Robust Q&BAE

Improving Out-of-domain Question Answering Performance with Data Augmentation Techniques Inspired by Adversarial Perturbation Methods

Stanford CS224N Default Project [Not Shared]
Track: RobustQA
TA Mentor: Eric Mitchell

Lynn Kong
ldkong@stanford.edu

Philip Weiss
weissp68@stanford.edu

Adam Pahlavan
adampah@stanford.edu

Abstract

Recent work by Siddhant et. al [1] has shown that BERT-based token replacement is an effective method for producing adversarial examples for natural language classifiers. We re-purpose these BERT-based token replacement techniques (BAE algorithms) and apply them to the domain of dataset augmentation. Specifically, we created a data augmentation pipeline in which we use a variation on the BAE algorithm to generate replacement tokens in question answering training data. We then use semantic similarity measures (USE or SBERT) to choose the best token replacement. Through this pipeline, we generated new training data which allows our model to generalize more effectively to unseen question answering examples. Our best performing model on validation was able to achieve an F_1 score of 52.80, and an EM of 38.74, significantly better than 48.21 and 32.46 for the baseline. Our best performing model on the test set was able to achieve an F_1 score of 59.58, and an EM of 42.25. We demonstrated that this data augmentation pipeline generates semantic coherent samples that can improve out-of-domain question answering model performance.

1 Introduction

Classical machine learning methods are based on the assumption that the training and test data come from identical data distributions. In cases where this assumption does not hold, it is common for models to exhibit a drop in performance on out-of-domain samples. The specific sub-field of machine learning aimed at the task of improving the generalization of models to out-of-domain data is known as domain adaptation [2].

One general class of approaches to tackling problems of domain adaptation is known as data augmentation (DA) [2]. In DA, the goal is to generate new synthetic data samples that look as if they were drawn from the out-of-domain data distribution and then to train a new model with these samples included. Since the newly trained model has adequate representation from both the in-domain (or source domain) and out-of-domain (or target domain) samples, the newly trained model can oftentimes exhibit improved accuracy on the target domain. Figure 1 visually shows how data augmentation could lead to a more out-of-domain accurate binary classifier in a simple 2-D case.

This project investigates the Robust QA domain adaption problem using a DA-based approach. The Robust QA problem asks us to improve the performance of a question answering system on out-of-domain data which has only a very small proportional representation in the training data. To evaluate performance, our model accuracy is gauged against validation and test set data which are compromised entirely of out-of-domain samples. The DA-based approach we will take is to use the out-of-sample training data available to us to bootstrap the creation of synthetic samples that represent new samples from the out-of-domain distribution.

Inspired by BAE adversarial sample generation [1], we created a data augmentation pipeline with multiple semantic and perturbation configurations. We trained the provided model on augmented

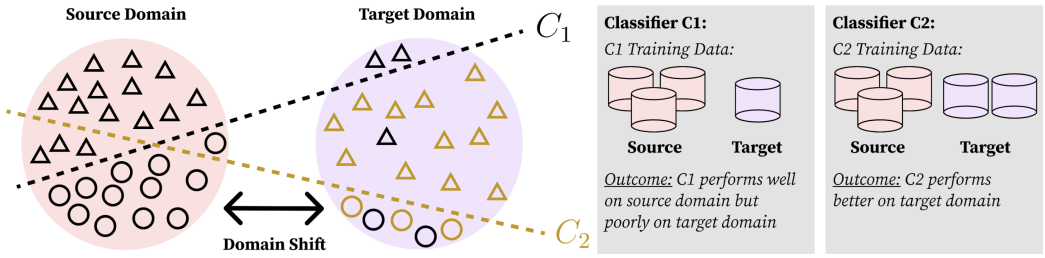


Figure 1: Depiction of data augmentation approach to domain adaptation on the training of a binary classifier. Classifier C_1 is trained on the black-outlined circles and triangles only, and classifier C_2 is trained with both black and gold-outlined circles and triangles. C_1 performs well on data in the source domain but performs poorly on the targeted domain due to exposure to limited data in the training set. C_2 performs much better on the target domain, albeit with slightly worse performance on the source domain. The styles for this figure were adapted from a figure in [3].

datasets with no other modification to the model architecture itself and saw significant out-of-domain QA performance improvement. Models trained on datasets augmented by BERT-MLM token generation and SBERT-based scorer performed best. We also investigated how the level of perturbation in the training set affects the model performance but were only able to see inconclusive results. Future work can easily expand on this pipeline and investigate new data mutation techniques to further improve out-of-domain QA performance.

2 Related Work

In this section, we will create a brief taxonomy of methods of tackling the domain adaptation problem and then categorize our approach within the taxonomy. A brief taxonomy is included in Figure 2, and the highest level of distinction is between supervised and unsupervised methods.

Supervised domain adaptation techniques rely on the use of labeled data in the target domain. On the other hand, unsupervised methods do not have labels for data from the target domain. Supervised methods are typically more powerful and successful at achieving domain adaptation since the labels are a powerful extra input to the problem [4]. The Robust QA problem focuses on the supervised category since our training dataset has labels for the out-of-domain data.

One layer deeper, we choose to focus on the data-centric method of DA as opposed to model-centric methods since we thought there was an opportunity to augment the very limited out-of-domain training set the problem started with (<500 data points were initially out-of-domain in a total training set of approximately 150,000 data points; 0.33% proportion of the initial training data was out-of-domain). Within data-centric methods, though we considered other methods including seq2seq and backtranslation as methods of tackling the Robust QA problem and thought they held promise as well [5], we choose to focus on rule-based token perturbations.

Specifically, we were inspired by the BERT-based adversarial example generation method (BAE) presented in the Siddhant et. al, [1] for text classification. This generation method selects tokens in the original input to mask, generates new text using BERT-based masked language modeling (MLM), and picks the best adversarial text by semantic similarity and degree to which the original model misclassifies the text. While our data augmentation does not require the adversarial optimizations in the BAE method, we leveraged the principle of the algorithm to create new semantically similar examples.

3 Approach

Our approach to improving performance on out-of-domain datasets for the RobustQ&A default project is through dataset augmentation. Specifically, we are co-opting techniques from Siddhant et. al, [1] which uses BERT-based token replacements to generate adversarial examples.

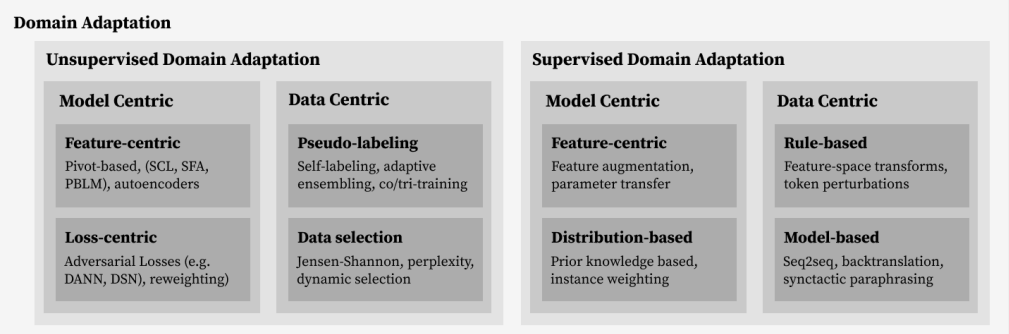


Figure 2: Overview of domain adaptation methods. The methods explored in this study are most similar to token perturbation techniques, which would fall under supervised, data-centric, and rule-based methods. This taxonomy was synthesized using the literature reviews in [6, 4, 5].

High level approach: Our technique works as follows: first, we created a class `BERTAdversarialDatasetAugmentation`, which is instantiated with a language model (WordNet synonyms [7] vs. DistillBERT-MLM), and a semantic similarity function sim (Universal Sentence Encoder [8] vs. pre-trained SBERT [9, 10]). For simplicity, we will refer to the language models as Synonym and BERT-MLM, and the semantic similarity function as USE or SBERT.

This class has a $perturb$ method, which takes in as input a sentence S, and a tuple of answer starts (the label corresponding to the question answer in the training data). The goal of the perturb method is to output a "perturbed" version of the original sentence for which semantic similarity to the original sentence is preserved. By passing an out-of-domain training dataset, one example at a time through $perturb$, we can generate a new training dataset, composed of semantically similar examples on which the baseline failed. By retraining from this new dataset, we aim to improve the performance of our question answering model on unseen data.

In our experiments section, we experiment with a number of these options (language model, sim function), but also with the number of perturbations per sentence, and the number of candidate perturbations to consider.

How does perturb work?: The original sentence is broken down into tokens on a word level, while only considering nouns as locations to perturb. On each iteration, we replace random k noun tokens in the original sentence with masks. Our language model (Synonym or BERT-MLM) creates the word fill-ins for this mask. This creates a new sentence- we then rank each of these sentences by similarity to the original input, using the sim function described above. This is either SBERT or a USE similarity score.

Baselines The baseline for our experiment was trained on the default project training data and came included with the starter code. Note, we described the model trained on augmented datasets with Synonym and SBERT as our secondary baseline in the Milestone, but we now categorize it as one of our Phase 1 experiments.

What part of this approach is original? What code did we write? We wrote the `BERTAdversarialDatasetAugmentation` class, as well as all necessary data pipelining, training, cleaning, and processing code. Our code can be found [here]. We used [11] as a guide for how to use DistillBERT for masked language modeling.

Our approach deviates from BAE since we are not looking for adversarial examples. The BAE paper continuously tries examples, until the baseline model fails on one. That model is essentially a random search through the BERT similar sentences. For us, we instead try to return sentences that are maximally similar to the original sentence. This is a major deviation since it means that instead of trying to find adversarial sentences, we are using similar techniques (BERT fill-ins) to find similar sentences.

4 Experiments

4.1 Data

Our team is using the question answering datasets provided in Robust Q&A, namely duorc, race, and relation_extraction for out of domain, and SQUAD, NewsQA, and Natural Questions for in domain. Since our project is centered around dataset augmentation, we also have augmented versions of each dataset, corresponding to the experiment that we are running. We are focusing on modifying the out-of-domain training examples, and each model will have a version of the three out-of-domain datasets that have been run through the perturb function. This will be discussed in further detail below.

4.2 Evaluation method

Our evaluation is the default one for the Robust Q&A Project. Namely, we have both a withheld test and validation set. We train on the training data (including the default training data, and our augmented training data), and then evaluate the F1 and EM accuracy on the validation set. Note that no augmentation is done on the validation set.

4.3 Experimental details

We ran our experiments in two phases. There are four main axes by which were were experimenting with the perturb function. Those are:

- Token generator: Synonym, BERT-MLM
- Semantic similarity function: USE-based, SBERT-based
- Number of mutations per sample: range from 1 to 5
- `num-index-upper-bound`, which controls number of augmented samples per original sample: 10, 20, 30, 50, 80 (conversion to number of samples in Table 1)

The cartesian product of these fields is 100, meaning that there are 100 possible models to train if we are to experiment with all of these combinations. Because each model takes about 3.5 hours to train on Azure GPU VMs, our team split our experimentation into two phases.

Total models trained: 16 :: $\sim$60 hours of training. Includes baseline model training.

Phase 1 - 4 models: We generated augmented datasets from original out-of-domain datasets with varied token unmasking methods (BERT-MLM vs. Synonym), as well as the similarity score function (USE vs. SBERT). We then trained models on those datasets and observed their validation performance.

Phase 2 - 11 models: Using the token generation and semantic similarity method from phase 1 that generated the best results (BERT-MLM+SBERT), we then proceeded to vary the number of mutations, and the `num-index-upper-bound`. `num-index-upper-bound` is our way to control the upper bound of the number of augmented samples generated from each original sample, thereby indirectly controlling the number of total augmented samples generated (Table 1). This way, we can investigate how the percentage of augmented samples in the overall training data affects the model performance.

We ran 5 experiments that varied the number of mutations per sample from 1 to 5 while keeping the index upper bound constant at 10. We ran another 5 experiments that varied the `num-index-upper-bound` (and in terms of the percent of total training data as augmented sample) while keeping the number of mutations per sample constant at 3. Note that we chose 3 mutations per sample so that we can guarantee we can generate enough unique augmented samples per original sample.

After the first 10 models, we examined the results and ran one more experiment using the best configurations from both Phase 1 and Phase 2.

Table 1: Number of augmented samples generated from each `num-index-upper-bound`, with number of mutations set to 3.

`num-index-upper-bound`	Total samples	% of total training data
10	6706	2.76
20	12537	5.15
30	16794	6.91
50	23836	9.81
80	31567	12.99

Table 2: Validation performance of the Phase 1 models.

Model Name	Token Gen	Semantic Sim	F_1	EM
Baseline-0.1	-	-	48.208	32.461
Synonym+SBERT*	Synonym	SBERT	47.927	33.77
Synonym+USE	Synonym	USE	50.424	36.649
BERT-MLM+SBERT	BERT-MLM	SBERT	**51.306**	**37.696**
BERT-MLM+USE	BERT-MLM	USE	49.43	36.39

*: For Milestone, we reported the results of a previous version of this model, where an augmented dataset was generated with Synonym, SBERT, and single mutation configuration. However, it was before we implemented `num-index-upper-bound`, which bounds the number of generated samples. The validation performance for that was F1: 49.412 and 35.34. The better performance is likely due to the larger number of generated samples.

4.4 Results

4.4.1 Phase 1

In Phase 1, we generated 4 different augmented datasets varying the token unmasking method and the semantic similarity scorer. We evaluated the models trained on these datasets on the validation dataset (Table 2).

Models trained on BERT-MLM-augmented data had better validation performance across the board, as expected given that BERT-MLM is a context-based language model while Synonym is a primitive word-based language model. The model trained on the BERT-MLM+SBERT augmented datasets performed best, so we used these configurations to move forward to Phase 2.

4.4.2 Phase 2

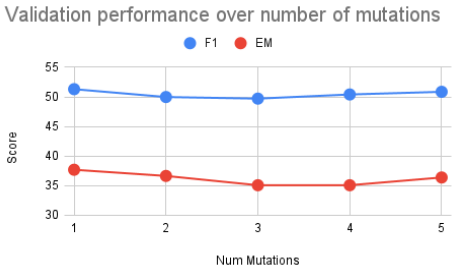
Phase 1 focused on picking configurations for the best semantic coherence. In Phase 2, we examined two axes that increased the level of perturbations in the training dataset: the number of mutations and the number of augmented samples generated per original sample. We wanted to see if the diversity of the out-of-domain samples increased model performance and if so to what extent. The results are summarized in Figure 3.

1 mutation worked best compared to multiple mutations which was a surprising result. We had hoped more mutations would allow the model to learn more general sentence structure, but we also understand that the inconsistency between the question and the context could have offset the benefit of more data.

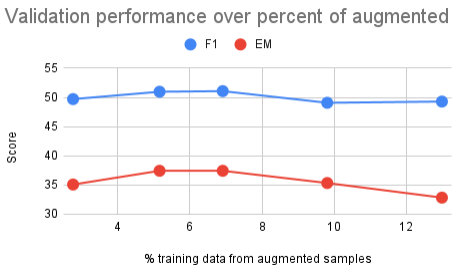
On the other hand, model performance peaked when 7% of the training data was augmented samples (`num-index-upper-bound` = 30). We anticipated this since we expected that the semantic suboptimality of the augmented samples has an impact on the model performance, outweighing the benefit of augmented samples.

4.5 Final model and test board

We decided to synthesize the outcomes of Phase 2 results and generate another augmented dataset with 1 mutation per sample and making up 7% of total training data. We call this model phase-2-



(a) Model validation performance when trained on the dataset with augmented out-of-domain samples with a given number of mutations from the original sample.



(b) Model validation performance when trained on a dataset with a given percentage of augmented out-of-domain samples and with each augmented sample having 3 mutations.

Figure 3: Validation performance of Phase 2 models..

Table 3: Test performance of final models.

Model Name	F_1	EM
BERT-MLM+SBERT (Phase 1 best)	**59.576**	**42.248**
phase-2-cumulative	59.044	40.872

cumulative, and it has a validation performance of **F1: 52.80, EM: 38.74**, the best of all the models so far, including Phase 1 models.

We decided to submit two models to the test leaderboard: the best model from Phase 1 (BERT-MLM+SBERT) and the cumulative model from Phase 2 (phase-2-cumulative). The test results are shown in Table 3.

Surprisingly, our Phase 1 model performed better than the Phase 2 cumulative model on the test set, even though Phase 2 performed better on the validation set. This means our hyperparameter tuning in Phase 2 was not making the model generalizable. If we reason backward from here, we can imagine that the increase in augmented samples may force the model to adapt to those particular sample domains but not necessarily all domains. Additionally, the increase in augmented samples in general means an increase in suboptimal synthetic samples where the question and context may be inconsistent, leading to poorer models.

5 Analysis

5.1 Phase 1 Qualitative Analysis

In Phase 1, we examined the model performance effect from using different token unmasking methods and semantic similarity scorer for data augmentation. BERT-MLM performed better as a token generator both quantitatively and qualitatively as expected and as shown in Figure 4.

We also examined the augmented examples generated by Synonym and BERT-MLM and observed that BERT-MLM, unsurprisingly, generated tokens that made more semantic sense even if it changed the meaning of the sentence. This is because BERT-MLM takes in the entire context as input whereas Synonym only uses the masked word as input.

The effect of semantic similarity scorer was more nuanced. When the token unmasking method was Synonym, the USE semantic similarity scoring configuration resulted in better model performance. When the token unmasking method was BERT-MLM, the SBERT semantic similarity scoring configuration performed better. We suspect this is because BERT-MLM and SBERT share the same BERT foundation, so they are more compatible together. This also demonstrates a deviation from the original BAE paper [1], which used BERT-MLM and USE-based semantic similarity to generate samples.

(i) Ex. 1: original sentence

Question: Why did Ray Eberle die?
Context paragraph: Ray Eberle died of a **heart attack** in Douglasville, Georgia on August 25, 1979, aged 60.
Answer: **heart attack**

(ii) Ex. 1: Synonym perturbed sentence (1 mutation)

Question: Why did Ray Eberle die?
Context paragraph: Ray Eberle died of a **heart blast** in Douglasville, Georgia on August 25, 1979, aged 60.
Answer: **heart blast**

(iii) Ex. 1: BERT-MLM perturbed sentence (1 mutation)

Question: Why did Ray Eberle die?
Context paragraph: Ray Eberle died of a **heart condition** in Douglasville, Georgia on August 25, 1979, aged 60.
Answer: **heart condition**

(iv) Ex. 2: original sentence

Question: What business published NHL FaceOff 2003?
Context paragraph: NHL FaceOff 2003 is an ice hockey video game made by SolWorks and published by **Sony Computer Entertainment** of America, released on the PlayStation 2.
Answer: **Sony Computer Entertainment**

(v) Ex. 2: Synonym perturbed sentence (1 mutation)

Question: What business published NHL FaceOff 2003?
Context paragraph: NHL FaceOff 2003 is an ice hockey_game video game made by SolWorks and published by **Sony Computer Entertainment** of America, released on the PlayStation 2.
Answer: **Sony Computer Entertainment**

(vi) Ex. 2: Synonym perturbed sentence (1 mutation)

Question: What business published NHL FaceOff 2003?
Context paragraph: NHL FaceOff 2003 is an ice dance video game made by SolWorks and published by **Sony Computer Entertainment** of America, released on the PlayStation 2.
Answer: **Sony Computer Entertainment**

Figure 4: 2 examples of original, Synonym, and BERT-MLM augmented samples. The target answer is in bold and blue, and the original and mutated words are both highlighted in yellow.

Question: What is the name of the chromosome where you can find C14orf159?
Context Paragraph: UUPF0317 protein C14orf159, mitochondrial is a protein that in humans is encoded by the C14orf159 gene (**chromosome 14** open reading frame 159).
True Answer: **chromosome 14**
Baseline Model Answer: mitochondrial is a protein that in humans is encoded by the C14orf159 gene (**chromosome 14**
BERT-MLM+SBERT Model Answer: **chromosome 14**

Figure 5: Example in which the baseline model failed to predict the exact answer, while our model trained on BERT-MLM+SBERT augmented data did predict the exact answer. The exact answer is bolded in blue.

5.2 Phase 2 Qualitative Analysis

The results were less conclusive in comparison to Phase 1, as highlighted in the Results section. Notably, 1 mutation performed quantitatively better than multiple mutations, which was unexpected. We examined several augmented samples with 3 mutations in Figure 6. While the generated tokens made semantic sense, the resulting context paragraph deviated more from the original meaning of the input. For example, in Ex. 3, a Flute sonata is no longer for flute. More significantly, in Ex. 4, the name of the subject of the question, Eadwulf, was changed, as well as their familial relationship to other persons in the context paragraph. A similar effect is observed in Ex. 5.

Two factors contributed to this meaning deviation. We limited the masked tokens to only nouns, and nouns are the most likely tokens to have an impact on the meaning of the context. Additionally, nouns are most likely to be referenced in the question and yet we are mutating only the context paragraph and not the corresponding question.

5.3 Performance improvement over baseline

Figure 5 shows an example where our models performed better than baseline. We hypothesize that since the `relation_extraction` training dataset contains several examples regarding genes and chromosomes, the reason for the BERT-MLM+SBERT model's success is the ability to perturb and expand these sentences into a wider variety to learn from when compared to the baseline.

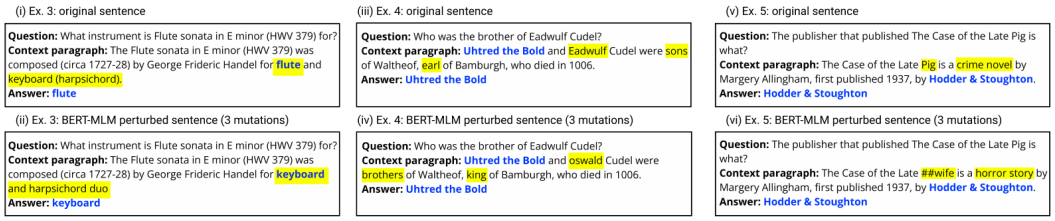


Figure 6: Original and augmented question-answer examples by BERT-MLM+SBERT augmentation, with 3 mutations in each example. The target answer is in blue and bold, and the original and mutated words are highlighted in yellow.

5.4 Limitations

We chose to forgo several implementations and axis of investigation in favor of running more complete experiments within the timeline.

One of the key limitations with our augmentation is that we only mutated the context paragraph, not the question. This means we can mutate key parts of the context paragraph without mutating the corresponding part in the question, leading to incomprehensible question-answer pairs, as seen in Example 4 and 5 in Figure 6.

We also deliberately chose to rank the tokens to mask by importance based on the baseline model predictions, but instead, we chose to only mask noun tokens. We would have needed to build out new input preprocessing in order to properly feed into the baseline model inference, and given the choice of the token is less important to the raw volume and semantic coherence of the augmented sentence, we felt our time is better served running more experiments.

While we implemented both REPLACE mutations and INSERT mutations, we chose to only investigate REPLACE mutations because we wanted to limit our axes of investigation and felt REPLACE, along with a number of mutations and `num-index-upper-bound`, was introducing enough perturbations.

We also want to note several shortcomings with our implementation of BERT-MLM inference. Our augmentation pipeline's tokenization on the input is inconsistent. First, we split the input to word-level tokens by NLTK's Whitespace Tokenizer so we can guarantee the masking of full words. However, the BERT tokenizer that we used for generating BERT-MLM input is a subword tokenizer. We can see the effect of the subword in Example 5 in Figure 6. Separately, we also limited masking to the first 200 words of the input, due to BERT-MLM's 512-token input length limit.

6 Conclusion

We implemented a data augmentation pipeline with multiple semantic and perturbation configuration parameters and successfully demonstrated that augmented data from this pipeline increases model performance on low-resource QA. We observed that models trained on BERT-MLM and SBERT-scorer augmented datasets performed best, which deviated from the original BAE paper that used BERT-MLM and USE-scorer. We also investigated how the level of perturbation in the training set (number of mutations per sample and `num-index-upper-bound`), and found that 1 mutation and 7% of training data being perturbed samples performed best, though we are less convinced by this result. We synthesized the outcomes of Phase 2 experiments and trained a new model that performed best out of all models of all Phases on the validation set. However, on the test set, the best model from Phase 1 actually performed better than the cumulative model Phase 2, which we want to investigate further.

In the future, we also want to resolve the implementation limitations, such as creating a sliding window approach on the input to BERT-MLM so token mutations can happen anywhere in the input text, rather than the first 200 words. Additionally, we want to investigate mutating both question and context together, so to create increasingly coherent QA samples.

References

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

[2] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in Data Science and Information Engineering*, pages 877–894, 2021.

[3] Xiang Li, Wei Zhang, Qian Ding, and Jian-Qiao Sun. Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal processing*, 157:180–197, 2019.

[4] Egoitz Laparra, Steven Bethard, and Timothy A Miller. Rethinking domain adaptation for machine learning over clinical language. *JAMIA open*, 3(2):146–150, 2020.

[5] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

[6] Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in nlp—a survey. *arXiv preprint arXiv:2006.00632*, 2020.

[7] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[8] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018.

[9] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[10] Sentence transformers documentation. `https://sbert.net`.

[11] How to use bert from the hugging face transformer library.