Comparing Mixture of Experts and Domain Adversarial Training with Data Augmentation in Out-of-Domain Question Answering

Stanford CS224N Default Project: RobustQA Track
Mentor: Mandy Lu

Irena Gao
Department of Computer Science
Stanford University
igao@stanford.edu

Nathan Marks
Department of Computer Science
Stanford University
nsmarks@stanford.edu

Ricky Toh Wen Xian
Department of Management Science and Engineering
Stanford University
rickytoh@stanford.edu

Abstract

Generalization is a major challenge across machine learning; Question Answering in Natural Language Processing is no different. Models often fail on data domains in which they were not trained. In this project, we compare two promising, though opposite, solutions to this problem: ensembling specialized models (a Mixture of Experts approach [1]) and penalizing specialization (Domain Adversarial Training [2]). We also study the supplementary effects of data augmentation [3, 4]. Our work suggests that Domain Adversarial Training is a more effective method at generalization in our setup. We submit our results to the class leaderboard where we place 10th in EM.

1 Introduction

While the field of Natural Language Processing has made huge advancements in the last decade, a remaining problem is the failure of models to generalize performance to datasets on which they were not trained. This failure may block models from being safely deployed, where all data will be new examples from potentially new distributions.

In this project, we study two promising methods for improving domain adaptation, Domain Adversarial Training [2] and Mixture of Experts [1], in the context of a Question-Answering task from NLP. We study whether these methods improve generalization between the datasets trained and validated on (the *in-domain* data) and disjoint, completely unseen data (the *out-of-domain* datasets). We additionally study the effect of semantics-preserving syntactical Data Augmentation [3, 4] on the performance of these two methods.

Our work suggests that Domain Adversarial Training is a marginally more effective method at generalization in our setup. We submit our results to the class leaderboard where we place 10th in EM and 12th in F1 out of 57 at the time of submission.

2 Related Work

Typically, models assume that problem data is independent and identically distributed so that all test-time observations will follow the same distribution as train-time examples. However, data observed at deployment — and even data from a similar dataset — can violate this assumption, a phenomenon called *distribution shift*. Recent work across all subfields of AI have found significant performance drops due to distribution shifts, suggesting that models learn spurious correlations unique to their train datasets, rather than invariant features they can leverage [5].

Recently, many methods have been proposed to handle distribution shift. A classic approach is to ensemble several "expert" models for prediction, each of which specializes in handling a certain distribution. The use of mixing multiple experts proved effective for Jacobs et al. in identifying and solving subtasks of spoken vowel recognition [1]. Out-of-domain question answering has a similar subtask structure in evaluating which trained domains the domain of a test question is most like. The corresponding expert models can then have the most weight in answering the given question.

An opposite approach has focused on forcing a single model to learn domain-invariant features by discouraging model specialization. Sato et al. proposed Domain Adversarial Training (DAT), a successful adversarial training method specific for domain adaptation. In this method, the model is penalized during training for output representations that vary widely by domain. By promoting domain-independent output, the model should be able to better generalize during out-of-domain testing [2].

Another classical add-on is data augmentation, in which additional examples are synthesized and added to increases the diversity of training data. In NLP, valid augmentations should be semantics-preserving while modifying syntax [3, 4].

3 Approach

We aim to compare the Mixture of Experts and the Domain Adversarial Training approaches, while also examining the ability of data augmentation to supplement both methods' performance. We review our approach below.

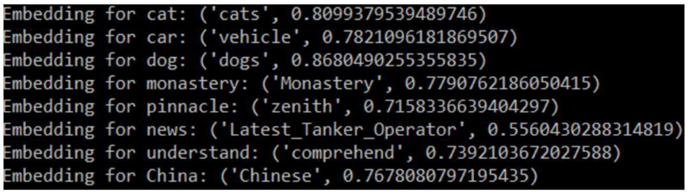
3.1 Data Augmentation

Our goal is to use data augmentation to improve the algorithms' performance on out-of-domain datasets. Augmenting data allows the model to be flexible with unseen data and to therefore generalize. Without augmentation, models tend to be somewhat brittle; they learn to memorize examples and not work as well on data with different syntax and vocabulary.

Our approach for data augmentation is based on methods by Ribeiro et al. [3] and Sugiyama and Yoshinaga [4]. Riberio et al., who successfully implemented data augmentation on question answering applications, suggest augmenting the most common paraphrases that generate adverse predictions using synonym replacement. We attempted a version of this method by using word2vec through the Gensim package [6, 7], to find the words with the most similar embeddings to an original word we would like to replace in a question. However, we determined, as shown in Figure 1, that this method generates examples that are closely associated to the original word (e.g. 'China' → 'Chinese') but does not generate substantially different words, and simple synonym replacement may result in grammatically incorrect sentences. As a result, we use back-translation, a method implemented by Sugiyama and Yoshinaga. Given a *(question, answer)* data example, we translate the question using external APIs from English to either French, German, or Chinese and then back to English (Figure 4). French, German, and Chinese were chosen because they are well-resourced languages that provide acceptable translations based on manual inspection. We only translate questions because answers must match the exact wording of the original context. Usage of APIs implemented as inspired by reference sites [8].

3.2 Domain Adversarial Training

One way to make a generalizable model is to learn domain-invariant features.We achieved this by implementing a domain adversarial training model based upon that of Sato et al. [2]. While they applied the technique to dependency parsing, we believe the domain-invariant learning will easily transfer to question answering. Additionally, where they used Bi-LSTMs, we use a transformer model in the form of Hugging Face's DistilBERT [9], as transformers are often more effective in calculating attention scores. The primary idea is to penalize the model for having hidden states that can be used



```
Embedding for cat: ('cats', 0.8099379539489746)
Embedding for car: ('vehicle', 0.7821096181869507)
Embedding for dog: ('dogs', 0.8680490255355835)
Embedding for monastery: ('Monastery', 0.7790762186050415)
Embedding for pinnacle: ('zenith', 0.7158336639404297)
Embedding for news: ('Latest_Tanker_Operator', 0.5560430288314819)
Embedding for understand: ('comprehend', 0.7392103672027588)
Embedding for China: ('Chinese', 0.7678080797195435)
```

Figure 1: Word2Vec most similar predictions.

to identify the domain of the input. This forces the model to not depend on the input domain, which should allow it to better generalize to domains it has never seen before.

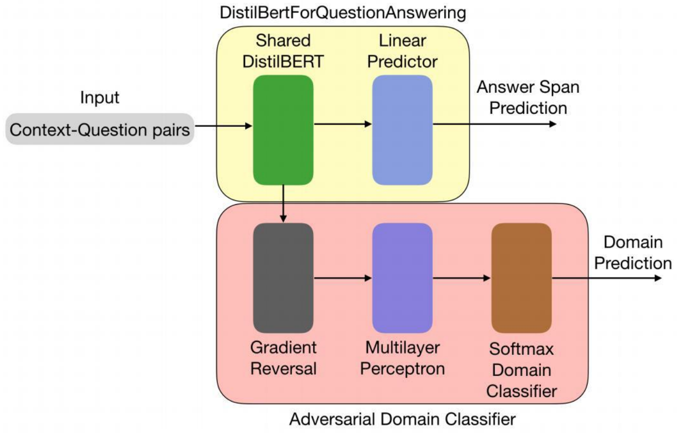


Figure 2: Diagram of domain adversarial training model.

Figure 2 portrays the structure of our model. Like in the baseline, context-question pairs are inputted into Hugging Face's DistilBertForQuestionAnswering, which is simply a DistilBERT model with a linear classifier head for predicting start and end logits for the answer span. During training, the DistilBERT model outputs not only for answer prediction but also for domain classification by the adversarial portion of the network, implemented as described by Sato et al. [2].

The transformer output first passes through a gradient reversal layer (GRL). In the forward pass, the GRL acts as an identity function. However, in the backward pass, the GRL multiplies the gradient by $-\lambda$, where λ is the GRL's only parameter and is set to 0.5. λ controls the influence of the domain classification objective in training the transformer. In this way, the GRL allows the adversarial portion of the network to update to better classify the domain while the transformer updates to hide the domain from the domain classifier. The GRL is followed by a feed-forward multilayer perceptron and a simple softmax domain classifier.

Once the DAT model has been trained on an in-domain dataset, it can be finetuned with an out-of-domain dataset. The adversarial portion of the network is not used in this scenario because different domains than the original training are used.

3.3 Mixture of Experts

A different approach to generalizability is to train multiple, specialized "expert" models instead of solely having one network. Jacobs et al. presented this idea for vowel discrimination in which each subnetwork specialized in a subtask of the problem [1], and a gating network mixed their outputs with normalizing constants. Here, we are taking a derivation of this method: we use one sub-network for each training domain, and a gating network that can appropriately determine for each example which expert should be used for the given examples. We use Hugging Face's

DistilBertForQuestionAnswering [9] for each of the experts. For our model, the gating network is a Multilayer Perceptron (MLP), as suggested by the default project. Each of the experts are trained (or fine-tuned) separately on different datasets; the gating network is then trained to choose between the pre-trained experts. Figure 3 portrays the structure of the model; Table 1 shows the structure of the MLP.

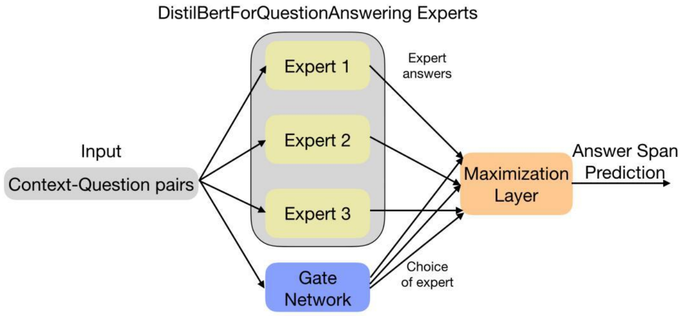


Figure 3: Diagram of mixture of experts model.

Layer	Input Size	Output Size
Linear Layer 1	384	192
ReLU	192	192
Linear Layer 2	192	384
ReLU	384	384
Softmax	384	384

Table 1: Gate Network: A simple 2-layer Multilayer Perceptron to determine experts. Input size is the example size, which for our dataset is 384.

The weights for the gating network are trained so that the expert with $Max(\sum G(x_{input})E(x_{input})_i)$ is chosen as the expert to be used. Here, where $E(x)_i$ is an expert and $G(x)$ is the output of the gating network given the input x. The loss is also defined as the $\sum G(x_{input})E(x_{input})_i$ of the expert chosen.

4 Experiments

Much of our pipeline code comes from the default RobustQA repository, at https://github.com/MurtyShikhar/robustqa. We've clarified in the following sections what additional code we wrote.

4.1 Data

Our data consisted of three in-domain datasets (SQuAD [10], NewsQA [11], and NaturalQuestions [12]) and three out-of-domain datasets (DuoRC [13], RACE [14], and RelationExtraction [15]). Each dataset was further split into train, val, and (where appropriate) test subsets. See Table 2 for details of each split.

In this report, we refer to dataset subsets by their category (in-domain or out-of-domain), split, and possibly specific dataset name and augmentation status. For example, `indomain-train-aug` refers to the train subset of the union of all augmented in-domain datasets.

We implement back-translation as a data augmentation pre-processing step. Given a *(question, answer)* pair, we translate the question from English to French and then back to English. Due to query limits, we distribute the translation task across four different API services, ranked by frequency of use: Microsoft Bing Translator [16], Google Translate [17], Baidu Translate [18], and Yandex

Dataset	Train	Train (Aug)	Val	Test
in-domain (`indomain-*`)				
SQuAD	50,000	62,140	10,507	-
NewsQA	50,000	71,029	4,212	-
Natural Questions	50,000	-	12,836	-
out-of-domain (`oodomain-*`)				
DuoRC	127	1012	126	1,248
RACE	127	964	128	419
RelationExtraction	127	999	128	2,693

Table 2: Size of our dataset splits. Because of API limits, we were not able to augment each in-domain set fully. We removed a few duplicate questions (where the back-translated English question is identical to the original) from the out-of-domain train datasets.

Translate [19]. We used our own implementation to use the API to back-translate existing examples and to integrate it into the train and validation datasets.

We ran augmentation on the `indomain-train` and `oodomain-train` subsets. Due to API limits, we were not able to augment all 150,000 examples in the in-domain train subsets.

To increase the size of our sparse `oodomain` dataset, we augmented all of our `oodomain-train` examples using French and *additionally* translated these examples to-and-from German and Chinese, increasing the size of each out-of-domain train split from 127 to around 1000. We additionally removed some duplicate questions (where the back-translated English question is identical to the original) from the out-of-domain train datasets. The exact sizes of our final datasets are given in Table 2. Example back-translation pairs are shown in Figure 4.

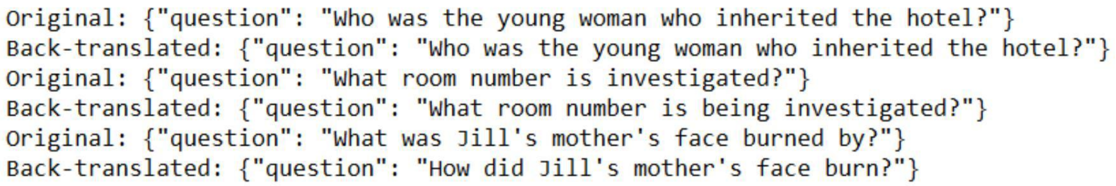
4.2 Evaluation Method

We evaluate each experiment using two standard metrics for question answering models: Exact Match (EM) scores and F1 scores. Exact Match is a binary score of whether or not the predicted answer is exactly the same as one of the gold standard answers. F1 is a harmonic mean of the precision and recall of the answer.

We report performance for each experiment on `indomain-val` and `oodomain-val`. We select the best model based on the `oodomain-val` performance to submit to the leaderboard and additionally report that model's `oodomain-test` performance.

4.3 Experimental details

Baseline Model. For our baseline, we use Hugging Face's DistilBertForQuestionAnswering [20]. It consists of a pretrained DistilBERT model with a span classification head on top. The loss is the cross-entropy of the start and end positions. We used the default hyperparameters specified by the starter repository: we used a batch size of 16, a learning rate of $\eta = 3\text{e-}5$, and we trained for 3 epochs on one seed. The baseline was trained on subset `indomain-train` and model selection was based on `indomain-val`. This model did not have access to augmented data.



```
Original: {"question": "Who was the young woman who inherited the hotel?"}
Back-translated: {"question": "Who was the young woman who inherited the hotel?"}
Original: {"question": "What room number is investigated?"}
Back-translated: {"question": "What room number is being investigated?"}
Original: {"question": "What was Jill's mother's face burned by?"}
Back-translated: {"question": "How did Jill's mother's face burn?"}
```

Figure 4: Back-translated examples

Domain Adversarial Training. We implement DAT ourselves based on the original paper [2]. For DAT models, we used a batch size of 16, a learning rate of $\eta = 3$e-05, and trained for 3 epochs on one seed. One DAT model was trained on `indomain-train` and the other on `indomain-train-aug`. Model selection was based on `indomain-val`.

The DistilBERT portion of each model was additionally finetuned using either `oodomain-train` or `oodomain-train-aug` for 3 epochs. Model selection was based on `oodomain-val`.

Mixture of Experts. We experimented with two sets of experts to determine what approach would be most effective. Experts are based on DistilBERT models, we implemented the gating network, with design inspiration mention in the Approach section.

1. *Mixture of Out-of-Domain Experts.* In the first mixture model, we trained three out-of-domain experts. First, we trained one baseline model (a DistilBERT trained on `indomain-train`) and then fine-tuned one expert per `oodomain-train-aug` dataset. This encourages each expert to specialize in an out-of-domain dataset, either Race, DuoRC, or RelationExtraction. The gating network is trained on `oodomain-train-aug`, *i.e.* the mixture model learns to weigh the experts in a way to maximize accuracy on the out-of-domain data.
2. *Mixture of In-Domain Experts.* In the second mixture model, we trained three in-domain experts. Each expert was trained on exactly one `indomain-train-aug` dataset and was not fine-tuned on out-of-domain data. This encourages each expert to specialize in an in-domain dataset, either SQUAD, NewsQA, or Natural Questions. We experimented with training the gating network on `indomain-train-aug` versus `oodomain-train-aug`, *i.e.* whether the mixture model learns to weigh experts to maximize in-domain accuracy or out-of-domain accuracy. We expect the in-domain experts mixture model to be more challenging, though it better reflects situations where we have no out-of-domain data a priori.

For all experts, we used a batch size of 16, a learning rate of $\eta = 3$e-5, and trained for 3 epochs on one seed to be consistent with our other models. Our out-of-domain experts were fine-tuned by training on `oodomain-train-aug` for 3 epochs with a learning rate of $\eta = 3$e-5. Additionally, our gating networks were trained for 3 epochs with a learning rate of $\eta = 3$e-5. We implement Mixture of Experts ourselves based on the original paper [1], with small design inspirations by Shazeer et al, Kang et al, and MLP tutorials [21][22][23].

4.4 Results

The results from our three experiments are presented in Table 3.

Domain Adversarial Training. Each Domain Adversarial Training model trained on the given in-domain dataset performed better than the baseline, while each DAT model trained on the augmented in-domain dataset did not perform as well. This may be because the adversarial portion of the network struggled to identify the domain of the augmented examples. The question in the input may have been one of the more useful factors in identifying the domain, so when it is changed by augmentation, the domain classifier would become less powerful, thereby mitigating its usefulness.

It is reasonable, though, that finetuning models generally improved their performance. Finetuning took place without adversarial training, so the above issue would have no effect. Ultimately, our most successful model was adversarially trained on the given in-domain training set and finetuned on the augmented out-of-domain set.

Mixture of Experts. None of our Mixture of Experts implementations reached the baseline level, though our Mixture of Out-of-Domain Experts was very close. The Mixture of In-Domain Experts performed significantly worse than the baseline. One reason why our Mixture of Out-of-Domain Experts may have had slightly lower, but very similar, results as compared to the baseline is the large size difference between `indomain-train` and each individual `oodomain-train` dataset. Because the fine-tuning datasets were so much smaller, it may be that our experts did not specialize enough, making the limited signal more detrimental than helpful. It seems like adding more data did not change the performance as the EM and F1 scores when trained on `indomain-train` and `oodomain-train` are the same even though `indomain-train` is significantly bigger. Similarly, the low performance for the Mixture of In-Domain Experts may have been because training the experts on each of the datasets may have created different models, but not ones whose specialization was optimal for the

	`indomain-val`		`oodomain-val`		`oodomain-test`	
Model	EM	F1	EM	F1	EM	F1
Baseline	55.07	**70.95**	33.25	48.43	-	-
DAT, no finetuning	54.68	70.63	35.08	50.21	-	-
DAT, finetuned on `oodomain-train`	55.20	70.75	34.82	49.80	-	-
DAT, finetuned on `oodomain-train-aug`	**55.22**	70.74	**35.34**	**50.50**	**42.385**	**60.525**
Augmented DAT, no finetuning	54.02	70.14	30.63	47.24	-	-
Augmented DAT, finetuned on `oodomain-train-aug`	53.93	70.08	31.15	47.55	-	-
MoE (out-of-domain experts), gated on `oodomain-train-aug`	55.01	70.88	32.46	47.87	-	-
MoE (in-domain experts), gated on `indomain-train-aug`	55.01	70.88	32.46	47.87	-	-
MoE (in-domain experts), gated on `oodomain-train-aug`	38.33	55.33	23.30	36.31	-	-

Table 3: Quantitative results for each experiment. All numbers reflect one seed. Finetuning refers to an additional 3 epochs of training on either `oodomain-train` or `oodomain-train-augment`. The "Augmented DAT" model refers to a DAT model trained on `indomain-train-aug` instead of `indomain-train`. The MoE "Out-of-Doman" experts refer to models trained for 3 epochs on `indomain-train`, and then fine-tuned for 3 epochs on an individaul `oodomain-train-aug` dataset, whereas the MoE "In-Domain" experts refer to models trained for 3 epochs on individual datasets of `indomain-train`. "Gated on" refers to which dataset the gate network was trained on.

Question Answering task. Instead, the gating network was just choosing models that were trained effectively on 1/3 of the data each, harming performance. Another potential factor to consider is that batch size was 16, which means that examples that would have had different experts were aggregated under one expert, diluting the ability for each expert to specialize.

Augmentation. Finally, we see that the augmentation of our `oodomain-train` data does provide a small boost to performance, approximately $+0.52$ EM between DAT fine-tuned on `oodomain-train` versus DAT fine-tuned on `oodomain-train-aug`. The effect of augmenting our `indomain-train` data is less clear, however. The Augmented DAT model (trained on `indomain-train-aug`) performs worse on our `oodomain-val` set than the standard DAT model (trained on `indomain-train`).

5 Analysis

For qualitative evaluation, we analyze a few examples *(Question, Prediction, Ground Truth)* to view in Tensorboard. There were various areas the models struggled. Questions phrased with synonyms could confuse the prediction.

> **Model:** Augmented DAT, finetuned on `oodomain-train-aug`
> **Question:** Whats the name of the English woman?
> **Context:** ...The CIA learns that its asset Tom Bishop (Brad Pitt) has been captured trying to free a Briton, Elizabeth Hadley (Catherine McCormack), from a People's Liberation Army prison in Su Chou near Shanghai, China. Bishop is being questioned under torture and will be executed in 24 hours unless the U.S. government claims him. If the CIA claims Bishop as an agent, they risk jeopardizing the trade agreement. Exacerbating Bishop's situation is the fact that he was operating without permission from the Agency. Attempting to deal quickly with the situation, CIA executives call in Nathan Muir (Robert Redford), an aging mid-level case officer on his last day before retirement and the man who recruited Bishop...
> **Answer:** Elizabeth Hadley
> **Prediction:** Robert Redford

It seems that because Elizabeth Hadley is described at “a Briton” instead of “English,” the model was unable to identify her. Sometimes predictions were not an exact match but were actually more specific versions of the correct answer. This shows the imperfections of the EM scoring system.

> **Model:** Mixture of Experts, indomain-expert
> **Question:** Which location offers the most direct view into daily life in the ancient world?
> **Context:** ...For travellers who want to experience some of the history and mystery of the ancient world, here is a list of cool destinations for your next holiday. Angkor Wat, Cambodia Built in the 12th century, Angkor Wat (meaning "capital monastery") was a temple in the ancient Khmer capital city of Angkor. .... Pompeii, Italy When Mount Vesuvius erupted in 79 A.D., Pompeii was buried under many layers of ash, preserving the city exactly as it was when the volcano erupted. Because so many objects were preserved, scientists and visitors are able to better understand daily life in the ancient Roman Empire....
> **Answer:** Pompeii
> **Prediction:** Angkor Wat, Cambodia

In the above example, we can see that the error is made likely because “Angkor Wat” is described as an “ancient” city, containing a key word in the question. While context clues, such as 79 A.D. shows the Pompeii also shows a view into the ancient world, this requires significant context clues from a wide range of sentences. Capturing this context is something our model needs to work on.

6 Conclusion

In this project, we compared Domain Adversarial Training with Mixture of Experts for increasing robustness in Question Answering. We found that Domain Adversarial Training is a more effective method at generalization in our setup, with a final out-of-domain EM score of 42.385 and F1 score of 60.525. Though there still remains a significant gap between in-domain and out-of-domain performance, we learned through this work that clever methods may improve model robustness.

Our work also suggests that data augmentation on small out-of-domain datasets gives a performance boost. We submitted our best model (using such augmentation) to the class leaderboard, where at the time of submission placed 10th in EM and 12th in F1 out of 57 submissions.

The two primary limitations of our work are

1. A lack of hyperparameter tuning — all reported experiments use the default learning rate. We did not tune learning rate for each experiment, though we did consider a learning rate one order of magnitude larger of $\eta = 3\text{e-}4$ for finetuning and observed worse performance. Additionally, the gradient reversal layer's hyperparameter λ was set to 0.5 for all experiments.
2. All experiments were only repeated on one seed — it's possible that the close performance between experiments, especially on `indomain-val`, could be due to chance.

In addition, our implementation using Mixture of Experts suggests that experts need to have significant differences (beyond simply fine-tuning differently) and that training on subsets of data is ineffective. This suggests that we should let the experts determine how to specialize by also back-propagating on the experts when training the gating network. We would also experiment with different batch sizes to see if there is a change in performance in the MoE, or we would find a way to structure the network so that each example can processed individually by an expert even within a batch.

In the future, we would like to correct these limitations and confirm the fairness of our comparisons.

References

[1] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geofrey E. Hinton. Adaptive mixtures of local experts. In *Neural Computation*, 1991.

[2] Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. Adversarial training for cross-domain universal dependency parsing. In *ACL Anthology*, 2017.

[3] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Association for Computational Linguistics (ACL) Anthology*, 2018.

[4] 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, Hong Kong, China, November 2019. Association for Computational Linguistics.

[5] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Sara Beery, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[7] Word2vec embeddings usage of gensim. `https://radimrehurek.com/gensim/models/word2vec.html`.

[8] Google trans new usage. `https://duyguaran.medium.com/how-to-use-google-trans-new-346ab827a4eb`.

[9] 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.

[10] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[11] 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.

[12] 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.

[13] Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. *CoRR*, abs/1804.07927, 2018.

[14] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[15] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. *CoRR*, abs/1706.04115, 2017.

[16] Bing microsoft translator api. `https://pypi.org/project/translators/`.

[17] Google translate api. `https://pypi.org/project/google-trans-new/`.

[18] Baidu translate api. `https://pypi.org/project/translators/`.

[19] Yandex translate api. `https://pypi.org/project/translators/`.

[20] Hugging Face. Distilbert.

[21] Chen Wen Kang, Chua Meng Hong, and Tomas Maul. Towards a universal gating network for mixture of experts. In *arXiv*, 2020.

[22] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *arXiv*, 2017.

[23] Pytorch: Introduction to neural network - feedforward/mlp. `medium.com/biaslyai/pytorch-introduction-to-neural-network-feedforward-neural-network-model-e7231cff47cb`.