Exploring Domain Adversarial Training and Data Augmentation for Out-of-Domain Question Answering

Stanford CS224N Default Project: RobustQA

Deveshi Buch, Caroline Choi, Melinda Zhu
Department of Computer Science
Stanford University
{deveshi,cchoi1,melinda7}@stanford.edu

Abstract

Current question-answering (QA) models such as Internet search engines have not matched human-level generalization [1]. In this work, we explore the effectiveness of domain adversarial training and three different data augmentation techniques in improving out-of-domain generalization. We find that a DAT model with a gradient-reversal layer and increasing the weight of the adversarial loss, using learning rate scheduling, and out-of-domain finetuning improved upon the baseline on the out-of-domain datasets, attaining EM 35.86 and F1 50.19 on validation and EM 40.09 and F1 57.79 on the test set. While we currently do not observe improvement with data augmentation, further exploration of different techniques may result in increased robustness to various phrasing and wording of questions.

1 Key Information to include

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

2 Introduction

Models may learn superficial representations that fail to generalize to data distributions which are distinct from the train data (i.e. “out-of-domain" data) [2, 3, 4, 5]. Several recent works have proposed techniques for boosting out-of-domain performance. In particular, pretraining on large datasets [6] and domain-adversarial training (DAT), in which a model learns domain-invariant representations to fool a domain discriminator, have been shown to improve generalization to out-of-domain tasks, such as QA [7, 8, 9]. In addition, [6, 10] suggest that augmenting training data improves model performance on out-of-domain data.

In this work, we explore domain adversarial training, in-domain and out-of-domain data augmentation, and out-of-domain finetuning to build a robust QA system that performs well on out-of-domain datasets. We implement domain adversarial training on top of a pretrained DistilBERT model and investigate how several architectural and experimental changes affect our model’s out-of-domain performance. We also perform three different data augmentation implementations—random insertion, synonym replacement, back translation—and investigate the impacts of each on out-of-domain performance.

We find that a DAT-based method is most effective, and improves upon the baseline. This paper is structured as follows. We explain related work in 3, model architecture details in 4, experimental details in 5, and analyze explanations for model performance in 6.

3 Related Work

Several works show that domain adversarial training (DAT) can be effective in improving model generalization. Ganin and Lempitsky [7] demonstrate the effectiveness of deep adversarial training for domain generalization. They include a gradient reversal layer (GRL) in between a feature extractor and domain classifier in order to discourage class-specific feature learning. Building on this, Sato et al. [8] apply DAT for graph-based neural dependency parsing. They combine GRL with a shared gated adversarial network, where the gates select either domain-invariant and domain-specific feature representations. They demonstrate improved generalization, although they primarily emphasize their model's performance on data from low-resource languages' treebanks instead of out-of-domain data, which may provide a more direct assessment of domain-invariant feature learning. More recent work by Lee et al. [9] applies DAT for question-answering with BERT. They implement an answer span classifier and domain discriminator on top of a pretrained BERT, trained alternatively in order to learn domain-invariant representations. As [9] is most relevant to our DistilBERT-based QA task, we implement DAT based on this architecture and incorporate several training and architectural additions, including GRL based on [7].

Data augmentation has been demonstrated to be effective in learning robust language representations. Wei and Zou [6] investigate the effects of language-based data augmentation for text classification. While they note that benefits of pretraining potentially outweigh those from augmentation, they report some improvements with synonym replacement (SR), random insertion (RI), random swap, and random deletion. We implemented SR primarily because it modifies words without much disruption to overall sentence structure and meaning. On the other hand, RI likely alters text meanings due to inserting words that make the grammatical structure incorrect or misleading; given this, we wanted to determine whether RI would help or hinder model performance. We also noticed that [6] did not combine techniques, so we experimented with SR + RI since these individually resulted in relatively consistent performance gains across [6]'s experiments. Sugiyama and Yoshinaga [10] explore back translation (BT), which involves translating the input to a different language, translating back to the original language, and training on the result to effectively convert inputs to more generalized representations of meaning, to develop "context-aware" neural machine translation models. By translating to different target languages, their models learned more robust meaning of ambiguous words in source languages. In our QA adaptation of BT, we sought to make questions more robust to slightly different phrasing, with back-translated questions still retaining the overall original meaning. For this reason, we expected BT to be more effective than RI.

4 Approach

4.1 Baseline

Our baseline is adapted from HuggingFace [11] and consists of a pretrained DistilBERTForQuestionAnswering model optimized with AdamW and cross-entropy loss of the start and end positions. [1] contains additional details.

4.2 Domain Adversarial Training

We implemented DAT for question-answering, based on [9]. Our DAT model consists of a conventional QA model and a domain discriminator, which are trained alternatively so that the QA model learns more domain-invariant representations that the domain discriminator cannot distinguish between. For the conventional QA model, we implement an answer span classifier on top of a pretrained DistilBERTModel (distilbert-based-uncased from [12]), which takes in context-question pairs as input, and outputs start and end logits for the answer span. The QA model minimizes a weighted sum of a conventional QA loss and adversarial loss $\mathcal{L}_{QA} + \lambda_a \mathcal{L}_{adv}$ where λ_a is a hyperparameter. In particular,

$$\mathcal{L}_{QA} = -\frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{N_k} [\log P_\theta(\mathbf{y}_{i,s}^{(k)} | \mathbf{x}_i^{(k)}, \mathbf{q}_i^{(k)}) + \log P_\theta(\mathbf{y}_{i,e}^{(k)} | \mathbf{x}_i^{(k)}, \mathbf{q}_i^{(k)})],$$

where $\mathbf{y}_{i,s}$ and $\mathbf{y}_{i,e}$ are the start and end positions of the answer. The discriminator D takes as input the hidden representation of the [CLS] token concatenated with the representation of the [SEP] token

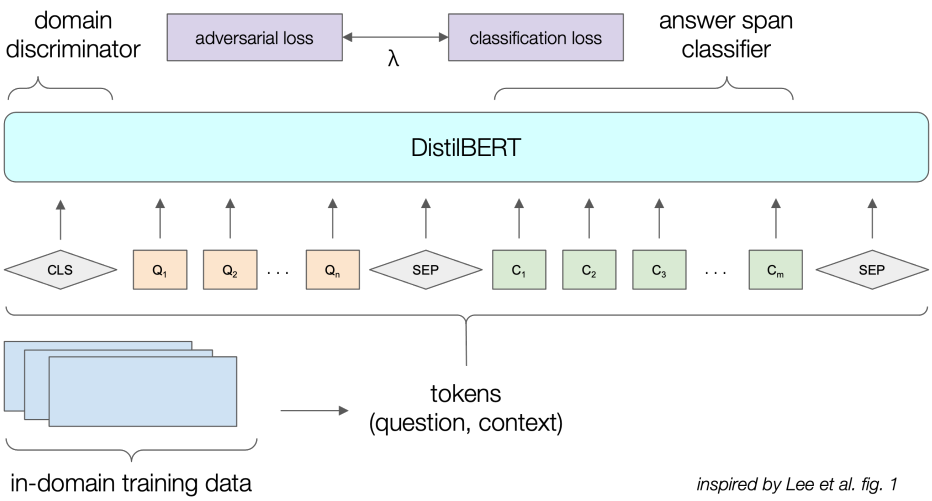

Figure 1: Overall DAT training process, with gradient reversal. The discriminator is a multilayer perceptron: GRL , [Linear($2 \cdot 768$, 768), ReLU, Dropout] , [Linear(768, 768), ReLU, Dropout] $\times$ 2. Here, $\lambda = \lambda_a$.

produced by the shared DistilBERT backbone. It predicts the domain of the input by minimizing the cross-entropy loss

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

where l is the domain label for the ith example and $\mathbf{h} \in \mathbb{R}^d$ is the (concatenated) hidden representation of the question and passage from the DistilBERT QA model. The adversarial loss then minimizes the Kullback-Leibler (KL) divergence between a uniform distribution over K classes, $U(l)$, and the discriminator's domain prediction:

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

We added a gradient reversal layer (GRL) between the DistilBERT outputs and domain classification according to [7, 8]. GRL acts as an identity transform on the forward pass and negates the gradient according to a hyperparameter λ_g on the backward pass. This negates domain-specific features learned by the model, encouraging it to learn more domain-invariant features.

Finally, we implement label smoothing based on [13] and scale all predicted domain labels by 0.9 before computing the discriminator loss to minimize training instability.

4.3 Data Augmentation on Questions

Random Insertion. Our random insertion (RI) procedure is modeled after [6] and involves obtaining random words within a question and inserting synonyms of these words in random locations. The purpose of RI is to promote robustness to unexpected perturbations that subtly change the grammatical structure of questions.

Synonym Replacement. Our synonym replacement (SR) procedure is also based from [6]: we replace random words in questions with their synonyms for robustness to varied wording. If a specific adjective was found in the question and context and that adjective was replaced by a synonym in the question, the model should be able to deduce the correct answer despite the different word that still implies similar meaning. The RI and SR implementations were adapted from [14].

Back Translation. Finally, we implement back translation (BT) based on [10]. The procedure involves translating original questions into another language, for which we used the Google Translate API [15], and translating them back to the original language (English, in our case) to capture more

generalized meanings that transcend domain-specific language details and preserve original semantics without significant word alterations. We chose to implement BT using French because it was used primarily throughout [10]'s experiments.

5 Experiments

5.1 Data

We use 3 in-domain datasets—Natural Questions (search log questions, Wikipedia passages) [16], NewsQA (crowdsourced questions, news article passages) [17], and SQuAD (crowdsourced questions, Wikipedia passages) [18]—and 3 out-of-domain datasets—RelationExtraction (synthetic questions, Wikipedia passages) [19], RACE (teacher questions, examination passages) [20], and DuoRC (crowdsourced questions, movie review passages) [21]. Data is in $\langle$context, question, answer$\rangle$ tuples, the answer a span from the context. Our training set contains primarily in-domain data (50,000 training examples per dataset); a small amount of out-of-domain training data (127 examples per dataset) can be used for finetuning. In-domain validation uses 10507, 4212, and 12836 examples from SQuAD, NewsQA, and Natural Questions respectively, and out-of-domain datasets contribute ~120 examples each for validation. The test set is entirely out-of-domain, with 2693, 419, and 1248 examples from the RelationExtraction, RACE, and DuoRC. Further dataset details are in Section 3 of [1].

5.2 Evaluation method

Answer span predictions from our model implementations are evaluated using exact match (EM) and F1 scores, useful for QA tasks, as described in [1]. These are then compared to the baseline (4.1) implementation's scores. Evaluation is performed on validation and test sets as described in 5.1, with val and test leaderboard results providing secondary evaluation insights.

5.3 Experimental details

5.3.1 Baseline

After loading the pretrained model (see 4.1), we initially trained the baseline using default hyperparameters: batch size of 16, learning rate of 3e-5, and 3 epochs, as well as $L_{\text{max}} = 15$, which is the maximum length of a predicted answer [1]. Subsequently, we finetuned this on the out-of-domain datasets described in 5.1 using 30 epochs and the same learning rate.

5.3.2 Domain Adversarial Training

We implemented DAT for question-answering based on the paper by [9], and adapted it for a DistilBERT backbone. We experimented with increasing the weight of the adversarial loss, learning rate scheduling, adding GRL, and freezing/unfreezing the discriminator during finetuning on the out-of-domain datasets.

All DAT models used a batch size of 16, learning rate of 3e-5, and all in-domain training was done for 3 epochs on one seed. In all experiments, both the DistilBERT QA model and the discriminator were optimized using PyTorch's AdamW optimizer [1], following [9].

Increasing λ_a. We set the weight of the adversarial loss, $\lambda_a = 0.5$ for one run (Model 7). For all other runs (Models 8-15), we initialized $\lambda_a = 0.01$ and increased λ_a across training steps according to $\lambda_a \leftarrow \lambda_a \frac{\tanh((t-3500)/1000)+1}{2}$, following [9].

Learning Rate. We originally used a fixed learning rate of 3e-5 to match the baseline. For Models 9-15, we used PyTorch's StepLR scheduler [2] to decrease the initial LR of 3e-5 by a factor of 1e-1 every 10000 training steps.

Gradient Reversal Layer. For the DAT models which include GRL (Models 10-15), we use $\lambda_g = 0.5$ following [8], as we focused on experimenting with other architectural and training differences between model runs.

[1] `https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html`

[2] `https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.StepLR.html`

Finetuning. We finetuned all models on out-of-domain train datasets with the default learning rate of 3e-5, which we found worked better than a lower learning rate of 3e-6. Models 13-15 froze the discriminator during finetuning, while Models 11-12 did not freeze the discriminator during finetuning. All finetuned models were finetuned for 3, 30, or 50 epochs as the out-of-domain set is significantly smaller than the in-domain set.

5.3.3 Data Augmentation

RI, SR, and BT are implemented as described in 4.3. The percent of words changed (α) in RI and SR was set to 0.1; higher values were experimented with, but we found best performance at this lower alpha value, which is consistent with [6].

5.4 Results

We find that DAT with GRL, learning rate scheduling, and longer out-of-domain finetuning produces the best out-of-domain performance with validation EM/F1 scores 35.86/50.19 and test EM/F1 scores 40.09/57.79. This constitutes our final submission to the dev and test leaderboards.

#	Model	indomain-val		oodomain-val	
		EM	F1	EM	F1
1	**Baseline**	54.54	70.31	**34.55**	**49.88**
2	SR Augmented Baseline	54.40	69.91	34.50	49.78
3	RI Augmented Baseline	50.06	68.29	31.40	46.26
4	SR+RI Augmented Baseline	53.33	69.33	32.20	47.28
5	BT Augmented Baseline	52.99	68.98	32.01	46.40
6	Baseline, FT 3E	31.15	47.29	29.58	44.90
7	DAT, $\lambda_a = 0.5$, fixed LR	51.72	67.92	29.84	44.22
8	DAT, increase λ_a, fixed LR	53.15	69.38	32.46	47.12
9	DAT, increase λ_a, LRS	54.12	70.02	30.89	47.80
10	DAT, increase λ_a, LRS, GRL	55.29	71.39	31.41	47.90
11	DAT, increase λ_a, LRS, GRL, FT 50E, DU	50.00	66.62	35.34	50.81
12	DAT, increase λ_a, LRS, GRL, FT 50E, DF	51.00	67.42	34.03	50.34
13	**DAT, increase λ_a, LRS, GRL, FT 30E, DF**	49.29	65.59	**35.86**	**50.19**
14	DAT, increase λ_a, LRS, GRL, FT 30E Aug, DF	48.37	64.24	33.25	47.12

Table 1: Summary of in- and out-of-domain validation results. FT = out-of-domain finetuned, followed by number of epochs as E, followed by "Aug" if finetuned on out-of-domain SR-augmented data; LR = learning rate; LRS = learning rate scheduler; DU = discriminator left unfrozen during finetuning; DF = discriminator frozen before finetuning.

We performed a variety of experiments using our implemented techniques, summarized in Table 1. Full list of experiments in Appendix Table 3.

5.4.1 Domain Adversarial Training

Increasing λ_a. We found that increasing λ_a over training (Model 8) as described in Section 5.3.2 rather than fixing $\lambda_a = 0.5$ (Model 7) significantly boosted both in-domain and out-of-domain performance, improving in-domain validation scores by +1.43/+1.45 (EM/F1) and out-of-domain validation scores by +2.62/+2.90 (EM/F1). During training, as the DistilBERT QA model loss $\mathcal{L}_{QA} + \lambda_a \mathcal{L}_{adv}$ decreases, this increases the weight of the adversarial loss $\mathcal{L}_{adv}$, encouraging the model to learn more domain-invariant representations.

Learning Rate. We found that using a step LR scheduler to decay the initial LR of 3e-5 by a factor of 1e-1 every 10000 training steps (Model 9) rather than a fixed LR (Model 8) boosted both in-domain and out-of-domain performance, improving in-domain validation scores by +1.21/+1.20 (EM/F1) and out-of-domain validation scores by +0.52/+1.33 (EM/F1). As the QA and discriminator losses decreased during training, decaying LR helped the model approach the local minima.

Gradient Reversal Layer. We found that adding GRL with $\lambda_g = 0.5$ with LRS improved our model's adversarial training stability and resulted in better convergence of our DistilBERT QA model

loss $\mathcal{L}_{QA} + \lambda_a \mathcal{L}_{adv}$, as shown in 2. By negating backpropagation gradients, GRL discouraged domain-specific feature learning in our model, improving in-domain validation scores by +1.17/+1.37 (EM/F1) and out-of-domain validation scores by +0.52/+0.10 (EM/F1) (Models 9, 10).

Freezing/Unfreezing Discriminator Before Finetuning. We found that freezing the domain discriminator before finetuning on the out-of-domain dataset improved both in-domain and out-of-domain performance when finetuned for many epochs. When finetuned for 50 epochs, freezing the discriminator (Model 12) improved in-domain validation scores by +1.00/+1.20 (EM/F1) and out-of-domain validation scores by +1.31/+0.47 (EM/F1) compared to unfreezing the discriminator (Model 13). Freezing the discriminator during finetuning likely improves performance because the model is leveraging the domain-invariant features already learned by the DistilBERT QA model during in-domain training. For our best model, we chose to finetune for only 30 epochs to avoid overfitting to the out-of-domain train dataset.

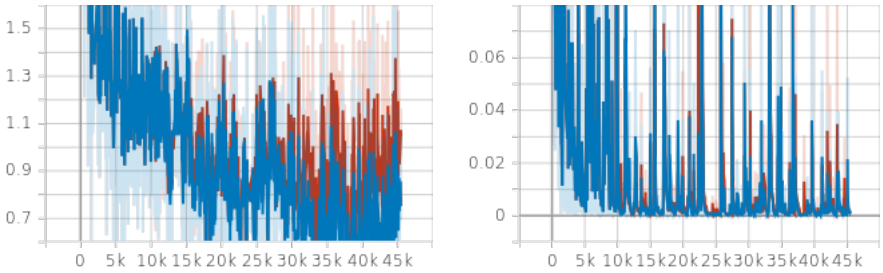

Figure 2: Train loss plots for DAT model with LRS, increasing λ_a, and GRL in blue and DAT model with LRS, increasing λ_a in red. Left plot is QA train loss and right plot is domain loss.

5.4.2 Data Augmentation

While we expected BT to perform well due to potential for generalizability, it performed worse compared to the baseline, and computational limits prevented us from testing other languages or delving into this further. One possible reason for our result may be that French translations were too structurally similar to English and the model failed to explore meanings that truly transcended language differences; alternatively, it may be that the Google Translate API was not sufficient for preserving original meanings, producing excessive changes to sentence structures. We found SR had better performance than RI and BT and investigate further in 6. We expected SR to improve on the baseline because of its ability to retain meaning while using different words, but we observed that words could be replaced with synonyms of the wrong part of speech (e.g. "What institute at Notre Dame *studies* the reasons for violent conflict?" becomes "What institute . . . *subject area* the reasons. . . "), which may have misguided the model's search for relevant context in some cases. We also observe lower-than-expected performance of Model 14 in 1, with SR-augmented out-of-domain train data potentially exhibiting similar issues during finetuning.

6 Analysis

We analyze the outputs of our best-performing model (Model 13 in 1) a domain-adversarially trained model with increasing λ_a, learning rate scheduling, GRL, and finetuned on the out-of-domain train dataset.

Example A:
Original Question: What year did Santer-Poos Ministry II start?
Context: The Santer-Poos Ministry II was the government of Luxembourg between 14 July 1989 and 13 July 1994.
Answer: 1989
Predicted Answer: 1989 and 13 July 1994

Example A illustrates a common weakness of our model. While it is able to identify the type of answer such as the date, it fails to understand the meaning of the word "start" in the question, and cannot extract the relevant year. It incorrectly includes unnecessary trailing dates.

Example B:
Question: How often do doctors suggest teens to have an eye test?
Context: Eyesight problems are common among all ages and if they are left untreated, they can cause serious headaches or other problems. The good news is that most eyesight problems can easily be sorted out by wearing glasses. Children and teenagers, under the age of 16 and up to the age of 19 for those full-time education, have the right to have eye tests for free in Britain. As the eye test is free, there's no excuse for not having a regular eye test. Doctors suggest that it's better to have an eye test about once a year. Wearing glasses isn't always regarded as all that cool and teens who suddenly need to wear glasses may find it difficult to accept. If they've grown up wearing glasses, then they may be more used to it.
Answer: once a year
Predicted Answer: about once a year

Meanwhile, Example B illustrates our model's ability to correctly synthesize a complex passage and reasonably answer a question about the passage. Our model answers the question just as accurately as the ground-truth answer, including the modifier "about."

Although data augmentation did not contribute to achieving our best model performance, we found that synonym replacement was more effective than random insertion due to more meaning preservation of the questions. The following ⟨question, context, answer⟩ examples illustrate our model's strengths and weaknesses, exposed by these two approaches.

Example C:
Original Question: In which stage will people feel most uncomfortable?
SR-Augmented Question: In which stage will individuals feel most anxious?
Context: There are four general stages of cultural adjustment. *(section omitted for brevity)* The second stage is called the withdrawal stage. The excitement you felt before changes to frustration as you find it difficult to deal with new problems. It is at this stage that you are likely to feel anxious or homesick. If you are one of those who manage to stick it out, you will enter the third stage – the recovery stage.
Predicted Answer (no augmentation): third stage
Predicted Answer (with SR): withdrawal stage *(expected)*

In Example C, SR contributed to answer accuracy in that the question was reworded to include a word, "anxious," that was found directly in the context. This demonstrates that without this particular instance of SR, the model is not quite robust enough to deduce that the word "uncomfortable" from the question is closely related to words in the context such as "anxious" and "frustration." For the original prediction, the system likely considered the word "stage" as a keyword in the question, and randomly chose an associated phrase in the context ("third stage") with this word. Therefore, the model could be improved, if space constraints allow, by generating additional instances of SR to allow the system to learn more associated phrasings of keywords within the question (in this case, the word "uncomfortable"). In the larger application of robust QA, models must be able to determine answers from context without the need for identical wording between question and context.

Example D:
Original Question: What is the name of the place where raphael cartoons can be found?
RI-Augmented Question: Make up what is localize the name of designation the place where raphael cartoons can be found?
Context: The Raphael Cartoons are seven large cartoons for tapestries, belonging to the British Royal Collection but since 1865 on loan to the Victoria and Albert Museum in London, designed by the High Renaissance painter Raphael in 1515-16 and showing scenes from the Gospels and Acts of the Apostles.
Predicted Answer (no augmentation): Victoria and Albert Museum *(expected)*
Predicted Answer (with RI): tapestries

Example D demonstrates the primary weakness of data augmentation techniques such as RI: a lack of meaning preservation. For this question, the main indicator of the type of answer expected is the phrase "what is the name of the place," which was modified significantly with the augmented

version with randomly inserted synonyms. Instead, the augmented question likely led the system to identify a phrase associated with the still-intact part of the question: "cartoons can be found?" This was answered through the following portion of the context: "the Raphael Cartoons are seven large cartoons for tapestries." In contrast with SR, RI can incorporate perturbations to a question to the extent that the question's core meaning is lost. Instead of making the system more robust to different phrasings that still preserve meaning, RI can avert answer searches toward incorrect locations in the context. As a result, SR, which generally retains meaning and sentence structure since a select few words are only replaced, was more effective than RI but did not contribute to the best model.

Overall, in our experiments we found that data augmentation does not result in any significant improvements to our model performance. This is likely the case because the model is too rigid to identifying identical phrasings and wordings in the context, which makes SR less effective, and answers questions primarily based on searching for sections of the context that match words in the question. Particularly with RI and BT, aspects such as part of speech and sentence structure are too heavily altered to sufficiently preserve the meaning of the original questions. Thus, in the case of QA, data augmentation may not be as useful in comparison to applications such as text classification or machine translation ([6], [10]).

Dataset	median context length (char)	Eval. F1	Eval. EM
DuoRC	3839	36.20	24.60
RACE	1632	30.35	17.97
RelationExtraction	129	77.40	57.81

Table 2: Out-of-domain dev set performance, by dataset.

While examining model output during analysis, we hypothesized that longer contexts would yield incorrect answers due to more opportunities for error and inefficiencies in finding identical or similar phrasings to questions. To determine whether a relationship existed between context length and answer accuracy, we applied our best-performing model (DAT with GRL, learning rate scheduler, and finetuning) to the individual out-of-domain datasets, which had varying median context lengths. As shown in 2, the EM and F1 scores on the RelationExtraction dev set were considerably higher than those of the other datasets, and happens to be comprised of short contexts in its examples. This is likely caused by (1) the passage source being Wikipedia, which could be more similar to the Wikipedia data in the in-domain training set, or perhaps more likely, (2) the shorter contexts allowing for the model's ability to pinpoint crucial sections of the context without being impacted by possibly "noisy" phrases in longer contexts. With long-context datasets such as DuoRC, while the model could identify the correct target phrase, the outputted prediction would also include unnecessary trailing phrases with no relevance to the question (see Appendix Fig. 3). This could be addressed by restricting context and/or answer length, or potentially using a search method to better identify the most relevant pieces of a larger context.

7 Conclusion

In this project, we explored the effectiveness of domain adversarial training and three types of data augmentation to improve out-of-domain generalization. Through several experiments and analyses, we found that our best-performing model was a DAT model with GRL that increased the weight of the adversarial loss during training, used learning rate scheduling, and was finetuned on the out-of-domain train set. Our model improved upon the baseline on both the out-of-domain validation and test sets, and we submitted our model to the class leaderboards. We contribute a DAT implementation for DistilBERT QA tasks, data augmentation applications to QA, and model result and output analyses. We also recognize some key limitations of this work. Our back translation implementation depended upon the Google Translate API, which was difficult to connect to reliably. For future work, we would find a more reliable BT implementation that may allow us to explore impacts of languages other than French on performance. For GRL, we followed [7] and used $\lambda_g = 0.5$ throughout; however other values of λ_g may yield more favorable results. We would also investigate different DAT architectures to QA, such as [8]'s gated approach. Finally, we would explore the effectiveness of other data augmentation techniques, such as LISA, a selective intra-domain and intra-label interpolation method [22].

References

[1] CS224N Staff. Cs224n default final project: Building a qa system (robust qa track). 2022.

[2] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

[3] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

[4] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[5] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*, 2020.

[6] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019.

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1180–1189. JMLR.org, 2015.

[8] Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. Adversarial training for cross-domain Universal Dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 71–79, Vancouver, Canada, August 2017. Association for Computational Linguistics.

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

[10] Amane Sugiyama and Naoki Yoshinaga. Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 35–44. Association for Computational Linguistics, December 2019.

[11] Michi Yasunaga and CS224N Teaching Team. Cs224n default final project: Building a qa system (robust qa track). `https://github.com/michiyasunaga/robustqa`, 2022.

[12] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[13] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[14] Jason Wei and Kai Zou. Code for eda: Easy data augmentation techniques for boosting performance on text classification tasks. `https://github.com/jasonwei20/eda_nlp`, 2019.

[15] Free and unlimited python api for google translate. `https://github.com/lushan88a/google_trans_new`, 2020.

[16] 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.

[17] 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.

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

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

[20] 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.

[21] 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.

[22] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. *arXiv preprint arXiv:2201.00299*, 2022.

A Appendix

#	Model	indomain-val EM	indomain-val F1	oodomain-val EM	oodomain-val F1
1	**Baseline**	54.54	70.31	**34.55**	**49.88**
2	SR Augmented Baseline	54.40	69.91	34.50	49.78
3	RI Augmented Baseline	50.06	68.29	31.40	46.26
4	SR+RI Augmented Baseline	53.33	69.33	32.20	47.28
5	BT Augmented Baseline	52.99	68.98	32.01	46.40
6	Baseline, FT 3E	31.15	47.29	29.58	44.90
7	DAT, $\lambda_a = 0.5$, fixed LR	51.72	67.92	29.84	44.22
8	DAT, increase λ_a, fixed LR	53.15	69.38	32.46	47.12
9	DAT, increase λ_a, LRS	54.12	70.02	30.89	47.80
10	DAT, increase λ_a, LRS, GRL	55.29	71.39	31.41	47.90
11	DAT, increase λ, LRS, GRL, FT 3E, DU	55.34	71.41	31.68	48.22
12	DAT, increase λ_a, LRS, GRL, FT 50E, DU	50.00	66.62	35.34	50.81
13	DAT, increase λ, LRS, GRL, FT 50E, DF	51.00	67.42	34.03	50.34
14	**DAT, increase λ, LRS, GRL, FT 30E, DF**	49.29	65.59	**35.86**	**50.19**
15	DAT, increase λ, LRS, GRL, FT 30E Aug, DF	48.37	64.24	33.25	47.12
16	DAT, increase λ, GRL	54.08	70.19	30.89	46.57
17	DAT, increase λ, GRL, FT 3E, DU	54.04	70.11	32.20	47.37
18	DAT, increase λ, GRL, FT 3E, DF	51.28	67.42	32.20	47.37
19	DAT, increase λ, GRL, FT 3E Aug, DF	47.88	63.93	34.82	49.07

Table 3: More comprehensive list of in- and out-of-domain validation results. FT = out-of-domain finetuned, followed by number of epochs as E, followed by "Aug" if finetuned on out-of-domain SR-augmented data; LRS = learning rate scheduler; DU = discriminator left unfrozen during finetuning; DF = discriminator frozen before finetuning. When LRS not used, fixed LR of 3e-5.

Question: who shouted daddy you brought the money
Context: In 1959, Michael Courtland (Robertson), a New Orleans real estate developer, has his life shattered when his wife Elizabeth (Bujold) and young daughter Amy are kidnapped. The police recommend that he provide the kidnappers with a briefcase of shredded blank paper instead of the demanded ransom, as the kidnappers will then be more likely to surrender when cornered, rather than attempt to escape with cash in hand. Courtland agrees to this plan. This leads to a bungled car chase in which both kidnappers and victims are killed in a spectacular explosion. Courtland blames himself for the deaths of his wife and daughter. Fifteen years pass. Courtland is morbidly obsessed with his dead wife, and regularly visits a monument he has had built in her memory. The monument is a replica of the church (Basilica di San Miniato al Monte) where he and Elizabeth had met many years before in Florence, Italy. His real estate partner Robert LaSalle (Lithgow) convinces Courtland to tag along on a business trip back to Florence. While there, Courtland revisits the church, and suddenly comes face to face with a young woman named Sandra (Bujold) who looks exactly like his late wife. The already slightly unhinged Courtland begins to court the young woman, and subtly attempts to transform her into a perfect mirror image of his dead wife. Courtland returns to New Orleans with Sandra so they can marry. On their wedding night, Sandra is kidnapped and a ransom note is left behind by her abductors. It is an exact replica of the kidnappers' message from fifteen years before. This time, Courtland decides to deliver the demanded cash. He withdraws massive quantities of money from his accounts and business holdings, financially ruining him and forcing him to sign over his interest in the real estate business to LaSalle. In the process, he discovers that his entire ordeal, including the original kidnapping, had been engineered by LaSalle as a way to gain sole control of Courtland's company share holdings. The now nearly insane Courtland stabs LaSalle to death. Knowing that Sandra must have been a willing accomplice in the plot against him, he goes to the airport to kill the escaping woman. On the plane, Sandra has a flashback to her part in the scheme; she is in fact Courtland's daughter: following the original kidnapping LaSalle concealed her survival and sent her to live in secret with an Italian caretaker. Over the years, LaSalle has told her lies about Courtland, convincing her that her father had not paid the ransom because he didn't love her. Sandra, who has come to love Courtland, attempts suicide on the plane and is taken off the flight in a wheelchair. Courtland sees her and runs toward her, gun drawn. A security guard attempts to stop him but Courtland smashes the briefcase full of money against the guard's head, knocking him unconscious. The briefcase breaks open and all of the money flies out. Sandra, seeing the fluttering bills, stands up and shouts: "Daddy! You brought the money!" Courtland now realizes for the first time who Sandra really is, and father and daughter fall into a deep embrace.
Expected Answer: Sandra
Predicted Answer: Sandra, seeing the fluttering bills, stands up and shouts "Daddy

Figure 3: Example ⟨question, context, answer⟩ triple from the DuoRC dataset, demonstrating the flaw in the model's ability to predict the most concise answer to the question. These extra phrases can be considered as "noise" that should not be included in the final predicted answer.