved good results. However, the previous work focused on zero-shot learning, while in this work we will explore how to effectively use adversarial training with few-shot training data on out-of-domain datasets.

3 Approach

Our model is shown in Figure 1. The system is built on top of DistilBertForQuestionAnswering from transformers library [18]. On top of that, we implemented a domain discriminator to help the QA model learn domain invariant features. During training, the discriminator is optimized to classify the domains of the data, while QA model is optimized to let the discriminator predict each domain equally. We also noticed that more than 95% of the answers are shorter than 6 words, while the prediction results contain more than 12% long answers. Based on this observation, we introduced an answer length penalty loss which computes an extra loss if the predicted answer is longer than a given hyperparameter k_{length}, to encourage model generates shorter answers. At last, to further improve the model robustness , we augmented the out-of-domain training data by randomly replacing words with synonyms.

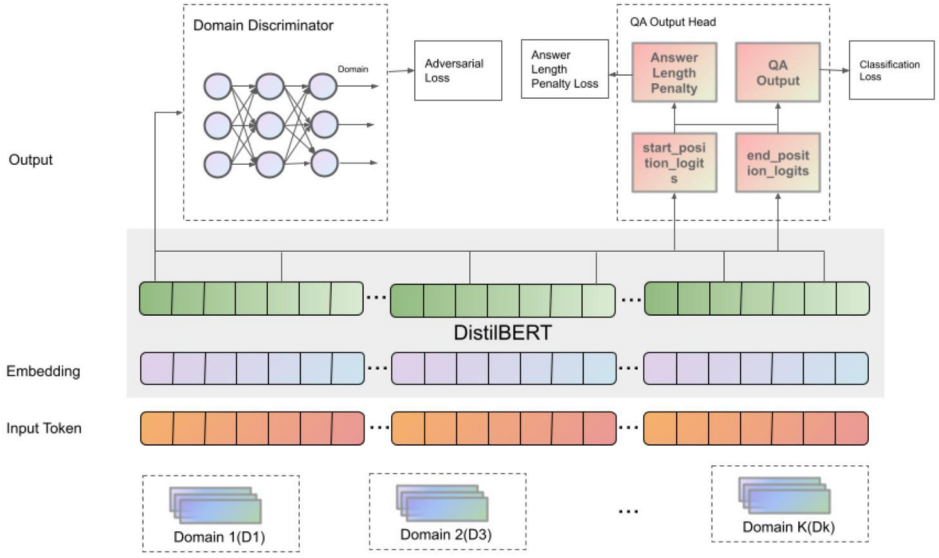

Figure 1: Model Architecture

3.1 Adversarial Training

Making sure deep models have domain invariant hidden layers is an efficient way to help model generalize to out-of-domain datasets [17]. Inspired by GAN [16] and domain-agnostic question answering [17], we implemented a domain discriminator on top of the last hidden layer of DistillBERT model output to encourage the base model to learn domain-agnostic hidden features.

We formulate the adversarial training as follows: A discriminator D is trained to minimize the cross-entropy loss as of equation (1), where l is domain category and h is the hidden representation of the last layer of DistilBERT; At the same time, the QA system is optimized to maximize the entropy of $P_\phi\left(l_i^{(k)} \mid \mathbf{h}_i^{(k)}\right)$ with an extra loss $\mathcal{L}_{adv}$.

$$\mathcal{L}_D = -\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N_k}\log P_\phi\left(l_i^{(k)} \mid \mathbf{h}_i^{(k)}\right) \tag{1}$$

In our implementation, we computes the discriminator loss $\mathcal{L}_{adv}$ using the Kullback-Leibler (BL) divergence between the uniform distribution over K domains and the discriminator's prediction logits. The discriminator loss is combined into the original QA loss after multiplied by a hyperparameter λ_{adv} with formula $\mathcal{L}_{QA} + \lambda_{adv}\mathcal{L}_{adv}$.

The domain discrimintor itself is a feed forward neural network with 3 hidden layers, predicting the log probability for each domain using log softmax at the end. Unlike domain-agnostic question answering [17], which only uses the last layer embedding at CLS word position as input to discriminator, we propose to use the whole last layer embeddings of DistilBERT as input to the discriminator. The intuition is that, since the output for question answering is computed based on the last layer embedding at each position, we need to make sure not only the embedding at CLS position is domain-agnostic, but the entire last layer embeddings should be domain-agnostic. This change will increases the discriminator input size and the amount of computation by $length_{context}$. To reduce the computation complexity, we reduce the size of discriminator hidden layer to 1/16 of the original size. Also, similar to GAN's algorithm [16], we choose to update the discriminator k_{adv} times for each batch in QA system to allow better convergence in the discriminator. Experiments show that both improvement make the discriminator works better.

3.2 Answer Length Penalty

After diving into the question answering datasets, we found that a large amount of the answers are relatively short. Especially on out-of-domain datasets duorc, race, and relation extraction, only 0.7%, 6.25% , 0% of the answers exceed 6 words long respectively. However, in the prediction of the baseline model on out-of-domain data, more than 12% answers are longer than 6 words. Thus we introduced two penalties based on answer length to encourage the model to generate shorter answers: Answer Length Penalty and Brevity Penalty.

Answer Length Penalty Answer length penalty aims at generating higher loss for long answers. It takes the predicted logits for start position p_{start} and end position p_{end} as input (both with shape $[sequence\_length, 1]$), where $p_{start}(i)$ / $p_{start}(i)$ represents the probability of the answer starting / ending at position i. For each starting position i, we consider all end positions beyond the range $[i, i + k_{length}]$ as invalid positions, as these positions either cannot form a valid answer with starting position i, or will form a answer longer than the given length threshold k_{length}. We sum up the p_{end} of each invalid ending position as loss $\mathcal{L}_{length,i}$ for starting position i. The final loss $\mathcal{L}_{length}$ is computed by a weighted sum for $\mathcal{L}_{length,i}$ based on the probability p_{start}. The formal computation of $\mathcal{L}_{length}$ is shown in equation (2). In the formula, $mask\_matrix(k_{length})$ is a pre-computed matrix with shape $sequence\_length * sequence\_length$ where, for each column i, $mask\_matrix[i : i + k_{length} + 1, i] = 0$ and other values are 1. We add $\mathcal{L}_{length}$ to the QA loss using formula $\mathcal{L}_{QA} + \lambda_{length}\mathcal{L}_{length}$. Our experiments show that after adding the answer length penalty loss with $k_{length} = 6$, the percentage of answers that exceed 6 words dropped from 12.79% to 11.5%. Implementation details can be found at [19].

$$\mathcal{L}_{length} = sum(diagonal(p_{start} * p_{end}^T * mask\_matrix(k_{length}))) \tag{2}$$

Brevity Penalty Inspired by the brevity penalty (BP) of BLEU [20], we introduced another penalty on length by multiplying the loss when the predicted answer is longer than the actual answer. The formula is shown in formula (3). Brevity penalty multiply the QA loss when the generated answers are long by a maximum of e, and only takes effect when the generated answer is longer. Experiments shows that this loss can also reduce the percent of answers exceed 6 from 12.79% to 12.36%.

$$\mathcal{L}_{qa} = exp(1 - \frac{length_{golden}}{length_{predicted}}) \times \mathcal{L}_{qa} \tag{3}$$

3.3 Data augmentation

To further improve the robustness of our model on out-of-domain datasets, we augmented the existing training data on out-of-domain training datasets, by applying the word synonym replacement augmenter from an existing data augmentation tool named nlpaug [21]. Specifically, we extract the contexts, questions, starting indices of each answer, and answer texts from the out-of-domain datasets (i.e. race, duorc, relation extraction). Then, we use nlpaug to replace some words in each context with different synonyms, formats, or forms. At last, we adjust the starting indices of answers and the answer texts accordingly with our algorithm. Specifically for every context, we replace 1 word at minimum and 30 words at maximum, each word with a probability 0.3 to be replaced. Our code is available on Github [22]. A sample data augmentation is shown in Table 1.

Original Text	Schweik is dragged down to the **cellar** where he is **savagely** beaten with heavy chains, tortured with quicklime acid.
Augmented Text	Schweik is dragged down to the **basement** where he is **viciously** flap with heavy chains, tortured with quicklime acid.

Table 1: Sample text augmented by synonym replacement and word addition.

3.4 Meta Learning

Meta learning is also one of the algorithm that can help the model generalize across datasets. We implemented meta learning and experimented on it. The algorithm conducts the following procedure: For each epoch, we sample a batch of datasets with probability proportional to their sizes (PPS), train different versions of the DistilBert model with Adversarial Training on each on them to obtain six different sets of model parameters, and perform a variant of Reptile MetaUpdate step to update the meta-learning parameters as introduced below. This algorithm is available as MetaLearningTrainer class on Github [23].

Data: Pretrained DistilBert model parameters $\theta_{pretrain}$, in-domain and out-of-domain datasets
Result: Meta-Learning parameters θ_{meta} that can be applied to the input DistilBert model
for *i **in** range* num_epochs **do**
  Sample a batch of datasets $\{T_i\}$ with probabilities proportional to their sizes;
  for *each* T_i **do**
    Compute $\theta_i = \theta_i - \alpha\times$ gradient loss on k samples; (θ_i is the parameter of the base model for this dataset)
  end
  Update meta-learning parameters by θ_{meta} =(1 - β) $\times$ θ_{meta} + $\beta\times$ the average of all θ_i;
  Propagate θ_{meta} to θ_i for all datasets;
end

Algorithm 1: Meta Learning Algorithm

4 Experiments

4.1 Data

Our data contains 3 in-domain reading comprehensive datasets (Natural Questions [24], NewsQA [25] and SQuAD [1]) and 3 out-of-domain datasets (RelationExtraction [26], DuoRC [27], RACE [28]). Each in-domain training dataset contains 50k training examples. The number of examples in each validation dataset ranges from 4k to 12k. Each out-of-domain dataset has 127 training examples, 127 validation examples, and a few thousands of unseen testing examples. Details are presented in Table 2.

Dataset	Question Source	Passage Source	Train	Val	Test
in-domain datasets					
SQuAD [1]	Crowdsourced	Wikipedia	50000	10,507	-
NewsQA [25]	Crowdsourced	News articles	50000	4,212	-
Natural Questions [24]	Search logs	Wikipedia	50000	12,836	-
oo-domain datasets					
DuoRC [27]	Crowdsourced	Movie reviews	127	126	1248
RACE [28]	Teachers	Examinations	127	128	419
RelationExtraction [26]	Synthetic	Wikipedia	127	128	2693

Table 2: Statistics for datasets used for building the QA system for this project.[29]

4.2 Evaluation method

For evaluation, we use two metrics, Exact Match (EM) score and F1 score. We validate our model using the EM and F1 score on out-of-domain validation datasets, and report the EM and F1 scores on the test set of out-of-domain data.

4.3 Experimental details

We trained our model on a combination of in-domain training datasets, out-of-domain training datasets, and augmented out-of-domain training datasets. All the following hyper-parameters are the best hyper-parameters searched using grid search by training on out-domain training data and evaluating on out-domain data (results available in Appendix).

For the QA model, we initialize the pretrained DistilBERT model using "distilbert-base-uncased" [18], set batch size to 16, apply AdamW with learning rate $3e-5$ as optimizer, and train 3 epochs in total. For the domain discriminator, we set learning rate to $3e-5$ and λ_{adv} to $1e-2$. The discriminator structure is a 3-layer feed forward network with hidden layer size 48. We udpate discriminator by 10 gradient steps for every batch. For the answer length penalty, we set the k_{length} to 6, and λ_{length} to 1. In data augmentation, for every context, we replace 1 word at minimum and 30 words at maximum, each word with a probability 0.3 to be replaced. For our best result we augmented each out-of-domain dataset 20 times. The data augmentation code and the full model implementation is available on Github [22][19]. Meta learning is trained separately and not integrated into the final model for quality reasons. The learning rate for meta-learning parameters is 1e-2; learning rate for base model is 3e-5; the number of meta-learning epochs is 2400; the number of datasets in a batch is 3; the number of samples trained in each dataset is 3.

4.4 Results

Our best result on the out-of-domain test dataset achives F1 60.962 and EM 43.005. The final model combined the improvement of adversarial learning, answer length penalty, and data augmentation. We also conducted experiments on potential improvements separately to understand the contribution of each approaches. The detailed results are shown in Table 3.

Model/Results (EM/F1)	oo-domain val	oo-domain test	in-domain val
Baseline	33.25/48.43		55.07/70.95
Baseline + oodomain data	34.29/50.75		53.82/70.06
Meta Learning	16.23/24.96		13.73/23.22
Domain Discriminator use CLS embedding	34.82/50.98		54.54/70.49
Domain Discriminator use full embedding	34.55/51.72		54.74/70.41
Answer length Penalty	36.39/50.97		53.76/69.89
Brevity Penalty	35.86/50.97		53.76/69.89
Data Augmentation	35.08/50.64		53.76/69.5
Domain Discriminator + Augmentation	34.55/51.72		54.74/70.42
Answer Length Penalty + Augmentation	34.82/52.66	41.28/60.653	54.32/70.46
Final model	36.13/51.51	**43.005/60.962**	53.86/70.06

Table 3: Validation set results for all the approaches separately and combined.

In general, data augmentation effectively improves F1; answer length penalty and brevity penalty improves EM by a large margin; while domain discriminator only slightly improves the result. Meta learning does not work well in our experiments.

Answer length penalty and brevity penalty contributed big EM improvement (35 to 36.39), this result is interesting as both of the loss penalty are original in our work. But we do observe a decent decrease in the long answers portion with our answer length penalties, which supported our results. While we were expecting bigger improvements from domain discriminator and meta learning, domain discriminator only slightly improves the model on out-of-domain datasets (EM 34.29 to 34.82), and the meta-learning does not improve the model. We believe the reason is that for domain discriminator, the domain-specific knowledge on the out-of-domain datasets potentially can improve performance; for meta learning, it is hard to learn a general knowledge that could be applied to new domains. Furthermore, we see great improvements on F1 using word synonym data-augmentation (51.72 to 52.66), where we believe that augmented context and answers provided more examples to help model figure out the position of the answer. Although it maybe still hard to figure out the exact match (answer could also be augmented), more augmented examples could help on approximate match.

5 Analysis

5.1 Adversarial Learning

Adversarial learning with domain discriminator only slightly improves the result on out-of-domain datasets, while it was shown to be very effective for zero-shot learning in domain-agnostic question-answering [17]. To analyze this, we first want to make sure that QA model did contains domain specific features if not trained with $\mathcal{L}_{adv}$; second, we want to make sure our discriminator structure is able to classify domains accurately. As shown in Figure 2, when we set λ_{adv} to 0, the discriminator loss can be very close to 0, and domain classification precision close to 100%. This means that the last hidden layer of baseline model indeed contains domain specific features when trained without $\mathcal{L}_{adv}$. On the other hand, enabling adversarial learning can actually make the features domain-agnostic. As shown in Figure 2, when we enable adversarial training, the discriminator loss decreases in the beginning, but increases quickly and stays at a high level during the entire training. We see similar patterns regardless of the discriminator hidden layer size, whether the discriminator using full embedding or only CLS embedding, or how many discriminator update steps. We think it is a evidence that the last DistilBERT layer of the QA system won't have any domain specific features with adversarial training enabled.

We believe the oo-domain training data makes adversarial learning less useful. In domain-agnostic question-answering [17], the validation is done on dataset unseen at training time, thus domain-agnostic features from in-domain data is helpful. However, in our task, including out-of-domain

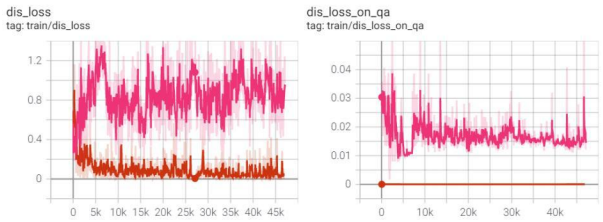

Figure 2: Discriminator loss (left) and $\mathcal{L}_{adv}$ of QA system (right) with λ_{adv} as 0 (lower orange line) and 1e-2 (higher pink line).

training data largely improved baseline results (improved EM from 33.25 to 34.29). That could mean allowing model to keep domain-specific features on oo-domain data could be beneficial to the results. One of the fix could be only do $\mathcal{L}_{adv}$ back-propagation for in-domain datasets, and skip for out-domain datasets. In that way the model will be able to learn domain-agnostic features from in-domain data, while keeping domain-specific knowledge from out-domain datasets.

5.2 Answer length penalty

Experiments show the new proposed answer length penalty and brevity penalty have been very effective in improving both performance. We think it is because these penalties successfully encouraged model to generate shorter answers. To confirm this theory, we counted the number of words longer than 6 generated by each model on out-of-domain validation set in Table 4. The results show that both penalties can reduce the percentage of long answers. One example answer predicted by the baseline model is *"vote annoys the other jurors, especially Juror 7 (Jack Warden"*, while the model with answer length penalty generated *"Jack Warden"* for the same question. These shorter answers effectively increase the chance for exact match. In fact, none of the models without answer length penalty can achieve EM more than 35.08, while with length penalty the model can easily achieve EM over 36.

As a conclusion, we believe that answer length penalty can be effectively generalized to Question-answering datasets with mostly short answers or long answers, but the limitation is that it requires prior knowledge on the length distribution.

Model	Percentage of answers > 6 words
Baseline + oodomain data	12.79%
Brevity Penalty	12.36%
Answer Length Penalty	11.5%
ALP + BP	10.97%
Golden	6.25%/0%/0.7% (race/relation/duorc)

Table 4: Persentage of long answers generated by each model on oo-domain validation set.

5.3 Data augmentation

Our experiments show that EM/F1 scores for the out-of-domain datasets are improved after augmented out-of-domain training dataset is applied. This could be explained by the nature of our model: Given the context and the question as input, the model needs to predict start and end positions of the answer. By augmenting each training example, both the answer and the start and end positions of the answer are very likely to be changed, thus preventing the model from making predictions directly based on the occurrence of certain words or remembering the positions of the answers.

Another observation is that the improvement on F1 scores is larger than the improvement on EM scores. This could also be caused by the changing start and end positions of the answers in the augmented data. For a specific training example and its augmented example, the questions are the same and the contexts are almost the same for both of them, while the start and end positions of the

answers can vary adequately. This could slightly impact the predictions of our model so that the predictions could be a few words away from an exact match. A few examples are shown in Table 5.

Question	Prediction	True Answer
Celebrations for Spring Festival in the UK started in .	in 1980	1980
What is the writer's attitude toward madness?	It seems that a little creative madness is good	little creative madness is good for us all

Table 5: Sample prediction and true answer comparison.

5.4 Meta Learning

Meta learning didn't perform well in our experiment. In the original paper [13], there is not quite a domain shift between the training datasets and the target datasets, as both contain textual similarity and relation classification tasks. The semantic embedding could be a general knowledge that is shared among different datasets, leading to success of the meta learning approach. However, from our experiments, the in-domain and out-of-domain datasets seem not to have such a general knowledge to be shared, as our meta learning model does not perform as well as the baseline model. As shown in Figure 3, we first trained a pretrained DistilBERT model with learning rate 3e-5 on both the in-domain and out-of-domain datasets, where the training loss decreased to 1 after 22k iterations; in comparison, we then applied our meta learning approach on the same pretrained model with the same learning rate, where the training loss decreased to 3 with lower decreasing rate and much larger oscillation after 22k iterations.

Besides, the experiment result shows that our meta learning model does not favor high-resource datasets comparing to transfer learning, as suggested by Gu et al. [30]. Yet finetuning on the parameters derived from the meta learning approach seems not to provide better results.

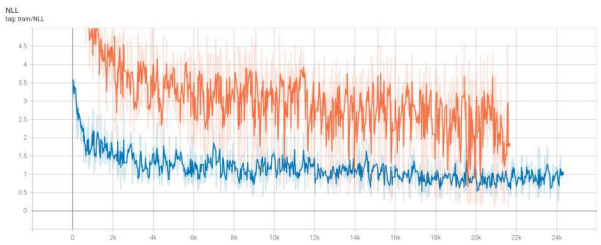

Figure 3: Train losses of the baseline model (blue) and the meta learning approach (orange).

6 Conclusion

In this project, we combined adversarial training, meta learning, data augmentation, and an original answer length penalty to built a robust fewshot question-answering system ALP-Net. We achieved 60.962 F1 and 43.005 EM on oo-domain testset. Our analysis demonstrated that answer length penalty is very effective in improving EM as most of the answers in datasets are short; data augmentation with word synonym replacement can effectively increase F1 score; domain adversarial training only has slight improvement when we have oo-domain training data; and meta learning does not perform well in our cross-domain question answering task despite its previous success in tasks such as natural language classification. The limitation of our work is that answer length penalty requires prior knowledge on answer length distribution. With more time, we would like to further explore the use cases and improvement of the proposed answer length penalty and brevity penalty as they are very effective in the experiments, it will be interesting to see if they can be a common approach for QA.

References

[1] Konstantin Lopyrev Pranav Rajpurkar, Jian Zhang and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *CoRR*, 2016.

[2] Cyprien de Masson d'Autume Jerome Connor Tomas Kocisky Mike Chrzanowski Lingpeng Kong Angeliki Lazaridou et al Yogatama, Dani. Learning and evaluating general linguistic intelligence. In *arXiv*, 2019.

[3] Jeffrey Wu Rewon Child David Luan Dario Amodei Radford, Alec and Ilya Sutskever. Language models are unsupervised multitask learners. In *OpenAI blog 1, no. 8 (2019): 9.*, 2019.

[4] Ming-Wei Chang Kenton Lee Devlin, Jacob and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *arXiv*, 2018.

[5] Benjamin Mann Nick Ryder Melanie Subbiah Jared Kaplan Prafulla Dhariwal Arvind Neelakantan et al. Brown, Tom B. Language models are few-shot learners. In *arXiv*, 2020.

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

[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] Mengqiu Wang, Noah A Smith, and Teruko Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, 2007.

[9] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*, 2015.

[10] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[11] Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*, 2020.

[12] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[13] Keyi Yu Zi-Yi Dou and Antonios Anastasopoulos. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *arXiv*, 2019.

[14] Kun Qian and Zhou Yu. Domain adaptive dialog generation via meta learning. In *ACL*, 2019.

[15] Rishikesh Jha Trapit Bansal and Andrew McCallum. Learning to few-shot learn across diverse natural language classification tasks. In *arXiv*, 2019.

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

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

[18] https://huggingface.co/transformers/model_doc/distilbert.htmldistilbertforquestionanswering. In *github*, 2021.

[19] https://github.com/NExPlain/robustqa. In *github*, 2021.

[20] Todd Ward Kishore Papineni, Salim Roukos and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL*, 2002.

[21] https://github.com/makcedward/nlpaug. In *github*, 2021.

[22] https://github.com/NExPlain/cs224n/blob/duanr-test-branch/augmentation.py. In *github*, 2021.

[23] https://github.com/NExPlain/cs224n/blob/duanr-test-branch/meta_train.py. In *github*, 2021.

[24] Olivia Redfield Michael Collins Ankur Parikh Chris Alberti-Danielle Epstein Illia Polosukhin Matthew Kelcey Jacob Devlin Kenton Lee Kristina N. Toutanova Llion Jones Ming-Wei Chang Andrew Dai Jakob Uszkoreit Quoc Le Tom Kwiatkowski, Jennimaria Palomaki and Slav Petrov. Natural questions: a benchmark for question answering research. in association for computational linguistics. In *ACL*, 2019.

[25] Xingdi Yuan Justin Harris Alessandro Sordoni Philip Bachman Adam Trischler, Tong Wang and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *ACL*, 2017.

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

[27] Mitesh M. Khapra Amrita Saha, Rahul Aralikatte and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. In *ACL*, 2018.

[28] Hanxiao Liu Yiming Yang Guokun Lai, Qizhe Xie and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*, 2017.

[29] Robin Jia Minjoon Seo Eunsol Choi Adam Fisch, Alon Talmor and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. In *Workshop on Machine Reading for Question Answering (MRQA)*, 2019.

[30] Yun Chen Victor OK Li Jiatao Gu, Yong Wang and Kyunghyun Cho. Meta-learning for lowresource neural machine translation. In *EMNLP*, 2018.

A Appendix

Hyperparameter	value	F1(oo-domain)	EM(oo-domain)	discriminator precision
repeat oodomain data (repeat times)	1	24.62	15.97	
	3	24.28	15.45	
	5	27.03	18.85	
	10	27.14	18.06	
	20	**29.17**	**18.59**	
	50	28.65	17.28	
length loss k (lambda = 1)	1	24.07	15.45	
	3	24.07	15.71	
	5	24.26	15.71	
	6	**24.62**	**15.97**	
	7	24.51	15.71	
	10	24.21	15.45	
	1000	23.63	14.4	
length loss lambda (k = 7)	10	22.73	14.66	
	5	23.8	15.45	
	1	**24.51**	**15.71**	
	5.00E-01	23.77	14.92	
	1.00E-01	23.62	14.66	
	1.00E-02	23.41	14.66	
	0	23.42	14.66	
adv lambda	1	4.47	0	57
	1.00E-01	21.25	13.61	44.38
	1.00E-02	**25.1**	**16.23**	**85.16**
	1.00E-03	23.54	15.18	97.36
	0	23.4	14.92	97.09
adv steps (lambda = 1e-2)	50	21.89	12.57	83.91
	10	**25.1**	**16.23**	**85.16**
	5	22.46	13.09	88.49
	3	23.51	14.66	89.04
	1	23.02	14.66	96.26
	0	22.24	14.14	15.53
full adv vs CLS adv	CLS adv	22.33	13.35	94.87
	full adv	23.91	15.18	95.28
bp loss vs length loss	bp loss	24.65	14.92	
	length loss	23.91	15.18	
	bp loss + length loss	24.72	15.97	

Table 6: Grid Search Result Table