Soft Contextual Data Augmentation for Robust Question Answering

Stanford CS224N Default Project RobustQA Track

Kevin Tien
Department of Computer Science
Stanford University
kttien@stanford.edu

Megumi Sano
Department of Computer Science
Stanford University
megsano@stanford.edu

Toby Frager
Department of Computer Science
Stanford University
tobiasfr@stanford.edu

Abstract

Question answering using state-of-the-art Transformer models shows brittleness to domain transfer. One contributing factor is changes in phrasing patterns between different domains. We attempt to train models that are robust to changes in phrasing patterns using soft contextual data augmentation (SCA), a data augmentation strategy in which token embeddings are replaced by "soft token" embeddings, which are weighted averages of token embeddings based on a distribution over the vocabulary produced by a language model conditioned on the rest of the sequence. We use pre-trained DistilBERT in the Masked Language Model setting to generate distributions for soft tokens and then use pre-trained DistilBERT in the Question Answering setting for fine-tuning on the QA task. Our results demonstrate that SCA improves performance specifically on the out-of-domain QA task and analysis suggests SCA is particularly helpful for improving performance on question-context pairs that involve paraphrasing.

1 Key Information

- Mentor: Sarthak Kanodia
- External Collaborators (if you have any): None
- Sharing project: No

2 Introduction

Question answering (QA) in modern NLP is frequently formulated as a problem of reading a question string and context string and producing a span in the context string which contains the answer, or replying that no answer exists [1]. As questions and contexts for each dataset are curated from one or few sources and processed uniformly, there are unavoidable patterns characteristic to the distribution of each dataset. This contributes to a loss in performance when a model finetuned on one QA dataset is evaluated on a different dataset.

Since data in real world applications is rarely generated uniformly, it is important that QA systems be able to handle shifts in distribution from their initial training distribution. Current methods, such as finetuning large pretrained language models like DistilBERT on the QA task [2][3] drop off in accuracy dramatically when changing domains.

Data augmentation is a strategy used to improve the robustness of ML systems by altering training datasets automatically to increase their size. It has been used to great effect in NLP, computer vision, and other domains [4][5][6]. We investigate the effect of data augmentation on robustness of QA systems to domain shift. We hypothesize that certain patterns of phrasing related to word choice exist in a dataset, and that by augmenting data to be less dependent on specific word choice we can improve performance on out-of-domain data.

In particular, we investigate soft contextual data augmentation (SCA), which replaces selected tokens in text sequences with soft tokens, which are distributions over the vocabulary for the original positions of the selected tokens, predicted by a language model. By taking a linear combination of the vocabulary with weight corresponding to the distribution, SCA aims to capture a more broad meaning combining multiple appropriate words, as suggested in [7].

We investigate the effectiveness of this method by finetuning DistilBERT on the QA task on a set of datasets and investigating its performance on other, held-out datasets. Our results show that SCA provides a boost to model robustness with proper hyperparameters for sampling and choice of masking strategy. Close analysis of individual examples suggests that it is particularly effective when the question-context pair involves paraphrasing.

3 Related Work

3.1 Data Augmentation for Question Answering

Generalization to out-of-domain distributions in question answering without any knowledge of the target domain is difficult. Data augmentation aims to solve the general problem of encouraging QA models to learn representations that are not biased by any one domain through exploring different ways of curating the training data and learning regime.

There are many data augmentation techniques, each of which attempt to improve models' ability to generalize in a different way. Longpre et. al [4] introduces the following methods.

1. **Domain sampling:** When multiple datasets are available for training, it is advantageous to determine which ones contribute best to out-of-domain performance by training the model on each separately and measuring out-of domain performance. By altering the sampling distribution between datasets to draw more from those which improve out-of-domain performance, and less from those that decrease it, out-of-domain performance for the model trained on all domains is increased.
2. **Negative sampling:** Including an abstention option for the model and adding naturally occurring negative samples (examples where the answer span is not present) can lead to significant improvement.
3. **Back-translation:** By translating data first through a pivot language unrelated to the task, then back into the language in question, this technique aims to reduce dependence on exact phrasing by preserving only the meaning.
4. **Active learning:** By sampling the training data based on the difficulty of examples, calculated by $1 - F1$, this technique aims to encourage learning in difficult cases.

3.2 Soft Contextual Data Augmentation for Neural Machine Translation

Gao et al. introduce the idea of soft contextual data augmentation (SCA) as a targeted data augmentation strategy for neural machine translation (NMT) tasks (see Section 4 for more mathematical details on how SCA works; this section outlines the motivation, performance, and how it may be suitable to our project) [8]. This strategy addresses two problem areas that the authors identified:

1. Existing methods included random transformations such as swapping two words, which often resulted in significant changes in the semantics of a sentence [9].
2. Similar methods that attempted to replace other words using a language model were unable to leverage all of the possible candidates for achieving good performance since they only utilized one replacement word at a time [10].

To evaluate SCA, the authors tested its performance on various translation tasks and compared it to other baseline augmentation techniques such as swapping and contextual data augmentation using sampling. They found that SCA achieved a higher BLEU score than the baseline Transformer system and each of the other augmentation methods on all four translation tasks.

Notably, the results demonstrated SCA's ability to generalize to all of the NMT tasks regardless of the dataset, which suggests it could be valuable for improving a model's performance on the out-of-domain QA task that we are focusing on.

4 Approach

- Our main approach is applying SCA to the problem of out-of-domain QA [8] (see Section 3.2 for more details on the background of SCA). Specifically, SCA can be summarized as follows:

  For each sequence x in the dataset,

  1. Sample a token x_t uniformly at random.
  2. Replace the embedding of the token $E(x_t)$ with its soft token embedding $E_s(x_t)$ where

  $$E_s(x_t) = \sum_{j=0}^{|V|} p(w_j|x_{<t})E(w_j)$$

  where p is taken from the output of a language model given $x_{<t}$ (the sequence $x_1, ..., x_{t-1}$) as input.
  3. Repeat until the percentage of tokens to be sampled per sequence is reached.

- We made two modifications to SCA for our project (Fig 1):

  1. Because the language model above needs to be pretrained using the same tokenizer as the QA model, we decided to use DistilBERT for the language model instead, where the prediction for token w_j depends bidirectionally on the rest of the sequence. In particular, we use HuggingFace's `DistilBertForMaskedLM` pre-trained model. (We tried different masking strategies, but the one with best performance was where the token to be predicted is masked and the rest of the sequence is not. We explain this further in Section 5.4).
  2. When computing the soft token embedding, we truncate the probability distribution to the top k tokens in the vocabulary with the highest probabilities (and re-normalize). This is both more time and space efficient (averaging over less embeddings and storing less probability values) and empirically we find that increasing k does not necessarily improve performance.

  Given these modifications, the soft token embedding in our approach is

  $$E_s(x_t) = \frac{\sum_{j=0}^{k} p(w_j|x_{<t}, x_{>t})E(w_j)}{\sum_{j=0}^{k} p(w_j|x_{<t}, x_{>t})}$$

  where w_j is the token in V with the jth highest value of $p(w_j|x_{<t}, x_{>t})$ and p is provided by a pre-trained `DistilBertForMaskedLM` model.

  To provide more intuition on what a soft token is, here is an example from the training data we used: The original token is `dedicated`. The soft token is the following distribution:

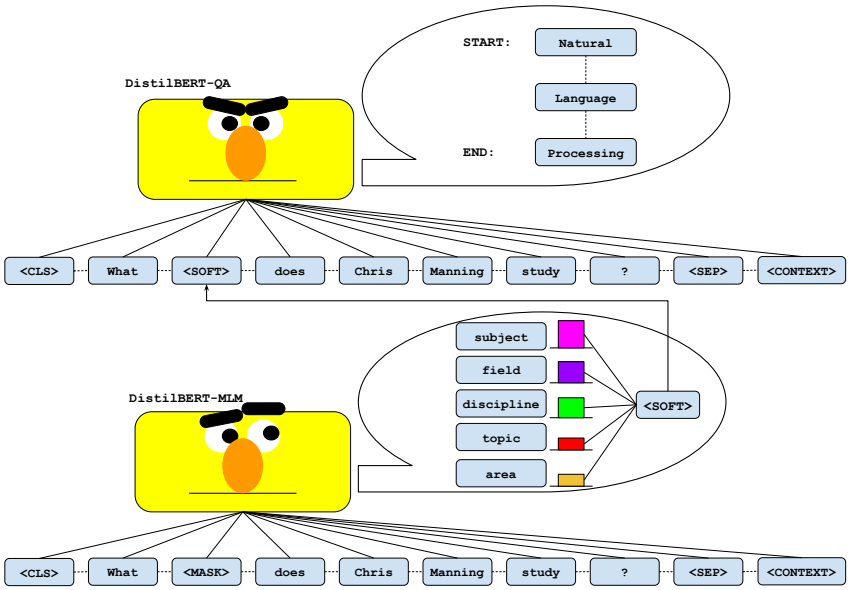
  - `dedicated` ($p = 0.705$)
  - `devoted` ($p = 0.199$)
  - `committed` ($p = 0.096$)
  - `open` ($p = 1e - 4$)
  - `oriented` ($p = 7e - 5$)

  and the soft token embedding is the weighted average of the embeddings of the above tokens.

  We implemented the modified SCA algorithm ourselves from scratch and did not rely on the original SCA paper's code. Our implementation can be found here: https://github.com/megsano/cs224n-robustqa, mainly in `train.py` and also in https://github.com/kttien/transformers, where we made sure DistilBERT can average over the embeddings during training.

- Our baseline is the baseline DistilBERT QA model provided by the class default project starter code, found here: https://github.com/michiyasunaga/robustqa.

Figure 1: Our method: Soft Contextual Data Augmentation with DistilBERT



5 Experiments

5.1 Data

We used the in-domain and out-of-domain datasets provided by the class. These datasets' examples come in the form of a context passage and a question, along with an answer that can be extracted as a span of text in the passage (if an answer exists in the context). In this report, we will not go into the details of the preprocessing (e.g., chunking) as the project handout already does.

The three in-domain datasets are Natural Questions, NewsQA, and SQuAD. Natural Questions and SQuAD are both based on Wikipedia, and NewsQA's examples come from news articles. The out-of-domain datasets are RelationExtraction, DuoRC, and RACE. RelationExtraction's contexts come from Wikipedia, but the questions are formulated based on relations between entities. DuoRC's examples come from movies, and RACE's examples come from reading comprehension exams. Each of the in-domain datasets is only used for train and dev sets, with 50,000 train samples and 4,000-13,000 dev samples each. The out-of-domain datasets are split into train, dev, and test sets, with a handful of samples being used for train and dev, and the vast majority being used for test.

5.2 Evaluation method

As in [4], we use F1 and Exact Match (EM) scores for our evaluation metrics. We use these metrics to evaluate the models' performance separately on in-domain datasets (ID) and out-of-domain datasets (OOD) so that we can track our approach's ability to make the model more robust towards OOD examples.

In addition, we perform qualitative analysis by examining examples where the outputs of our model and the baseline disagree (see Section 6 for details).

5.3 Experimental details

We finetuned our models for 3 epochs on NC6s v3 virtual machines on Microsoft Azure. Our standard learning rate was 3e-5 and training took about 3 hours across experiments. We used the

`DistilBERTForMaskedLM` model from HuggingFace for SCA and the `DistilBERTForQA` model from HuggingFace for QA.

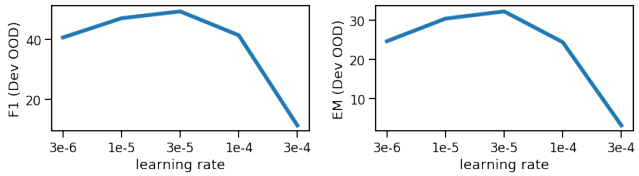
5.3.1 Experimental design

1. We ran experiments varying the % of tokens replaced with a soft token per sequence between 1%, 10%, 20%, and 40%.
2. We also experimented with different masking strategies: `masking together`, `masking separately` and `no mask`. In `masking separately`, for each sampled token, we create a copy of the input with just the sampled token replaced with the `MASK` token, and run DistilBERT separately to calculate each soft token. In `no mask`, we feed the input into DistilBERT directly, so it has access to the sampled token when making its prediction. In `masking together`, we replace all sampled tokens with the `MASK` token at once and generate the soft tokens in one pass.
3. We also ran experiments only augmenting the context and not the question. We present results for augmenting 10% and 20% of context tokens.
4. We also ran an experiment augmenting the input data at evaluation time in the same manner as in training time (motivations discussed more in Section 5.5.2).

5.3.2 Hyperparameters

For each hyperparameter of interest we ran a sweep of experiments perturbing that hyperparameter. In particular, we trained on a variety of learning rates scanning logarithmically centered at 3e-5, and found that 3e-5 performs best (Fig 2).

Figure 2: Performance across learning rate, (k = 5, replacement rate = 10%, `masking separately`)



We also explored $k = \{3, 5, 7\}$ and found that $k = 5$ performs best. The below numbers are with `masking separately` 10% of tokens and learning rate = 3e-5.

k **(number of top tokens)**	OOD **F1**	OOD **EM**
$k = 3$	47.93	31.68
$k = 5$	49.2	32.2
$k = 7$	46.8	30.89

5.3.3 Compute time

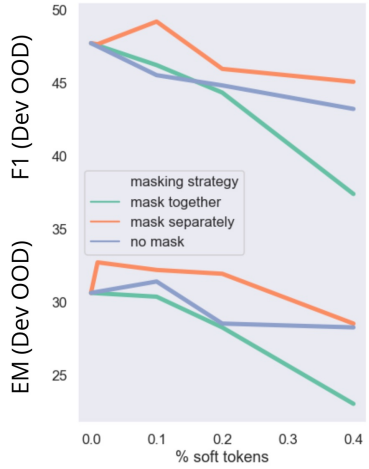
We found that the augmentation process took approximately 0.6 seconds per batch. For `masking together` and `no mask`, this corresponds to 128 input sequences, but for `masking separately` this corresponds to $\frac{128}{\textbf{\# replaced tokens}}$ input sequences (since we run DistilBERT separately for each replaced token, we need to scale down the batch size to fit on the GPU) and the compute time increases by a factor of # replaced tokens.

5.4 Results

While we discuss performance on the test dataset for our best performing models, the main results we discuss below are on the dev dataset since we only had three submissions to the test leaderboard. All results reported below are for the dev set unless specified otherwise.

5.4.1 Masking strategies and token replacement rate

The best performing augmentation strategy on the dev set was `masking separately` 10% of tokens to generate soft tokens, with improved performance on the OOD dev dataset in both F1 and EM scores



(a) Out-of-domain performance across masking strategies and varying % tokens replaced

F1	Squad	Nat Q	Newsqa	Race	DuoRC	RX
0%	77.77	69.47	57.72	36.65	39.53	66.84
1%	76.76	68.9	58.74	32.05	44.22	66.64
10%	76.3	68.65	57.66	38.71	41.56	67.21
20%	76.12	67.49	56.66	32.54	40.26	64.99
40%	74.55	65.92	55.26	33.45	37.03	64.62

(b) F1 performance across in-domain and out-of-domain datasets and varying % tokens replaced

EM	Squad	Nat Q	Newsqa	Race	DuoRC	RX
0%	63.22	52.55	39.34	21.88	30.16	39.84
1%	62.49	52.13	40.03	18.75	35.71	43.75
10%	61.71	51.59	39.74	23.44	30.16	42.97
20%	61.84	50.55	38.82	21.88	34.13	39.84
40%	59.46	48.9	37.82	17.97	29.37	38.28

(c) EM performance across in-domain and out-of-domain datasets and varying % tokens replaced

Figure 3: Main experimental results

(Figure 3a) (0%, no augmentation, indicates the baseline). This model achieved a test performance of **F1: 58.478, EM: 40.757** (see Section 5.5.1 for our best result on the test leaderboard). While this improvement demonstrates that SCA can be effective in the OOD QA task, it seems that it did not lead to as marked of an improvement as it did in the translations task in [8]. Perhaps this is due to the two tasks' inherent differences, as finding similar meanings for words is more directly pertinent to NMT, which must output words according to semantics, than it is to the QA task, which outputs a span of text, without much flexibility in phrasing or word choice. Nevertheless, our results show that SCA can lead to improved robustness and ability to generalize across various datasets.

We observed a curve in out-of-domain performance with respect to token replacement rate, with small values improving performance, but large values degrading it below baseline level. We believe that when much of the example is replaced by soft tokens, too much of the meaning is lost for the QA model to learn effectively.

A similar effect is observed in our experimentation with different masking strategies. The `masking together` strategy performs worse than `masking separately`; masking too many tokens at once could also cause our input to lose too much of its original meaning during the process of generating soft tokens using `DistilBERTMaskedLM`, leading to less accurate soft tokens, which could lead to less effective training and worse performance on the QA task.

Finally, the `no mask` strategy also performs worse than `masking separately`. This may be because it is not introducing enough new information to make up for the increased noise in the input.

5.5 Domain invariance and robustness

While our best performing models showed improvement on each of the OOD datasets, it led to a decrease in performance on each of the ID datasets (Figures 3b, 3c). Specifically, our experiment with using the `masking separately` strategy to replace 10% of tokens with soft tokens performs marginally worse on the ID datasets but noticeably better on the OOD datasets. In fact, ID performance decreases approximately linearly with token replacement rate. This suggests that SCA may not be increasing the model's performance on the QA task as a whole, but rather is helping the model learn domain invariant features, leading to improved robustness (generalization to OOD data).

5.5.1 Context-only augmentation

We ran experiments with `masking separately` 10% and 20%. With 10%, the model achieved **F1: 47.12, EM: 32.98** and with 20%, it achieved **F1: 47.28, EM: 30.1**. The 10% result has a slightly higher EM score than the results discussed above where both questions and contexts are masked.

Furthermore, interestingly, on the test dataset, the context-only version of our best model achieved the highest scores, performing slightly better on both F1 and EM with **F1: 59.275, EM: 40.803**, whereas masking both question and context achieved **F1: 58.478, EM: 40.757**. Overall, these results may suggest that masking tokens in the questions can be problematic for the QA task due to loss or alteration of its original semantics. Because the questions are usually much shorter than the contexts, replacing some of the tokens can completely change the meaning of the text.

5.5.2 Augmentation during evaluation time

We also tried applying SCA to evaluation data. While this is at first counter-intuitive since it degrades the quality of the input data, we hypothesized that this might bring the input distribution at evaluation time closer to the training distribution, and thus lead to better performance from the model. This model was trained with 10% replacement using `masking separately` (our best configuration), and achieved **F1: 43.55, EM: 29.58** on the dev set, which is several points worse than the corresponding model without evaluation-time augmentation. This may suggest that the decay in signal far outweighs the benefit from matching training and evaluation distributions.

6 Qualitative Analysis

Out of 382 examples in the dev set, there were 25 where our model predicted the exact answer and the baseline didn't, and 9 where the baseline predicted the exact answer and our model didn't. Upon manual inspection of these examples, we find that our model has a lower rate of **complete misses** in its answers and a higher success rate on context-question pairs where **paraphrasing** is present.

6.1 Lower rate of complete misses

Out of the 9 examples our model got wrong, 22% were complete misses (CM): incorrect answers which have no containment relation with the correct answer. Out of the 25 that the baseline got wrong, 44% were CM. This demonstrates that even when our model's answers are incorrect, it is still somewhat able to pick out the pertinent information to answer the question, relative to the baseline, which may reflect improved learning during training, that is not captured with the EM score.

Example of an answer that is not a CM produced by our model:

> **Context**: NKG2D is encoded by KLRK1 gene which is located in the NK-gene complex (NKC) situated on chromosome 6 in mice and chromosome 12 in humans.
>
> **Question**: What is the name of the chromosome where you can find NKG2D?
>
> **Correct answer**: Chromosome 12
>
> **Our model's answer**: Chromosome 6 in mice and chromosome 12

Example of a CM produced by the baseline:

> **Context**: ...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...
>
> **Question**: Which location offers the most direct view into daily life in the ancient world?
>
> **Correct answer**: Pompeii
>
> **Baseline model's answer**: Angkor Wat, Cambodia

6.2 Higher success rate for question-context pairs with paraphrasing

Out of the 9 examples our model got wrong, 22% had paraphrasing, where a word in the context near the answer span is a synonym to a word in the question and serves the same syntactic function. Out of the 25 that the baseline got wrong, 40% had paraphrasing. To investigate the role of SCA in this finding and to analyze effectiveness of our soft tokens, we fed the context into the pre-trained

DistilBERT we used in SCA to see if paraphrased words in the question were included in the truncated soft token distribution.

Example of a question-context pair with paraphrasing:

> **Context**: ...Griffin gives them the ArcNet and explains it can only work in zero gravity: K gets the idea to ***head*** to Cape Canaveral on 16th July 1969 (the day the Apollo 11 ship launched)...
>
> **Question**: Where must they ***go*** to attach the ArcNet?
>
> **Correct answer and our model's answer**: Cape Canaveral
>
> **Baseline's answer**: No k

In the example above, given the context with the token "head" masked, DistilBERT predicts: "return" (22.5%), "fly" (14.4%), "travel" (14.4%), **"go" (9%)**, "sail" (5.7%) for that token. Here we see that the paraphrased word in the question is actually included in the words comprising the soft token embedding that SCA generates for the model to train on. This may suggest that incorporating soft tokens can help the model better handle questions when paraphrased words are present.

Another example of a question-context pair with paraphrasing:

> **Context**: ...Dong-jin then binds Ryu and ***returns*** him to the riverbed where Yu-sun died...
>
> **Question**: Where does Dong-Jin ***take*** Ryu to?
>
> **Correct answer and our model's answer**: riverbed
>
> **Baseline's answer**: Yeong-mi's apartment building

Here the soft token (distribution outputted by DistilBERT given the context with the word "returns" masked) is "carries" (27.4%), "transports" (10.9%), "brings" (9.3%), **"takes" (5.2%)**, "leads" (4.9%). Again we see the paraphrased word (in conjugation) appear in the soft token. In both examples we see other synonyms (e.g. "fly", "travel", "transports", "brings") that would also be appropriate and others that could be depending on the question, suggesting that soft tokens are providing a broad set of meanings that could be associated with a token which prove helpful in the QA setting.

7 Conclusion

We investigated the application of SCA as a data augmentation strategy to an out-of-domain question answering task, and found that using SCA helps improve our model's performance on out-of-domain data. Specifically with masking 10% of tokens separately to generate soft tokens, our model outperforms the baseline in both F1 and EM scores on the out-of-domain datasets. Notably, our model performs slightly worse than the baseline on the in-domain datasets, which suggests that the improvement in performance on the out-of-domain datasets is due to SCA improving the model's robustness and ability to generalize rather than improving the model performance overall, which could be considered a limitation. Our qualitative comparison of the examples where the answers differed between the baseline and our model suggests that SCA can be particularly effective when paraphrasing is involved, which is in line with our expectations given the nature of the algorithm.

There are a few avenues for future directions that could build upon our approach. One idea is to look into combining SCA with other data augmentation techniques. In particular, the sampling techniques we encountered during our literature review could pair well with SCA since they focus on a dimension separate from the semantic dimension that SCA modifies in the data. For example, we could use active learning to upsample training examples that our model does poorly on.

Another possible set of experiments would be to be more deliberate with choosing which tokens to mask and augment with SCA. In our current implementation, we sample a proportion of tokens in the example at random to generate soft embeddings for. However, we could adjust our approach to sample only certain words that might be more likely to have more synonyms. For example, sampling more nouns might be more effective than sampling articles or helper verbs, which do not have many plausible synonyms.

References

[1] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] 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.

[4] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. An exploration of data augmentation and sampling techniques for domain-agnostic question answering. *arXiv preprint arXiv:1912.02145*, 2019.

[5] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621, 2017.

[6] Qingsong Wen, Liang Sun, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *CoRR*, abs/2002.12478, 2020.

[7] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *CoRR*, abs/1601.03764, 2016.

[8] Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, Florence, Italy, July 2019. Association for Computational Linguistics.

[9] Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. Data noising as smoothing in neural network language models. *arXiv preprint arXiv:1703.02573*, 2017.

[10] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018.