Roles of Words: What Should (n't) Be Augmented in Text Augmentation on Text Classification Tasks?

Anonymous ACL submission

Abstract

Text augmentation techniques are widely used in text classification problems to improve the performance of classifiers, especially in low-resource scenarios. Previous text-editing-based methods augment the text in a *non-selective* manner: the words in the text are treated without difference during augmentation, which may result in unsatisfactory augmented samples. In this work, we present four kinds of *roles of words* (ROWs) which have different functions in text classification tasks, and design effective methods to automatically extract these ROWs based on *statistical* and *semantic* perspectives. Systematic experiments are conducted on what ROWs should (n't) be augmented during augmentation for classification tasks. Based on these experiments, we discover some interesting and instructive potential patterns that certain ROWs are especially suitable or unsuitable for certain augmentation operations. Guided by these patterns, we propose a set of *Selective Text Augmentation* (STA) operations, which significantly outperform traditional methods and show outstanding generalization performance.

1 Introduction

Text classification is one of the fundamental tasks in Natural Language Processing (NLP), which has wide applications in news filtering, paper categorization, sentiment analysis and so on. Plenty of algorithms, especially deep learning models, have achieved great success in text classification, such as recurrent neural networks (RNN) (Liu et al., 2016; Wang et al., 2018), convolutional networks (CNN) (Kim, 2014) and BERT (Devlin et al., 2019). The success of deep learning is usually built on the large training data with good quality, which is often difficult to obtain in real applications. Therefore, text augmentation techniques have attracted more and more attention both in academic and industrial communities and plenty of methods have been proposed to improve the generalization ability of text classification models when training data is limited, such

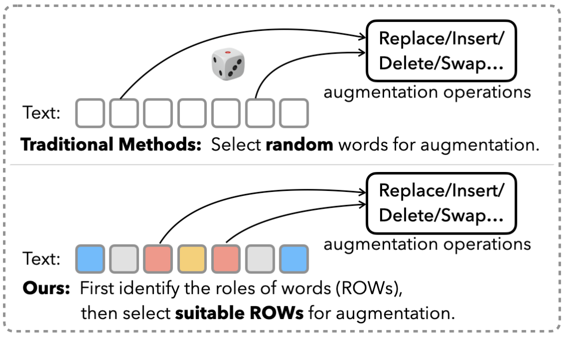

Figure 1: Comparison of our proposed method and traditional TE-based augmentation method. The different colors in the example represent different roles of words.

as synonyms replacement or insertion (Kolomiyets et al., 2011; Zhang et al., 2015; Wang and Yang, 2015a; Wei and Zou, 2019), random word deletion (Xie et al., 2017; Wei and Zou, 2019), back-translation (Yu et al., 2018; Silfverberg et al., 2017) and contextual augmentation (Kobayashi, 2018a).

Among these methods, text-editing(TE)-based augmentation techniques, including word replacement, deletion, insertion and swap, are widely used in industry and academy (Bayer et al., 2021) due to their simplicity and effectiveness (Wei and Zou, 2019; Feng et al., 2020). Thereby, in this work, we mainly focus on these TE-based augmentation methods. Previous works of TE-based methods are usually in a non-selective manner: augmentation is applied to all words (sometimes may exclude the stop-words) in the given text without difference. However, different words have different impact to the down-stream tasks. If we simply apply these augmentation operations on random words, we are likely to encounter some unsatisfactory situations where the augmented samples bring little performance gain or even hurt the classification performance:

1. Important class-indicating words may be altered, resulting in some damage to the original meaning or even changing the label of the original text;

2. Noisy or misleading words may be introduced after augmentation, which may hurt the generalization ability.

Therefore, we begin to think this question: How to selectively augment the text to avoid these bad situations and generate a better augmented training set for stronger generalization ability?

In this work, we first explore what are the different roles of words in text classification tasks through analysing some real cases. Based on the analysis, we conclude four types of **roles of words (ROWs)**: *Common Class-indicating words* (CC-words), *Specific Class-indicating words* (SC-words), *Intermediate Class-indicating words* (IC-words) and *Class-irrelevant words/Other words* (O-words). We then design effective methods to automatically extract these roles. Based on these roles, we conduct extensive experiments to investigate what ROWs should or shouldn't be augmented for commonly used TE-based augmentation techniques (delete, insert, replace and swap). Based on these experiments, we discover some interesting patterns for each augmentation operation, and then summarize a set of Selective Text Augmentation (STA) techniques which outperform traditional non-selective augmentation methods in a large margin. An illustration of our proposed augmentation methods compared with traditional methods is shown in Figure 1.

We conclude our contributions as follows:

- We for the first time present four types of **roles of words (ROWs)** for text classification tasks, and design effective methods to automatically extract these ROWs based on *statistical* and *semantic* perspectives, which are important for understanding the behaviors of classifiers and can also inspire related research;
- We systematically investigate what ROWs should (n't) be augmented for text augmentation, and discover inspiring instructive patterns for guiding us on how to select suitable words for text augmentation, through comprehensive experiments on 9 benchmark datasets;
- We propose a set of **STA** (**S**elective **T**ext **A**ugmentation) methods, which significantly outperform traditional non-selective text-editing based augmentation methods, both in single-dataset and cross-dataset evaluation tasks.

2 Related Work

According to how the augmented samples are generated, existing techniques of text augmentation can be categorized into three groups: **Text-editing(TE)-based augmentation**, such as token/phrase deletion (Xie et al., 2017), insertion (Wei and Zou, 2019), replacement (or substitution) (Kolomiyets et al., 2011; Zhang et al., 2015; Wang and Yang, 2015a) and swapping (Wei and Zou, 2019). **Text-generation(TG)-based augmentation**, like back-translation (Xie et al., 2019; Yu et al., 2018; Silfverberg et al., 2017), sentences synthesizing (Anaby-Tavor et al., 2020) and language modeling-based approaches (Jiao et al., 2019; Kobayashi, 2018b,a). **Feature space augmentation**, such as utilizing Mixup (Zhang et al., 2018) for sentence embeddings (Guo et al., 2019; Sun et al., 2020). Apart from these three types of text augmentation techniques, many other creative methods are proposed such as compositional augmentation (Jia and Liang, 2016; Andreas, 2019) and adversarial text augmentation (Morris et al., 2020).

Due to the dependence of large deep learning models or complex training process, TG-based or feature space augmentation are relatively inconvenient to implement, especially for those non-experts in NLP. Instead, TE-based augmentation methods are much easier to implement without using large models or altering the training process and are also proved to be effective for limited datasets (Wang and Yang, 2015b; Wei and Zou, 2019). Therefore, TE-based augmentations are quite popular in research and industry (Bayer et al., 2021). However, previous TE-based methods may get unsatisfactory augmented samples because of randomness during words selection for augmentation (Bayer et al., 2021; Chen et al., 2021).

In this work, we mainly focus on TE-based augmentation methods and study how to generate better augmented samples with simple text-editing operations.

3 Roles of Words in Text Classification Tasks

3.1 Categorization of Roles of Words

To explore the roles of different words and what have been learned by a text classifier, we conduct some exploratory case studies. We choose a small

No.	sentence	prediction
1	"basketball" / "athletes"	sport (✔)
2	"Based on" / "team"	sport (?)
3	"Schools should invest more in teachers"	education (✔)
4	"Schools should invest more in the teaching **team**"	sport (✗)
5	"Shanghai Bilibili hit a **three-pointer** in the last minute and won the final victory!"	computer (✗)

Table 1: Case study: Some hand-crafted examples to evaluate a trained BERT-based classifier which gets high accuracy on the original test set. ✔/ ✗/ ?: correct/ wrong/ confusing prediction.

dataset from the FD News[1] which contains four classes: "politics", "sport", "education" and "computer". Then we train a BERT-based classifier on this small dataset and the model obtains a test accuracy at 98.92%, which means the model already performs quite well in this dataset. We then use some hand-crafted sentences to test its performance as shown in Table 1. Inputting the word "basketball" or "athletes" to the model will directly get correct prediction, since they are common words related to "sport" class. However, examples in No.2 and 4 tell us the trained model is not as good as we thought: simply passing phrases like "based on" or "team" to the model will get prediction of "sport" class, even if sentence 4 should belong to "education" class. After checking the training set, we find that phrases like "based on" and "team" are highly correlated with the "sport" class in the training set, perhaps due to the bias during the dataset collection. The last example shows an case where the model cannot recognize a sport-related phrase "three-pointer" that seldom appears in the training set.

Obviously, different words in this Table 1 play different roles in this classification task. The words/phrases like the "based on" and "team" are those co-occur frequently with the corresponding classes but have little semantic overlap. However, the words "basketball" and "athletes" are both statistically and semantically close to the their corresponding classes. The word "three-pointer", is not quite common like "basketball", but is also semantically related to its corresponding class. The differences of these words in this case study inspire us to view the words of a given text from two perspectives:

[1] https://www.cluebenchmarks.com/

• **Statistical Correlation** with the class. This measures how frequent a word co-occurs with a class while not with other classes in the given dataset.
• **Semantic Similarity** with the class label, which measures how much semantics a word share with the class label.

Therefore, we can naturally divide the roles of different words of a given text through these two perspectives and get four **ROWs** (**R**oles **O**f **W**ords), as shown in Figure 2:

1. **CC-words**: **C**ommon **C**lass-indicating words, with *high* statistical correlation and *high* semantic similarity;
2. **SC-words**: **S**pecific **C**lass-indicating words, with *low* statistical correlation but *high* semantic similarity;
3. **IC-words**: **I**termidiate **C**lass-indicating words, with *high* statistical correlation but *low* semantic similarity;
4. **O-words**: Class-irrelevant words or **O**ther words, with *low* statistical correlation and *low* semantic similarity.

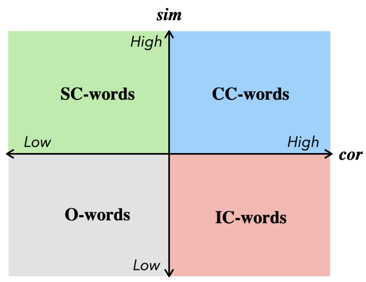

Figure 2: Four kinds of ROWs. ***sim*** refers to semantic similarity and ***cor*** refers to statistical correlation.

3.2 ROWs Extraction

To extract the roles of words in a dataset, we should decide proper metrics to measure the above two perspectives. For the measurement of **statistical correlation** with the class, we employ weighted log-likelihood ratio (WLLR) to select the class-correlated words from the text sample. This is inspired by (Yu and Jiang, 2016) where WLLR is used to find out the "pivot words" for sentiment analysis. The WLLR score is computed by:

$$wllr(w, y) = p(w|y)log(\frac{p(w|y)}{p(w|\bar{y})})$$

where w is a word, y is a certain class and $\bar{y}$ represents all the other classes in the classification dataset. $p(w|y)$ and $p(w|\bar{y})$ are the probabilities of observing w in samples labeled with y and with other labels respectively. We use the frequency of a word occurring in the certain class to estimate the probability.

To measure the **semantic similarity** between a word and the meaning of the class label, a straightforward way is to use word vectors pre-trained with skip-gram (Mikolov et al., 2013) or Glove (Pennington et al., 2014). We are not using transformer-based models like BERT for similarity measuring due to their high inference cost. Some also reveal that static word-embeddings can achieve comparable and even better performance than BERT-like models in similarity measurement tasks, especially in word-level (Reimers and Gurevych, 2019). We compute the cosine similarity between a word and a class label to see their semantic distance:

$$similarity(w, l) = \frac{v_w \cdot v_l}{\|v_w\| \|v_l\|}$$

where l represents the label and v_w, v_l are word vectors for the word w and the label l. We can also use a description of the label to obtain v_l by averaging the word vectors of each word in the description for better label representation. In our experiments, we find that simply using the word or phrase of the label itself is enough to measure the similarity between a word and the category.

We compute the WLLR and similarity of each word in a given sample, and set a threshold to divide the *high* and *low* scores. We call the words with high (low) WLLR socres as C_h (C_l) and words with high (low) similarity socres as S_h (S_l). By combining these words, we can extract the words of different roles as follows:

$$W_{CC} = \{w | w \in C_h \cap S_h\}$$
$$W_{SC} = \{w | w \in C_l \cap S_h\}$$
$$W_{IC} = \{w | w \in C_h \cap S_l\}$$
$$W_O = \{w | w \in C_l \cap S_l\}$$

where W_{CC}, W_{SC}, W_{IC} and W_O are CC-words, SC-words, IC-words and O-words respectively. A real ROWs extraction example in our experiments is depicted in Figure 3.

Figure 3: A real ROWs extraction example.

4 Text Augmentation based on ROWs

Traditional TE-based augmentation methods utilize text-editing operations on random words in the text, which we call *Random Text Augmentation* (RTA). From the perspective of ROWs, all roles have equal chance to be augmented during RTA, which means important class-indicating words may be changed and noisy or misleading words may be enhanced, resulting in undesirable augmented samples.

Instead, we propose to *augment the text based on the ROWs*. The reasons are twofold: 1) Different ROWs have different functions for the downstream classification tasks. Therefore, when utilizing different text augmentation operations, we should consider the role of each word in the text and select proper roles for augmentation, instead of randomly choosing the words. For example, CC-words are usually important class-indicating words, which should be protected from being damaged during augmentation; IC-words usually contain some noisy features thus better not be enhanced after augmentation. 2) Different augmentation operations are quite different in nature. Specifically, insertion aims to add more information to the sample, deletion aims to remove certain features from the sample, replacement can be seen as an insertion followed by a deletion, swap instead aims to change the formality of the original text. Therefore, different ROWs may be suitable for different augmentation operations.

From the case studies in Figure 1, we can see that *what the model actually learned are the **features of the training set**, rather than the **features of the classes***. A good augmentation on the training set should enlarge the overlap between the features of the training set and the features of the actual classes. RTA can bring in more features of the classes to the original dataset, but may also take in some undesirable features. However, by choosing the proper roles for augmentation, we are able to bring in more useful features to the training set while avoiding taking in undesirable features.

In the following section, we will conduct extensive experiments to see *what ROWs should (n't) be augmented for each text-editing augmentation operation* and discuss in detail why certain ROWs are suitable for certain augmentation operations. After the experimental results, we will propose a set of *Selective Text Augmentation* (STA) methods for better TE-based augmentation.

5 Experiments

5.1 Experimental Setup

Datasets. We use 9 benchmark text classification datasets for evaluation: **NG**, a subset from the 20NG datasets[2]; **Talk** and **Sci** are the news from the "talk" and "sci" groups of 20NG respectively; **BBC**, a small set from the BBC News dataset (Greene and Cunningham, 2006); **Games** and **Finance** are both subsets of the iflytek Chinese classification dataset (Xu et al., 2020); **TNews** is Chinese short text classification dataset (Xu et al., 2020); **FD** and **TH** are two subsets from the news classification datasets collected by Fudan University and Tsinghua University respectively[3]. Note that some datasets are small subsets from the original large versions, for simulating the low-resource scenarios. The meta information of these datasets is shown in Table 2.

Datasets	# train	# test	# labels	Avg Len
Games (zh)	2.4k	0.5k	9	255
Finance (zh)	1k	0.2k	8	306
TNews (zh)	53k	10k	15	22
FD (zh)	0.5k	2.9k	6	5233
TH (zh)	0.5k	2k	13	994
BBC (en)	0.5k	0.9k	5	476
NG (en)	1k	7.5k	20	575
Talk (en)	1.9k	1.3k	4	654
Sci (en)	2.4k	1.6k	4	480

Table 2: Datasets information. "zh" and "en" refer to Chinese and English respectively.

Training Settings. In this work, we use TinyBert (Jiao et al., 2020) as the backbone of our text classifiers, which is a lighter transformer-based model but shows comparable performance with large transformer-based models. For synonyms/similar words searching, we use public skip-gram word embeddings. For ROWs extraction, we use the *median* number as the bar for dividing high and low scores (we also tested the mean, upper/lower quartile but found using median is relatively better). We set the proportion of words changed during augmentation to be 10%, as recommended in (Wei and Zou, 2019). If the role words are less than 10%, random words will be sampled from the text as supplements. We randomly choose 20% of the training set as the validation set, use the AdamW (Loshchilov and Hutter, 2017) optimizer and use early-stopping with patience $p = 3$ to choose the best model. We run all experiments 10 times and report the average performance.

5.2 Experiments of Augmentation on Different ROWs

Deletion, (synonyms) replacement, (synonyms) insertion and swap are all widely used text-editing operations for augmentation. In this part, we conduct experiments on the impact of augmenting certain ROW by each TE-based operations.

In the experiments of each operation, we compare **six** methods: **non-aug**, which means no augmentation is applied; $\star$**-RTA**, which means augmenting using the given operation in a non-selective manner, like the practice in (Kolomiyets et al., 2011; Zhang et al., 2015; Wang and Yang, 2015a; Wei and Zou, 2019; Feng et al., 2020); $\star$**-CC**, $\star$**-SC**, $\star$**-IC** and $\star$**-O** are augmentation based on different ROWs. ($\star$ refers to a certain operation.) Note that RTA in previous works may differ in some details, therefore, for fair comparison, we implement RTA methods in the same way as our ROWs-based augmentation with the only difference in words selection process. The experimental results are shown in Table 3 to Table 6.

5.2.1 Augmentation by Deletion

According to Table 3, we have two important observations: 1) Deleting the **CC-words** are likely to hurt the performance, since the classification accuracy of **d-CC** is worse than **non-aug** in most datasets. 2) Deleting **SC-words** or **IC-words** brings more performance gain than other deletion strategies.

The reason why d-CC performs worst is that the CC-words are usually important class-indicating words, if these words are destroyed, the label of the original text are likely to be changed. On the other hand, SC-words are usually those less frequently co-occurred with the corresponding category but are semantically similar with the category, deleting these words will force the model to concentrate

[2]https://www.cs.umb.edu/ smimarog/textmining/datasets/
[3]https://www.cluebenchmarks.com/

Methods	Games	Finance	TNews	FD	TH	BBC	NG	Talk	Sci	*Avg.*	*rank*
non-aug	62.51	80.67	52.68	95.71	91.02	93.44	59.04	82.30	93.94	79.03	3.6
d-RTA	61.95	80.27	**52.89**	95.55	90.60	94.25	59.84	**83.17**	**94.99**	79.28	3.4
d-CC	58.85	80.22	52.24	95.24	90.85	93.18	47.51	82.57	92.98	77.07	5.7
d-SC	**62.55**	**82.65**	52.64	95.66	**91.77**	94.39	**60.22**	82.87	94.44	**79.69**	**2.2**
d-IC	62.42	80.67	52.38	**95.89**	90.61	95.39	60.14	83.12	94.95	79.51	2.8
d-O	62.03	81.43	52.53	95.59	91.41	**95.45**	56.11	82.78	94.89	79.14	3.2

Table 3: Classification accuracy (%) comparison of different **Deletion** strategies. RTA: Random Text Augmentation. CC, SC, IC and O are four ROWs. *rank* means the average rank of certain methods across all datasets. The **bold** numbers are the best across all methods and the underlined numbers are the best among all ROWs.

Methods	Games	Finance	TNews	FD	TH	BBC	NG	Talk	Sci	*Avg.*	*rank*
non-aug	62.51	80.67	**52.68**	95.71	91.02	93.44	59.04	82.30	93.94	79.03	3.9
i-RTA	61.56	81.88	52.51	95.38	91.22	94.39	53.91	83.48	92.52	78.54	4.1
i-CC	**63.55**	**82.38**	52.36	95.82	90.96	94.51	**59.52**	83.31	94.31	**79.64**	**2.8**
i-SC	61.58	81.30	52.28	95.83	**91.65**	94.01	57.70	**83.65**	**94.62**	79.18	3.2
i-IC	62.09	80.09	52.63	**95.86**	91.36	**95.27**	57.62	83.43	93.49	79.09	3.1
i-O	62.12	81.57	52.29	95.84	91.60	94.34	49.52	82.97	93.32	78.17	3.9

Table 4: Classification accuracy (%) comparison of different **Insertion** strategies. The meanings of RTA, CC, SC, IC, O, *rank*, **bold**/underlined numbers can be found in Table 3.

more on the CC-words, which server as common class-indicating features in most samples of the same class. IC-words usually contain some noise brought by the biased data distribution of the limited dataset. Deleting these IC-words thus helps the model to avoid learning some incorrect features about the categories. As for random deletion, all ROWs will have equal chance to be deleted, therefore the performance of d-RTA is better than d-CC but worse than d-SC or d-IC.

5.2.2 Augmentation by Insertion

The results of insertion shown in Table 4 illustrates different patterns from the results for deletion: 1) The best choice is to inserting the similar words of the **CC-words**. 2) **i-RTA** or **i-O** are relatively worse than other methods, even worse than **non-aug** on average.

I-CC performs best in insertion because more class-relevant words are inserted into the text, resulting in a high quality augmented sample whose class-related information is enhanced and is also different from the original text in representation in the same time. As for SC and IC-words, though these words are not such representative as the CC-words, they are still class-indicating in some degree, therefore inserting their synonyms can also generate useful samples. However, inserting the similar words of O-words may face lots of uncontrollability, since these words may include some class-indicating words of other classes, which can severely change the meaning of the original text. This may be the reason why i-RTA and i-O are not performing well in many datasets.

5.2.3 Augmentation by Replacement

The results of the replacement experiments shown in Table 5 are though-provoking: 1) Replacing the **CC-words** by their similar words usually leads to worse performance; 2) Replacing **SC-words** is the best strategy among these replacing methods relatively.

It is interesting why r-CC gets the worst results. Intuitively, replacing those class-indicating words with their similar words won't change the label of the original text, if so, the augmented samples should bring more diversity of the category and bring some performance gain. To investigate this, we check the similar words given by the word embeddings or WordNet (Miller, 1995), and found that these similar words usually don't have identical meaning of the original word, which may cause semantic drift or bring in some noise. Therefore, replacing the CC-words may have risk to influence the core meaning or even change the label of the original text. Compared with CC-words, SC-words are not that important to represent the core meaning of the text, but are also class-indicating in semantics, thereby, replacing these words can bring in more diversity of the class-indicating features

Methods	**Games**	**Finance**	**TNews**	**FD**	**TH**	**BBC**	**NG**	**Talk**	**Sci**	*Avg.*	*rank*
non-aug	**62.51**	80.67	52.68	**95.71**	91.02	93.44	59.04	82.30	93.94	79.03	3.2
r-RTA	61.83	**81.48**	52.38	95.49	91.09	94.47	57.15	83.26	93.48	78.96	3.4
r-CC	61.42	79.64	52.50	94.82	90.44	94.31	51.33	**83.61**	93.77	77.98	4.8
r-SC	62.34	81.03	52.15	95.10	91.06	**95.51**	**60.46**	82.88	**94.77**	**79.48**	**2.9**
r-IC	62.44	81.39	**52.78**	95.07	90.35	95.34	52.26	83.36	94.71	78.63	3.1
r-O	60.49	80.76	52.56	95.57	**91.71**	95.38	54.99	82.20	93.83	78.61	3.6

Table 5: Classification accuracy (%) comparison of different **Replacement** strategies. The meanings of RTA, CC, SC, IC, O, *rank*, **bold**/underlined numbers can be found in Table 3.

Methods	**Games**	**Finance**	**TNews**	**FD**	**TH**	**BBC**	**NG**	**Talk**	**Sci**	*Avg.*	*rank*
non-aug	62.51	80.67	52.68	95.71	91.02	93.44	**59.04**	82.30	93.94	79.03	3.9
s-RTA	62.07	79.46	**52.85**	**95.79**	90.79	93.72	52.53	82.77	94.40	78.26	3.4
s-CC	61.03	**82.69**	52.46	94.89	90.62	94.67	57.66	**83.69**	93.26	79.00	4.1
s-SC	61.64	81.48	52.72	95.70	**91.76**	93.63	54.39	82.44	93.55	78.59	3.9
s-IC	62.85	82.33	52.65	95.75	91.33	94.69	55.19	81.52	94.13	78.94	3.1
s-O	**63.20**	81.39	52.49	95.74	91.36	**94.73**	56.59	82.69	**94.75**	**79.22**	**2.6**

Table 6: Classification accuracy (%) comparison of different **Swap** strategies. The meanings of RTA, CC, SC, IC, O, *rank*, **bold**/underlined numbers can be found in Table 3.

while not changing the true label.

5.2.4 Augmentation by Swap

Swap is less effective compared with other operations (deletion, insertion and replacement) according to our experimental results in Table 6, and the impact of swapping different ROWs varies a lot across these datasets. However, we can see that swapping the **O-words** is relatively better and stable across the datasets according to the average rank and accuracy, since swapping these words has least impact on the core semantics but can also bring some change to the formality of the original text.

5.3 STA: Select Suitable ROWs for Text Augmentation

Based on the experimental results from the above experiments, we can now summarize a set of general recommendations for selecting ROWs as listed in Table 7. With these general recommendations we can implement our text augmentation in a *selective* manner, which we call **STA** (**S**elective **T**ext **A**ugmentation).

According to (Wei and Zou, 2019), the augmentation performance is usually stronger if we use these augmented samples generated by different operations together. Therefore, we aggregate these operations altogether and see whether STA can perform better than traditional RTA method. We call the aggregated random augmentation operations as **agg-RTA** and the aggregated STA operations as **agg-STA**.

Operations	**Recommend**	**Non-recommend**
deletion	SC-words, IC-words	CC-words
insertion	CC-words	O-words
replacement	SC-words	CC-words
swap	O-words	-

Table 7: ROWs recommendation/non-recommendation board for TE-based augmentation methods.

Specifically, for **agg-STA** in each dataset, we use the same rules for augmenting: Select **IC-words** for **deletion**, **CC-words** for **insertion**, **SC-words** for **replacement** and **O-words** for **swap**. The evaluation results are illustrated in Table 8.

The results demonstrate that **agg-STA** can bring significant performance gain for all 9 datasets with an average improvement of 2.22%, and is superior to **agg-RTA** on 7 out of 9 datasets with an average improvement of 0.86%. By contrast, **agg-RTA** even hurts the classification performance of TNews and FD datasets. Note that we are not using the best practice of ROWs for each dataset for **agg-STA**, instead, we use the general recommendations for all the datasets. Therefore, by carefully study the nature (data source, text style, etc.) of certain dataset and tune the ROWs allocations, STA will have potential to perform even better, which we will study in future work.

Methods	Games	Finance	TNews	FD	TH	BBC	NG	Talk	Sci	*Avg.*
non-aug	62.51	80.67	52.68	95.71	91.02	93.44	59.04	82.30	93.94	79.03
agg-RTA	62.57	84.39	52.43	95.40	91.40	95.06	62.44	84.38	**95.44**	80.39
improvement	*+0.06*	*+3.72*	*-0.25*	*-0.31*	*+0.38*	*+1.62*	*+3.40*	*+2.08*	*+1.50*	*+1.36*
agg-STA (Ours)	**64.78**	**84.51**	**52.97**	**96.02**	**92.30**	**95.84**	**64.94**	**84.47**	**95.44**	**81.25**
improvement	*+2.27*	*+3.84*	*+0.29*	*+0.31*	*+1.28*	*+2.40*	*+5.90*	*+2.17*	*+1.50*	*+2.22*

Table 8: Classification accuracy (%) and improvement (%) of different augmentation strategies. Underlined numbers are significantly superior based on student t-test.

Methods	FD⇒TH	TH⇒FD	FD⇒BBC	BBC⇒FD	TH⇒BBC	BBC⇒TH	*Avg.*
non-aug	75.62	38.50	74.25	34.12	45.53	70.86	56.48
agg-RTA	**76.54**	39.93	77.24	38.38	46.34	69.81	58.04
improvement	*+0.92*	*+1.43*	*+2.99*	*+4.26*	*+0.81*	*-1.05*	*+1.56*
agg-STA (Ours)	75.31	**46.68**	**82.66**	**41.44**	**52.03**	**74.57**	**62.12**
improvement	*-0.31*	*+8.18*	*+8.41*	*+7.32*	*+6.50*	*+3.71*	*+5.64*

Table 9: Cross-dataset prediction tasks. A⇒B means the model is trained on A dataset and evaluated on the shared classes of B dataset. Underlined numbers are significantly superior based on student t-test.

5.4 Cross-dataset Evaluation

As we have mentioned in former parts, the classification models may be ill-trained even if they perform well on the held-out test set, since the test set may contain the same biases of the training set. This phenomenon is also described in (Ribeiro et al., 2020). A more convincing evaluation is to test the models "in the wild", which however is too expensive. Fortunately, we find that the FD, TH and BBC datasets share two common classes: "politics" and "sport". Though different in data source, text style and even in language, the common categories of these datasets have the same general meaning. Therefore we can design a series of cross-dataset evaluation tasks to simulate the "wild" evaluation scenarios, which can serve as a supplementary test set of the original test set.

Specifically, we train the classifier on the original dataset A, and then test it on the common categories of dataset B. If B is of a different language, we will first translate B into the same language of A using open-sourced translation models [4]. The results are shown in Table 9 which demonstrate that **agg-STA** significantly outperforms **agg-RTA** in 5 out of 6 cross-dataset prediction tasks with more than 4% accuracy improvement over **agg-RTA**.

Compared with Table 8, we can see that the improvement of STA over traditional methods is much larger in this cross-dataset evaluation. This is likely due to the fact that some of the STA operations are aimed to decrease the biases of the training set or enhance the core semantics of the class-related parts of the samples. For example, deleting the IC-words will help the model to learn less "fake" class-indicating features brought by the biases of the training set. Specifying the insertion on CC-words will enhance the class-indicating parts of the sample. Both of these two operations are vital for better generalization ability. Evaluating only on the original test set may obscure some of the actual effect of our proposed STA methods.

6 Conclusion & Future Work

In this work, we present four types of roles of words (ROWs) and design effective methods to extract them. Each ROW has unique function for downstream tasks like text classification. We conduct comprehensive experiments to investigate the impact of augmenting on different ROW and discover interesting patterns behind popular augmentation methods including deletion, insertion, replacement and swap. We then propose *Selective Text Augmentation* methods with which we can generate a better augmented training set with higher quality and significantly improve the generalization ability of text classifiers. Actually, the idea of ROWs can also be applied to other tasks like keyphrases extraction, document representation and even image classification (where we can study the Roles of Superpixels), which will be in our future work.

[4]https://huggingface.co/Helsinki-NLP/opus-mt-en-zh, https://huggingface.co/Helsinki-NLP/opus-mt-zh-en

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Jacob Andreas. 2019. Good-enough compositional data augmentation. *arXiv preprint arXiv:1904.09545*.

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2021. A survey on data augmentation for text classification. *arXiv preprint arXiv:2107.03158*.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2021. An empirical survey of data augmentation for limited data learning in nlp. *arXiv preprint arXiv:2106.07499*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

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.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4163–4174.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Sosuke Kobayashi. 2018a. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.

Sosuke Kobayashi. 2018b. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2011. Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, volume 2, pages 271–276. ACL; East Stroudsburg, PA.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for nlp tasks. *arXiv preprint arXiv:2010.02394*.

Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiaji Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin. 2018. Topic compositional neural language model. In *International Conference on Artificial Intelligence and Statistics*, pages 356–365.

William Yang Wang and Diyi Yang. 2015a. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563.

William Yang Wang and Diyi Yang. 2015b. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

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

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.