BoBA: Battle of BERTs with Data Augmentation

Stanford CS224N Default Project
Track: RobustQA

Ishira Fernando
Department of Computer Science
Stanford University
ishira00@stanford.edu

Rishi Desai
Department of Computer Science
Stanford University
rdesai2@stanford.edu

Abstract

Out-of-Domain Question Answering is a task that tests the ability of QA models to generalize to domains they were not previously exposed to during train-time. In this paper we conduct a survey of three methods for improving the out-of-domain performance of a pre-trained DistilBERT model: Mixture of Experts, Data Augmentation and Adversarial Training. We find that Adversarial Training is not able to improve domain generalization. Through our experiments on data augmentation and mixture of experts, we introduce **BoBA**, **B**attle **o**f **B**erts with Data **A**ugmentation), a QA model that combines Data Augmentation and Mixture of Experts. BoBA utilizes unfrozen, fine-tuned out-of-domain experts, along with synonym replacement and random swapping data augmentation to achieve a 5.17 point increase in F1 and 6.55 point increase in EM score over the baseline DistilBERT. Our evaluation on a held-out test sets demonstrates strong domain generalization with an F1 of 59.03 and EM of 40.69.

1 Key Information to include

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

2 Introduction

Modern NLP models are capable of learning vastly complex representations of languages. One application of such complex language models are to Question-Answering (QA). QA tasks involve providing a model with a passage, a question on that passage, and expects the model to highlight the position of the answer to the question within the corpus. One of the most researched QA tasks is the Stanford Question Answering Dataset challenge [1]. Since it's release in 2018, SQuAD has had many successful models improve on the state-of-the-art. Currently the best performing model outperforms human question answering by nearly four points (F1 score).

In many real world applications, NLP models are required to generalize to unseen examples from distributions different to the models' training distribution. However, adapting models to these distributions (known as domains) is difficult without directly fine tuning on them. In fact, many models that outperform humans on SQuAD perform significantly poorly on unseen datasets, indicating and inability to generalize beyond the training domain. *Domain generalization* is a measure of how well a model performs on data sourced from domains exterior to those of it's training data. It remains an open and difficult problem in the world of NLP.

Our Contribution In this paper we introduce **BoBA**: **B**attle **o**f **B**ERTs with Data **A**ugmentation. Boba is a QA model trained on the SQuAD [1], NewsQA [2] and Natural Questions [3] datasets that

generalize well to the DuoRC, RACE and Relation Extraction datasets. We achieve this by combining two approaches that have been shown to improve domain generalization: Data Augmentation and Mixture of Experts (MoE). BoBA improves the performance of the DistilBERT baseline by 5.17 F1 points and 6.55 EM points on the ood-validation set.

3 Related Work

DistilBERT One of the most widely utilized models for NLP tasks is the Bi-Directional Encoder Representation from Transformers model (BERT) and it's associated variants [4]. In particular, DistilBERT [5] has become extremely popular for QA tasks. DistilBERT itself is obtained by knowledge distillation. DistilBERT was constructed as a student model, and taught to learn the behavior of the teacher model, which in this case was BERT. By learning the behavior of BERT instead of the direct task that BERT is trained on DistilBERT is able to closely approximate the behavior of BERT. [5] asserts that DistilBERT is able to retain 97% of BERT's performance on the GLUE benchmark [6] with 40% fewer parameters and 60% faster inference times. For our experiments we use the DistilBERT model as both a baseline, and a general architecture to which we apply the methods mentioned in the introduction.

Data Augmentation Domain generalization is a widespread task that applies to many other sub-fields beyond just NLP. Augmenting training data through random transformations is well known to help with domain generalization and general robustness in other fields such as Computer Vision/Audio and Generative Modeling. In NLP, augmentation is often more complicated than it is for vision or audio tasks due to the complexity and structure of language data. One of the most widely cited methods for Data Augmentation for QA tasks is proposed in [7]. The authors propose a range of simple techniques which we outline in detail in 4.5, and adapt in tackling the problem of domain generalization. [7] also demonstrated that the strongest performance of augmented models is seen not when the model is fine-tuned on the augmented data instead of being trained directly on it. We test these conclusions in our experiments.

Another approach known to be highly beneficial to NLP tasks is pre-training. This is the process by which a model is trained on some other task (in the case of DistilBERT, usually language modeling) and using a different dataset, before being retrained on the actual training dataset for the task at hand. [5] uses the student-teacher method to pre-train DistilBERT for a variety of downstream tasks. We use one such pre-trained model in our experiments as detailed in 4.1

4 Approach

4.1 DistilBERT Baseline

We use a pre-trained DistilBERT for QA model as the atomic architectural component for all our models. We downloaded the "distilbert-base-uncased" model from Hugging Face [5] in accordance to the project specifications. The model consists of an encoder with six transformer blocks with 12 attention heads per attention layer.

4.2 Mixture of Experts

MoEs are a class of ensemble models that are widely used in domain generalization tasks. They consist of individual models (experts) that are trained on each individual domain in the training data. Then a gating function is trained on all the domains and learns to aggregate the outputs of each individual expert conditioned on the input:

$$\text{MoE}(x) = \sum_{i=1}^{k} g_i(x) f_i(x), \tag{1}$$

where k is the number of domains, f_i is the i-th expert, and g_i the gating output for the i-th expert. In essence the outputs of each expert are combined using a weighted sum, where the conditioned gating function produces the weights.

When building MoEs, there are two critical components: training the experts and finding the ideal gating function. We trained two different types of experts: in-domain experts and out-domain experts.

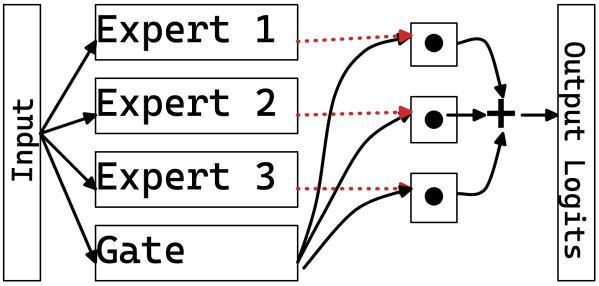

Figure 1: Architecture of a traditional Mixture of Experts

In-domain experts were trained on each of the in-domain training sets (one of SQuAD, NaturalQ or NewsQA). Out-domain experts were trained on the entire in-domain training set (all of SQuAD, NaturalQ and NewsQA) and then fine-tuned on one of the out-domain training sets (one of Race, DuoRC, or Relation Extraction). We trained several models that used either the in-domain experts or the out-domain experts in our experiments.

We experimented with multiple gating function architectures. We tried several different configurations (detailed in the appendix) with a Multi-Layer Perceptron (MLP). Due to the complex nature of the tokenized input passage and the associated attention masks, we also decided to use a significantly larger gate: DistilBERT itself. This DistilBERT was modified to have an output vector of shape $\mathbb{R}^k$ per example, where the i-th value is the weight $g_i(x)$ for the i-th expert's output.

4.3 Adversarial Training

Another method to improve domain generalization we tried was Adversarial Training (AT) [8, 9]. AT uses a discriminator model that attempts to classify the embeddings generated by the encoder model (DistilBERT) into one of the k domains in the training dataset. Both the discriminator and the encoder's QA head are trained simultaneously, and the encoder learns to produce domain invariant features. For brevity, we will very briefly summarize the adversarial training procedure.

Formally, the discriminator must minimize the following loss function, $\mathcal{L}_{adv} = -\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N_k}\log P_\phi(l_i^{(k)}|\mathbf{h}_i^{(k)})$, where $l_i^{(k)}$ is the domain category for the encoding $\mathbf{h}_i^{(k)}$. The final loss for training the model is $\mathcal{L}_{QA} + \lambda\mathcal{L}_{adv}$, where λ is a hyper-parameter for handling the impact of adversarial loss, and $\mathcal{L}_{QA}$ is the Categorical Cross Entropy specified in the project handout.

We adapted the codebase from [9] to use our training datasets and modified the model to use DistilBERT instead. The results in Table 1 show that it was unable to outperform the baseline. Hence, we did not pursue AT any further.

4.4 Data Augmentation

We adapt two approaches detailed in [7] to augment our training data: Synonym Replacement and Random Swapping. Synonym replacement entails randomly choosing words from the sentence that are not stop words, and then replacing them with one of their synonyms chosen at random. This enables the model to learn representations of similar words to those in the corpus that may not have been explicitly used otherwise, and in turn helps it become more robust to shifts in vocabulary. The probability of swapping a word in a sentence is controlled by a parameter β_{sr}.

```
Original:
Shi Qian is a fictional character in Water Margin, one of the Four Great Classical Novels of Chinese literature.
Augmented:
Shi Qian is a fancied character in Water Margin, unmatched of the Four big Classical novel of chinese literature.
```

Figure 2: An example of synonym replacement augmentation. Due to the nature of homonyms (multi-meaning) the sentence may become nonsensical with very high β_{sr}

.

Random swapping entails repeatedly choosing two words in the sentence and swapping their positions. This forces the model to pay more attention to vocabulary and word choice over the relative positioning of the words. It can also make the model more robust to formatting issues often found in open-source datasets. The probability of swapping a word in a sentence is controlled by a parameter γ_{rs}.

```
Original:
Ryan McLachlan is a fictional character from the Australian soap opera Neighbours, played by Richard Norton.
Augmented:
is Australian Ryan McLachlan fictional character from the a soap opera Neighbours, played by Norton. Richard
```

Figure 3: An example of random swapping augmentation. Very high γ_{rs} can result in extremely difficult passages to parse.

Augmentation can be tricky because we must ensure the answer (part of the string containing the answer) is unperturbed by augmentation and must accordingly update the start indices. We accomlpish this by implementing an augmentation splitter that splits the answer(s) for a given passage from the context, augments the context, and then reattaches the answer while updating the start index to account for the changed context. This preserves the overall meaning of the text and relative position of the answer while enabling augmentation.

4.5 Training and Hyper-parameter Tuning

We use Cross-Entropy Loss for our training:

$$L(\hat{y}, y) = \sum_{i}^{M} y^{(i)} \cdot \log(\hat{y}^{(i)}) \tag{2}$$

We use the AdamW optimizer [10] to train two model outputs, the logits for the start index of the answer, and the logits for the end index. Due to runtime and compute limitations, we hand-tuned hyperparameters. Parameters that we tuned (excluding those related to the gating function architecture) included β_{sr}, γ_{rs}, learning rate (α), batch size and the number of training epochs.

We train the model using Early-Stopping where the model is evaluated regularly as it trains on a validation set. The model with the best performance at all the evaluation points is then saved and used for further evaluation/development.

5 Experiments

5.1 Datasets and Evaluation Method

Our model is trained primarily on the *in-domain* SQuAD, NewsQA, Natural Questions and finetuned and evaluated on the *out-of-domain* DuoRC, RACE and Relation Extraction datasets. We performed data augmentation on all six datasets. We use EM (Exact Match) and F1 scores to evaluate the performance of the model on the available validation splits as required.

5.2 Single Domain Experts and the MLP Gate

Our first Mixture of Experts method involved training 3 experts, where each expert was trained on a single in-domain training set. We then used an MLP gate to combine the results. Note that we did not finetune on the ood-train set for fair comparison with the baseline. Table 1 shows how the performance of the exeprts and the MoE is far lower than the baseline, indicating a likely unusable approach to QA MoE models.

5.3 Training BoBA

The training pipeline for BoBA consists of several steps:

1. Train model M on the 3 in-domain train sets with data augmentation and validate on the 3 ood-train sets.

Model	In-Val F1	In-Val EM	Out-Val F1	Out-Val EM
DistilBERT Baseline	70.35	54.54	**46.86**	**30.89**
Squad Only	**75.67**	**61.93**	42.83	27.49
NewsQA Only	55.54	38.25	38.85	25.92
NaturalQ Only	66.82	50.79	36.70	20.68
MoE with MLP	62.13	45.12	41.85	25.65
Adversarial Training	20.62	N/A	12.11	N/A

Table 1: Baseline results for DistilBERT experts trained on *one* of the in-domain training sets.

Model	Batch Size	Learning Rate	Epochs	γ_{rs}	β_{sr}
Race Only	64	8e-7	3	0.00	0.30
Relation Extraction Only	32	1e-5	3	0.40	0.90
Duorc Only	32	1e-5	3	0.50	0.70
DistilBERT Gate (in-domain)	16	3e-5	1	0.00	0.00
DistilBERT Gate (out-of-domain)	16	3e-6	1	0.90	0.80

Table 2: Hyperparameters for training each expert. We used different hyperparameters when training the Gate on the in- and out-of- domain training sets. γ_{rs} and β_{sr} are the random sequence percentage and synonym replacement percentage, respectively.

2. Let expert E_i be M after finetuning and validating on the i-th ood-train set with data augmentation.

3. Train MoE model $B = f(E_1, E_2, E_3)$ on the 3 in-domain train sets without augmentation and validate on the 3 ood-train sets, where f is the gating function.

4. Finetune and validate B on the three ood-train sets with data augmentation.

5.3.1 To Freeze or not to Freeze

Our DistilBERT gating function has two varieties: i) one where we froze all but the last transformer block of each expert (B_1) and ii) one where the experts were completely unfroze (B_2). We hypothesized freezing the transformer blocks would preserve the learned integrity of the expert, so the expert would remain exceptionally performant on its respective dataset. However, our experiments showed that the unfrozen model severely outperformed the frozen variant. Because the experts are optimized in sync with the gating function, we believe this may have created a more cohesive MoE compared to the more discretized frozen experts.

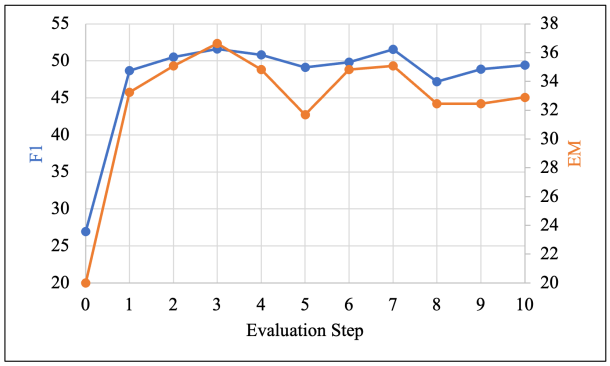

Figure 4: The F1 and EM scores during the single epoch of training the gating function on the in-domain training set with data augmentation.

Model	Out-Val F1	Out-Val EM
DistilBERT Baseline*	46.86	30.89
Unfinetuned Expert (M)	49.54	34.55
Race Only (E_1)	49.50	34.55
Relation Extraction Only (E_2)	50.06	35.86
Duorc Only (E_3)	48.91	32.98
MoE with Frozen Experts	43.21	26.71
MoE with Unfrozen Experts*	50.14	36.65
MoE with Unfrozen Experts	**52.03**	**37.44**

Table 3: The performance of the unfinetuned expert, the three experts, and two gating function variants of our MoE model. The astrix * denotes no data augmentation was used.

5.3.2 Experimental Details

Each of the experts E_1, E_2, E_3 was trained on all three augmented in-domain train sets. Then, each expert was finetuned on *one* of the augmented ood-train sets. We validated the expert only on the dataset it was being finetuned on, so it would become an expert at its own dataset. Table 2 shows the hyperparameters for BoBA's experts. Through experimentation, we found that high values of γ_{rs} and β_{sr} worked well for Race and DuoRC, but not for Relation Extraction. This may be due to the latter having a significantly narrower domain (in terms of domain, content and structure) than the former two datasets. As a result when excess augmentation significantly alters the structure of the dataset, the learned representations could no longer be representative of the domain, and lead to decreasing performance. The unfinetuned expert (M) and the MoE gate were trained with the same hyperparameters as the DistilBERT baseline: batch-size of 16, learning rate of 3e-5. We trained the gating function for only one epoch, because validation performance decreased due to overfitting. Figure 5.3 shows how the best performance was acheived relatively early in the epoch.

5.4 Baseline Model

Our baseline model is the vanilla DistilBERT for Question Answering trained on the in-domain train dataset for 3 epochs with learning rate 3e-5 with dropout and cross entropy loss. For more information refer to the description of DistilBERT [5, 11, 4] and the description of the baseline model in the project instructions.

5.5 Results

BoBA achieves **F1: 59.03, EM: 40.69 on the test set** and F1=52.03, EM=37.44 on the validation set. We were happy to see an increase of 7 F1 points, because this implies our model was able to generalize well to unseen data. We were surprised for the substantial increase, because we worried we had potentially overfit the model's hyperparameters to the validation set after extensive tuning.

Table 3 shows the scores of the various experts and MoE variants. The data augmentation (DA) evidently helped the model become more robust, as MoE with DA outperformed MoE without DA by almost 2.0 F1 points. Furthermore, our experiments show DA can create robust experts. We also see that unfrozen MoE's performance show us how how the model's performance is amplified when gating function is tuned in sync with the experts. We were surprised to learn that the training the gating function and the experts are not independent processes, as we assumed an unfrozen model would simply return to baseline performance.

6 Analysis

We will qualitatively analyze of our model to understand its improvement over the baseline. Figure 5 shows some sample outputs from our model for reference. We found that the model's predictions could be clustered into three significant groups: exact matches, overlapping matches, and complete misses. Overlapping matches are model predictions that contain the ground truth answer or are contained in the ground truth answer without being exact matches. Complete misses are predictions

Question
How does Yu-sun die?
Ground Truth Answer
drowns
Model Output
drowns

Question
Who's corpse does Dong-jin dismember ?
Ground Truth Answer
Ryu
Model Output
Yu-sun

Question
What was the name of the garden that Adam and Eve were cast out of?
Ground Truth Answer
Garden of Eden
Model Output
Garden of Eden

Question
Where does Stubby go hoping to get a job in the local casino?
Ground Truth Answer
Salt Flat Utah
Model Output
the small Wild West town

Figure 5: Sample model outputs.

that contain no overlapping text with the answer. We plotted the relative frequency of each type of miss in Figure 6.

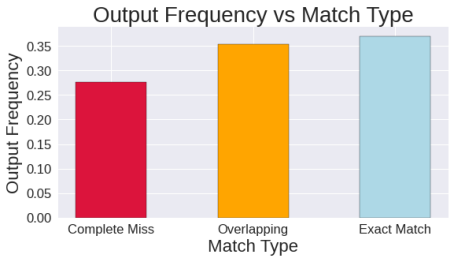

Figure 6: Match type of model predictions.

A significant amount of answers are overlapping (31.5%) but not exact matches. This indicates the model's utility as a QA system is stronger than the EM score might indicate at first glance as indicated by some sample overlapping answers in Figure 7. To investigate potential causes for this we plotted a histogram of answer lengths for both overlapping and missed answers in Figures 8 and Figures 9.

Ground Truth: "Facing the Flag" || **Pred:** "Jules Verne's 1896 novel Facing the Flag"
Ground Truth: "acid" || **Pred:** "acid. The others pull him back, but he dies as the acid"
Ground Truth: "Nazis" || **Pred:** "the Nazis"
Ground Truth: "farm" || **Pred:** "in a farm in the Dutch countryside."
Ground Truth: "lawyer" || **Pred:** "father's lawyer"
Ground Truth: "blond" || **Pred:** "dyed blond"
Ground Truth: "microphone" || **Pred:** "hidden microphone"

Figure 7: Sample overlapping answers.

For overlapping answers (Figure 8), we clearly see a different distribution in the answer lengths. The distribution of predicted answers is skewed right (longer tail) than the distribution for ground truth answers. We also see that the most common answer length is shifted further right (around 10) when compared to that of the ground truth (around 0-3). The differing distribution shape is also evident in the histogram of complete misses (Figure 9, where the same skewed tail, along with the right-shifted distribution is present. This implies our model is failing at finding the shortest correct answer as it appears to prefer guessing longer answers over shorter answers. Going forward, reducing this error may require the use of a length penalty through a custom loss function that penalizes non-exact matches that are longer more than those that are shorter.

7 Conclusion

In this project we developed BoBA, which uses two domain generalization techniques: data augmentation and mixture of experts. The combination of random swapping and synonym replacement along with unfrozen, fine-tuned experts and a DistilBERT gating function provided us with a 5.17 point

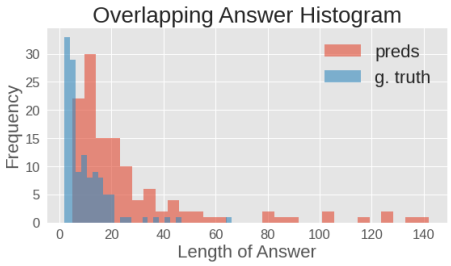

Figure 8: Answer-length histogram for overlapping answers in the OOD validation set.

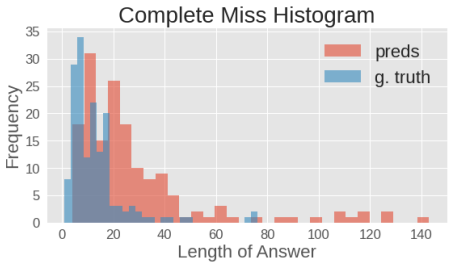

Figure 9: Answer-length histogram for complete misses in the OOD validation set.

increase in F1 score and 6.55 point increase in EM score. On an unseen test-set our model reached an F1 of 59.03 and an EM of 40.69 indicating strong generalization to the new domain.

7.1 Future Work

We did find many shortcomings with our approaches. We found that data augmentation is difficult to utilize when dealing with some of the fine-tune datasets, and the relevant augmentation parameters must be carefully fine-tuned on a case-by-case basis. Furthermore, hyper-parameters (largely learning rate and batch size) have a significant impact on the performance of the mode. While we performed the relevant tuning, we were not entirely thorough in our efforts due to our computational limitations. Going forward it would be prudent to perform this via a comprehensive grid-search. Furthermore, we felt that our exploration of the effects of layer freezing was incomplete. We were unable to train models that had different numbers of frozen transformer blocks, and as a result we felt that we were not utilizing our experts as efficiently as possible. Conducting an extensive search on this front would also provide us with valuable insights on improving the domain generalization of BoBA. We also neglected exploring other methods of augmentation such as random deletion and insertion as well as back-translation. We hope to explore these avenues in the future.

References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[2] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830, 2016.

[3] Tom Kwiatkowski, Jennimaria Palomaki, 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, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] 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.

[6] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461, 2018.

[7] Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. Data augmentation for BERT fine-tuning in open-domain question answering. *CoRR*, abs/1904.06652, 2019.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

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

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.