Exploring Data Augmentation Techniques for Text Classification

Anonymous ACL submission

Abstract

We investigate the use of data augmentation (DA) in improving the accuracy of text classification tasks. Using 6 datasets with different numbers of classification classes, and FastAI's implementation of ULMFiT as our model, we test the use of 8 augmentation techniques in isolation and in combination, and investigate its effects on classification accuracy change. We find that in terms of performance, DA increases accuracy, but no clear technique is superior.

1 Introduction

Often, large and high quality datasets are necessary to train robust models in NLP, which are costly to compile and acquire. We saw this issue when completing the text classification task in Assignment 1, where we identified limited data as a potential limiting factor to accuracy increase. Then, seeing Zhang's brief exploration of synonym replacement as a data augmentation (DA) technique in Reading Assignment 1 (Zhang et al., 2016), we grew interested in exploring text DA techniques as a method of increasing data size and accuracy. DA's widespread use in computer vision as a way to reduce model over-fitting further strengthened our motivation to explore DA for text.

Studies (discussed in the next section) have showed the success of various DA techniques in isolation. We build upon their findings by aiming to answer the following questions: 1. Does a DA technique's approach in changing data (eg. semantic vs. syntactic change) effect its success? 2. Do certain DA techniques work better on some types of datasets than others? 3. How much augmentation is too much? (ie. can augmenting augmented data bring improvement?) Also, as the techniques in isolation have been successful, we hypothesize that as they would further diversify the change in a dataset, individual DA techniques could be combined together to achiever even higher accuracy. We proceed with a brief introduction of related works which motivated choices in our experiments. We then explain our experiments, give our results, and end with a discussion and conclusion.

2 Related Works

A previous study by (Feng et al., 2021) presented a comprehensive survey of key DA methods used in NLP. We expand by selecting methods presented by Feng which can be easily applied for the text classification task, and experimenting exclusively on them. For instance, one technique we explore is Easy Data Augmentation techniques (Wei and Zou, 2019), or EDA, which combines random insertion, random deletion, random swap and synonym replacement for DA, and show that EDA performs significantly well on small datasets. Several studies also concluded that several DA techniques like EDA do not work as well for pretrained models like ULMFiT Shleifer 2019; Wei and Zou 2019. Regardless, we run all DA techniques to get a fair comparison of all the techniques, and will state if results contradict previous paper's findings. Other techniques presented by Feng include Contextual Word Embedding (Kobayashi, 2018), where words in the training data are replaced by words predicted by a language model, as well as adding synthetic noise (Feng et al., 2020), which involves inserting, removing, or swapping characters for each word.

In class (lecture 20), backtranslation (Sennrich et al., 2016) was introduced to us, where an augmented dataset is produced by translating texts into another language and back with neural machine translation models. This technique has already been applied on the ULMFiT model (Howard and Ruder, 2018) for sentiment analysis on the IMDb dataset (Shleifer, 2019).

However, to the best of our knowledge, the use of these DA techniques in combination have not been thoroughly explored. With that, our project aims to uncover the unexplored interactions between the proposed methods when used individually (in isolation), in combination, and different text classification tasks of our 6 datasets.

3 Methods

Here we present the several DA methods that we used in our experiments.

3.1 Individual Methods

Backtranslation (Sennrich et al., 2016): A sentence is translated into a target language then back to its source language. We used Facebook's WMT19 model (Ng et al., 2019) for this.

Contextual Word Embeddings: Use the BERT-base-uncased model (Kobayashi, 2018) to apply word level operations to a sentence based on contextual word embedding and find the top n similar words for augmentation.

Random Swap (Wei and Zou, 2019): Randomly choose two words in a sentence and swap their positions. Do this n times.

Random Insertion (Wei and Zou, 2019): Find a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence. Do this n times.

Random Deletion (Wei and Zou, 2019): Randomly remove each word in the sentence with probability p.

Synonym Replacement (Wei and Zou, 2019): Randomly choose n words from the sentence that are not stop words. Replace each of these words with one of its synonyms chosen at random.

Synthetic Noise (Wei and Zou, 2019): For every word, at every character, we either insert, delete, or swap a character (equal chance of choice). We fix the addition of 50% noise per data entry and ignore the first and last character of each word to imitate typos (Feng et al., 2020).

EDA (Wei and Zou, 2019): Randomly select and perform one of synthetic noise, random insertion, synonym replacement and random swap.

EDA (None): A variation of EDA where we include the option of doing no augmentation in the selection.

3.2 Combined Methods

We define a combination as either 1. Doing multiple different types of DA on the same text separately, or 2. Doing DA on top of the augmented data.

Double EDA: A modified version of EDA where we randomly select two operations to perform instead of one.

Double EDA (None): Similar to Double EDA but we select from synonym replacement, backtranslation, random swap, random insertion, no augmentation.

Noise + EDA: Perform both synthetic noise and EDA on the data.

Noise + Translation: Perform both synthetic noise and backtranslation on the data.

EDAx2: Perform EDA again on data augmented with EDA.

Translated EDA: Perform EDA on data augmented with backtranslation.

Double Synonym: First perform synonym replacement and then perform contextual word embedding on top of the augmented data.

Translated Synonym: Perform synonym replacement, backtranslation and contextual word embedding in the following order.

Noisy EDA: Perform synthetic noise on data augmented with EDA.

Embedded EDA: Perform contextual embedding on data augmented with EDA

Similar to the EDA study (Wei and Zou, 2019), we vary the parameter α. The number of words to change for synonym replacement, random insertion and random swap n, is dependent on $n = \alpha L$, where L is sentence length. For synthetic noise and random deletion, we take probability $p = \alpha$.

4 Experimental Setup

4.1 Model

Since our focus is DA, we choose the ULMFiT model (Howard and Ruder, 2018) for our experiments because of the model's relatively high performance. For ease of implementation, we use the implementation of ULMFiT done by Fast.ai in their text tutorial, which makes use of a language model pretrained on WikiText-103 and an ASGD Weight-Dropped LSTM classifier (Fast.aiTeam, 2021). However, we fine-tune the initial language model over 5 epochs instead of Fast.ai's 10, due to computational constraints.

4.2 Datasets

We evaluate on 1 Pinyin and 5 English widely used text classification datasets obtained from the torchtext library (TorchContributors, 2017), which include sentiment analysis and topic classification tasks. Our sentiment analysis datasets include the IMDb (movie reviews), YelpReviewPolarity and AmazonReviewPolarity, which all have binary labels (positive or negative). Our topic classification datasets include SogouNews (5 labels), AG_News (4 labels), the Yahoo Answers (10 labels) dataset compiled by (Zhang et al., 2016). SogouNews is the Chinese dataset, with data entries converted from Chinese characters to pinyin form.

Furthermore, a previous study using UMLFiT and EDA methods on the IMDb dataset showed that the improvements due to DA on full datasets are minimal (Shleifer, 2019). This, along with limited computational resources, leads us to reduce our dataset sizes. We do so by randomly sampling from the full datasets to form datasets of the following sizes: 500 - ’small’, 2000 - ’medium’, 5000 - ’large’.

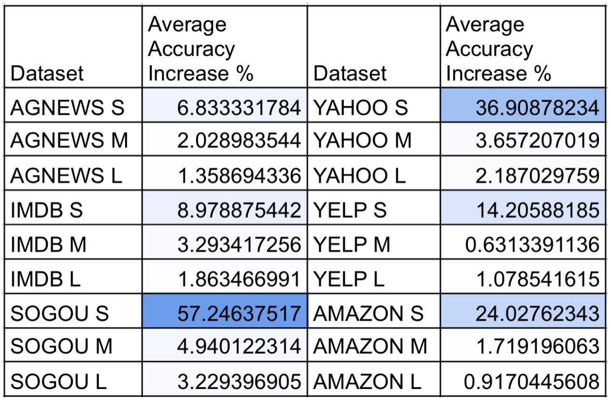
4.3 Procedure

We use a 80-20% train-test split for all of our experiments. First, we run ULMFiT on all the original datasets, using these results as our comparison benchmark. Then, we run individual DA methods with an augmentation probability $p = 2$ (chosen randomly) on each dataset and run ULMFiT on the combination of augmented and original data. After, we attempt to find an ideal augmentation probability p by iterating through different p values on small datasets. Lastly, we run combinations of several DA methods only on the small datasets (due to limited computational resources) in an attempt to look for an ideal DA combination.

5 Results

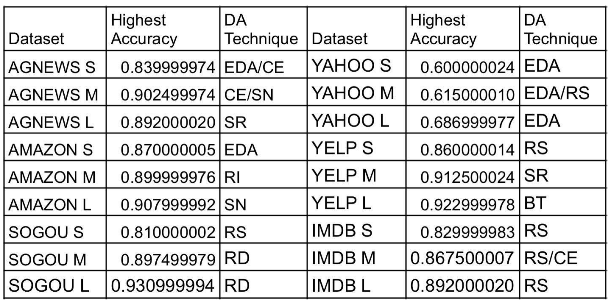
5.1 DA Methods in Isolation

As shown in Figure 1, the increase in accuracy on the large datasets is minimal, regardless of the DA techniques applied. Indeed, the average accuracy increase does not exceed 3.3% on any large dataset with any DA technique. On the small datasets, however, the performance with augmentation grows significant. For example, random swap increases the performance on the small Sogou dataset by more than 76%. Note that since the Sogou dataset is in the Chinese language transcribed in Pinyin, DA techniques making use of English languages models (like synonym replacement and backtranslation) are not applicable to the data. Moreover, while the contextual word embeddings technique uses the BERT-base-uncased model for the English model, it works surprisingly well for the SogouNews dataset; therefore, its results are retained in the table.



Dataset	Average Accuracy Increase %	Dataset	Average Accuracy Increase %
AGNEWS S	6.833331784	YAHOO S	36.90878234
AGNEWS M	2.028983544	YAHOO M	3.657207019
AGNEWS L	1.358694336	YAHOO L	2.187029759
IMDB S	8.978875442	YELP S	14.20588185
IMDB M	3.293417256	YELP M	0.6313391136
IMDB L	1.863466991	YELP L	1.078541615
SOGOU S	57.24637517	AMAZON S	24.02762343
SOGOU M	4.940122314	AMAZON M	1.719196063
SOGOU L	3.229396905	AMAZON L	0.9170445608

Figure 1: Average accuracy increase percentage when each DA technique is done individually per dataset.



Dataset	Highest Accuracy	DA Technique	Dataset	Highest Accuracy	DA Technique
AGNEWS S	0.839999974	EDA/CE	YAHOO S	0.600000024	EDA
AGNEWS M	0.902499974	CE/SN	YAHOO M	0.615000010	EDA/RS
AGNEWS L	0.892000020	SR	YAHOO L	0.686999977	EDA
AMAZON S	0.870000005	EDA	YELP S	0.860000014	RS
AMAZON M	0.899999976	RI	YELP M	0.912500024	SR
AMAZON L	0.907999992	SN	YELP L	0.922999978	BT
SOGOU S	0.810000002	RS	IMDB S	0.829999983	RS
SOGOU M	0.897499979	RD	IMDB M	0.867500007	RS/CE
SOGOU L	0.930999994	RD	IMDB L	0.892000020	RS

Figure 2: Highest Accuracies per dataset, with the DA used to reach that accuracy, when augmented with DA’s in isolation. CE: Contextual Embeddings; SN: Synthetic Noise; SR: Synonym Replacement; RI: Random Insertion; RS: Random Swap; RD: Random Deletion; BT: Back-translation

Figure 2 shows the highest accuracy reached and the DA technique used to augment the dataset. Compared to other techniques, random swap and EDA resulted in the highest accuracies most frequently, showing the highest results 6 and 5 times out of 18 datasets respectively. Figure 1 and Figure 2 contain information derived from “main_exp.jpg” table in our GitHub, which includes results of all individual DA techniques.

5.2 Tuning the Probability Parameter

We attempt to tune the probability parameter of techniques that require one (ie. random swap, random insertion, random deletion, synonym replacement, and EDA). As shown in Figure 3, the re-

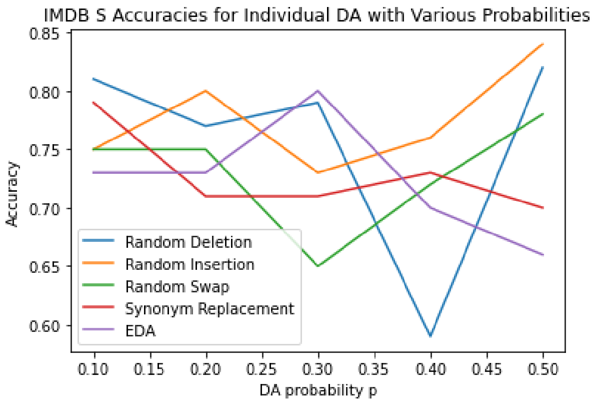


Figure 3: Accuracy of the IMDB S dataset when augmented with individual DA techniques using different probability hyper-parameter values.

sults of using DA with different probabilities on the IMDB S dataset indicate that there is no single optimal value. Figure 4 shows results we obtained for the other datasets. It shows that no optimal probability value that applies universally exists (ie. optimality is dataset specific), but that a probability of 0.2 for EDA on the small datasets is consistently optimal. The latter notably testifies against the original EDA paper (Wei and Zou, 2019), which suggests the value of 0.05 for a dataset size of 500 (same size as our ”small datasets”) as optimal.

5.3 DA Methods in Combination

Due to time and computational resource constraints, we experiment with the combination of multiple DA methods only on the small datasets. We show our results in Figure 5. We observe that, although combining different DA techniques can further increase benchmark accuracies (highest accuracies from when DAs were used in isolation), this is not always the case. Indeed, combining DA only results in an increase in performance in 3 out of the 5 data sets. Moreover, the results do not show the superiority of any certain combination.

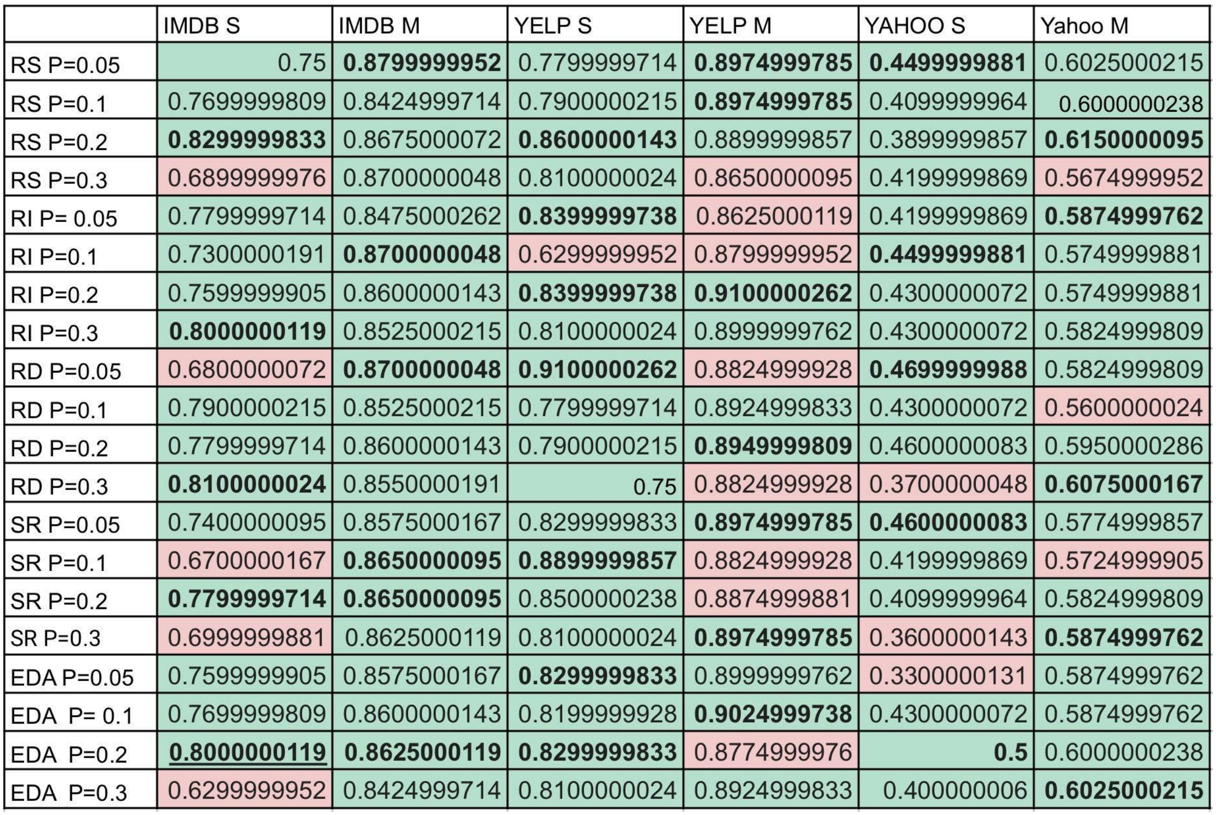
6 Discussion

Our results confirm the claims from previous papers that DA techniques work the best on small datasets. If we use the magnitude of accuracy increase as a metric, our findings do not indicate the absolute superiority of any DA technique when applied individually. However, we did identify EDA and Random Swap to have the highest accuracy improvement most frequently. On top of this, if we consider the pragmatism of each DA technique, EDA and Random Swap are the least computationally expensive, easiest to implement, and therefore can be worth considering over other DA techniques.

In relation to this, we have not found any clear pattern of consistent accuracy change that would indicate a relationship between certain DA techniques and dataset types: all techniques are able to enhance the performance on most tasks. We observe however, that contextual word embedding does not perform well on the Yahoo dataset. This may be due to the higher possibility of typos in the Yahoo dataset, which may make contextual word prediction less accurate. On the other hand, a surprising result is the good performance of contextual word embeddings with the BERT-base-uncased English language model on the Pinyin (Chinese) dataset Sogou News. In principal, since the contextual word embeddings method replaces words with other words that could have been at its position as predicted by the language model, an English language model would not be able to produce sensible words for Chinese. One possible explanation is that the BERT language model captures some aspects of language that are not unique to English.

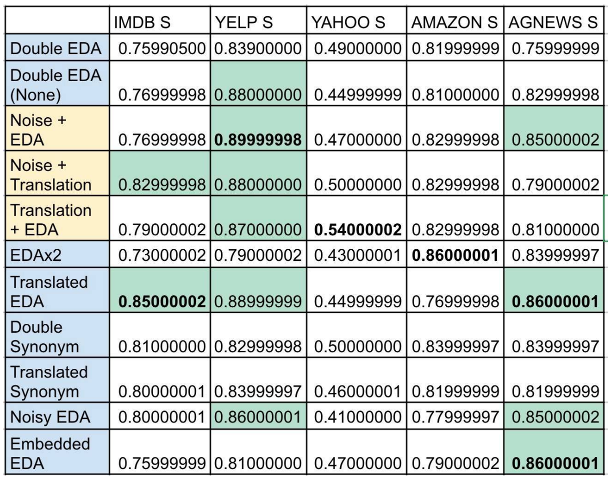
We also note a finding that does not agree with the claim in the original EDA paper: while the EDA paper claims that the technique does not work very well for pre-trained models (Wei and Zou 2019; Shleifer 2019), we notice a significant increase in performance with ULMFit when EDA is applied on the small datasets.

Moreover, the experiments on DA technique combinations reveal that combining DA techniques sometimes helps, but does not identify an optimal combination of DA techniques. The fact that more augmented training data does not necessarily mean increased model performance is interesting. We can consider the experiments of Double EDA and Double EDA (None): the former includes more instances of augmented data, yet the latter generally performs better. This may indicate that there is a threshold to the augmentation of training data, which when passed, renders adding more augmentation counterproductive. We also note that while augmenting the training data with more than one DA technique and concatenating the results (forming a bigger augmented dataset) often helps with performance, augmenting augmented data does not. This may be due to the fact that augmenting the same data instance multiple times results in an instance that deviate too much from the original data



	IMDB S	IMDB M	YELP S	YELP M	YAHOO S	Yahoo M
RS P=0.05	0.75	**0.8799999952**	0.7799999714	**0.8974999785**	**0.4499999881**	0.6025000215
RS P=0.1	0.7699999809	0.8424999714	0.7900000215	**0.8974999785**	0.4099999964	0.6000000238
RS P=0.2	**0.8299999833**	0.8675000072	**0.8600000143**	0.8899999857	0.3899999857	**0.6150000095**
RS P=0.3	0.6899999976	0.8700000048	0.8100000024	0.8650000095	0.4199999869	0.5674999952
RI P= 0.05	0.7799999714	0.8475000262	**0.8399999738**	0.8625000119	0.4199999869	**0.5874999762**
RI P=0.1	0.7300000191	**0.8700000048**	0.6299999952	0.8799999952	**0.4499999881**	0.5749999881
RI P=0.2	0.7599999905	0.8600000143	**0.8399999738**	**0.9100000262**	0.4300000072	0.5749999881
RI P=0.3	**0.8000000119**	0.8525000215	0.8100000024	0.8999999762	0.4300000072	0.5824999809
RD P=0.05	0.6800000072	**0.8700000048**	**0.9100000262**	0.8824999928	**0.4699999988**	0.5824999809
RD P=0.1	0.7900000215	0.8525000215	0.7799999714	0.8924999833	0.4300000072	0.5600000024
RD P=0.2	0.7799999714	0.8600000143	0.7900000215	**0.8949999809**	0.4600000083	0.5950000286
RD P=0.3	**0.8100000024**	0.8550000191	0.75	0.8824999928	0.3700000048	**0.6075000167**
SR P=0.05	0.7400000095	0.8575000167	0.8299999833	**0.8974999785**	**0.4600000083**	0.5774999857
SR P=0.1	0.6700000167	**0.8650000095**	**0.8899999857**	0.8824999928	0.4199999869	0.5724999905
SR P=0.2	**0.7799999714**	**0.8650000095**	0.8500000238	0.8874999881	0.4099999964	0.5824999809
SR P=0.3	0.6999999881	0.8625000119	0.8100000024	**0.8974999785**	0.3600000143	**0.5874999762**
EDA P=0.05	0.7599999905	0.8575000167	**0.8299999833**	0.8999999762	0.3300000131	0.5874999762
EDA P= 0.1	0.7699999809	0.8600000143	0.8199999928	**0.9024999738**	0.4300000072	0.5874999762
EDA P=0.2	**0.8000000119**	**0.8625000119**	**0.8299999833**	0.8774999976	**0.5**	0.6000000238
EDA P=0.3	0.6299999952	0.8424999714	0.8100000024	0.8924999833	0.400000006	**0.6025000215**

Figure 4: Accuracy after tuning the probability parameter. RS: Random Swap; RI: Random Insertion; RD: Random Deletion; SR: Synonym Replacement



	IMDB S	YELP S	YAHOO S	AMAZON S	AGNEWS S
Double EDA	0.75990500	0.83900000	0.49000000	0.81999999	0.75999999
Double EDA (None)	0.76999998	0.88000000	0.44999999	0.81000000	0.82999998
Noise + EDA	0.76999998	**0.89999998**	0.47000000	0.82999998	0.85000002
Noise + Translation	0.82999998	0.88000000	0.50000000	0.82999998	0.79000002
Translation + EDA	0.79000002	0.87000000	**0.54000002**	0.82999998	0.81000000
EDAx2	0.73000002	0.79000002	0.43000001	**0.86000001**	0.83999997
Translated EDA	**0.85000002**	0.88999999	0.44999999	0.76999998	**0.86000001**
Double Synonym	0.81000000	0.82999998	0.50000000	0.83999997	0.83999997
Translated Synonym	0.80000001	0.83999997	0.46000001	0.81999999	0.81999999
Noisy EDA	0.80000001	0.86000001	0.41000000	0.77999997	0.85000002
Embedded EDA	0.75999999	0.81000000	0.47000000	0.79000002	**0.86000001**

Figure 5: Accuracy of combined DA techniques. Bold font indicates optimum among the batch, white indicates increase from baseline (no augmentation), and green indicates increase from previous benchmark (highest among individual DA experiments). Blue cells are means a technique augments an already augmented dataset, while yellow augments initial dataset twice using different techniques, and combines them.

entry; thus, distorting or losing the original meaning of the data instance (i.e. sentence or text).

7 Conclusion

In this project, multiple textual DA techniques and combinations of them have been explored. We have confirmed findings from previous work that DA provides the most benefit to small datasets. We then found that combining multiple DA techniques can sometimes yield better performance than using one single technique. However we expect the increase to be minimal (3% in our results). If time and computational resources are limited, we recommend EDA or random swap to be prioritised over combinations. Further work may be done with testing textual DA with other models such as character level CNNs, as well as on smaller and unbalanced datasets to test for further generalization.

8 Statement of Contribution

All of us were equally involved in creating this write-up, implementing DA techniques, and running the experiments.

References

Fast.aiTeam. 2021. Transfer learning in text tutorial.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. GenAug: Data augmentation for finetuning text generators. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42, Online. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Sam Shleifer. 2019. Low resource text classification with ulmfit and backtranslation.

TorchContributors. 2017. Torchtext datasets.

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level convolutional networks for text classification.