Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/21/20231140

Research on improving the generalization ability of NLP categorization models: A comparison of Mixup and generative adversarial networks

Jiyan Chen

School of International Liberal Studies, WASEDA Univ., 1–104 Totsukamachi, Shinjuku-ku, Tokyo, Japan

chin_kigen@toki.waseda.jp

Abstract. The application of deep neural networks in tasks related to natural language processing has grown in recent years. However, training deep learning models often requires a large amount of data to produce accurate and generalizable results. When the data are insufficient, categorization models may suffer from overfitting or poor performance. This paper compares two methods for enhancing the generalization ability of natural language processing categorization models when data are insufficient: Mixup and Generative Adversarial Networks (GANs). This paper uses the Internet Movie Database (IMDB) dataset for a binary classification task, Mixup method, and Relational Generative Adversarial Networks (RelGAN) model to generate new data, respectively, and compares them with the original dataset. Experimental results indicate that Mixup method can reduce overfitting risk effectively and enhance the model's robustness, while the GAN model does not show obvious advantages. This paper supports the idea that using noise-adding operations to enhance the dataset is feasible in the case of small samples rather than using generative models.

Keywords: natural language processing, generative adversarial networks, Mixup, text categorization, deep learning.

1. Introduction

Among the tasks performed in natural language processing (NLP), deep learning has become increasingly useful for sentiment analysis, topic modelling, and text categorization. Typically, these deep learning models use millions of parameters, makes produce an accurate and generalized model requiring a dataset with abundant samples [1]. According to the OpenAI team, the high-performance model GPT-3 contains roughly 175 billion parameters, and about 400 billion tokens have been used to train the model [2]. Even though deep learning models perform optimally on massive data sets, feature-engineered statistical models still outperform deep learning models in the absence of sufficient data, argued by Şahin and Steedman [3]. Hence, it is of great interest to devise a universal approach to augment data in scenarios where data availability is limited.

In the field of computer vision, image manipulation methods such as flipping, rotation, scaling, and noise injection are applied to augment the dataset, through which the accuracy and robustness could be improved after training [4]. Wei and Zou got inspiration from these methods and applied random

DOI: 10.54254/2755-2721/21/20231140

operations, which do not change the potential meaning of the sentence, on the dataset and finally improved the performance of the model [5]. This proves the feasibility of using data augmentation as a complement with a limited amount of dataset. Nevertheless, since words with close or exact meanings are rarely found, operations will not change the potential meaning would be limited at a part of the data [1]. Lately, using a Mixup method, which combines two or more images to generate new images, was introduced to augment data [4]. Similarly, Guo et al. come up with a Mixup method that can be applied to NLP tasks. Through performing Mixup by interpolating at the word embedding or sentence embedding stage, they succeeded in improving the accuracy of the model [1]. However, the Mixup method may introduce noise or irrelevant information, making it difficult for the model to distinguish important features from irrelevant ones [6].

In addition to Mixup methods, using Generative Adversarial Networks (GANs) to generate new data for data augmentation has appeared in recent years. GANs consist of two sub neural networks: a generative network which is trained for capturing the distribution of data and a discriminative network that determines if a sample originated from the real data or the generative network. [7]. An original purpose of GANs is to generate continuous data such as images [8]. Recently, to generate discrete data including text sequences, many researchers have tried to extend this model to the field of NLP [8]. However, the major problem with the GAN model is model collapse, which means that the generated sentences are too similar and not diverse enough to improve the dataset [8].

Both Mixup methods and GANs have their own advantages and drawbacks, which makes it hard to determine the best method for a given dataset. To provide guidance on the conditions under which traditional Mixup methods or GANs models should be taken, as well as exploring how these two methods can be improved in the future, this paper compares the performance of the categorization model using traditional Mixup methods and GANs models to generate new augmentation data on datasets with different sizes for training.

2. Methodology

2.1. Collection and preprocessing of the dataset

This paper uses one public dataset, namely the Internet Movie Database (IMDB) dataset, in the categorizing task. The IMDB dataset is a collection of movie reviews from the Internet Movie Database, a popular online platform for movie information and reviews. The dataset includes 50000 samples. Each sample has a piece of review and a label that indicates it is a positive review or a negative one [9].

It is a typical step to apply preprocessing before transforming the text into vector representation in order to reduce the noise that can exist in raw data, such as spelling mistakes, grammatical errors, internet terms, and abbreviations [10]. Tokenization, case transformation, stop-word removal, weighting, and stemming are commonly applied during preprocessing [11]. Finally, all texts are padded to the same length for input into the embedding layer [12], [13]. Figure 1 shows the detailed steps of preprocessing in this paper.

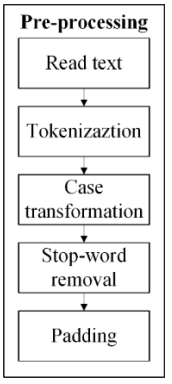

Figure 1. Steps of preprocessing. Picture credit: Original.

DOI: 10.54254/2755-2721/21/20231140

2.2. Augmenting data

2.2.1. Improving dataset by using wordMixup. The wordMixup method is a data augmentation technique for text classification that generates new samples by mixing two original samples with different labels [14]. The mixing process can be represented by the following equation:

$$\begin{cases} \tilde{x}^{ij} = \lambda x^i + (1-\lambda)x^j \\ \tilde{y}^{ij} = \lambda y^i + (1-\lambda)y^j, \end{cases} \quad (1)$$

where (x^i, y^i) and (x^j, y^j) are a pair of samples (x indicates the input and y the corresponding class of the sample), λ is the mixing ratio and belongs to the interval $[0,1]$. For example, if there are two original samples, "I love this movie. It was so funny and entertaining.", and "This movie was terrible. It was boring and predictable." with positive and negative labels, respectively. A new sample can be generated by mixing them with an interpolation coefficient of 0.6: "I love this movie. It was boring and entertaining.", which has a label of 0.6 * Positive combined with 0.4 * Negative. Figure 2 demonstrates the flow of data augmentation using the Mixup method.

2.2.2. Improving dataset by RelGAN model. The Relational Generative Adversarial Networks (RelGAN) model uses a relational memory-based generator that can model long-distance dependencies in text sequences using self-attention mechanisms, which allows the generator to capture the semantic and syntactic structures of natural language and produce coherent and fluent text samples [8]. To augment the raw data, a pre-trained RelGAN generator is used to sample the raw dataset and get new text samples. Then, combine the new text sample with the original data source as the augmented dataset [15]. Finally, the augmented dataset is used for training the classification model. The detail of the flow is shown in Figure 3.

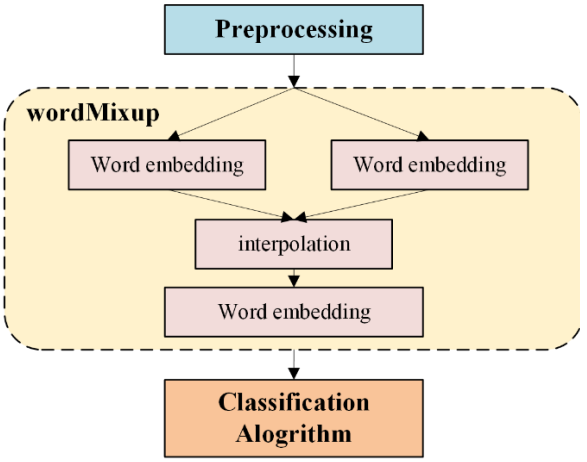

Figure 2. Steps of wordMixup augmenting.

Picture.

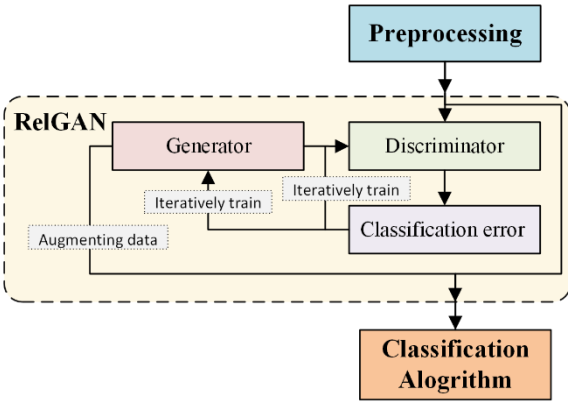

Figure 3. Steps of RelGAN augmenting.

Picture.

2.3. Classification algorithm

According to Minaee's review of several popular deep learning algorithms for text classification, the LEAM model has demonstrated strong performance across multiple datasets [16]. The LEAM model simultaneously embeds text sequences and category labels into a single hidden space to construct document representations [17]. However, the Mixup method also performs interpolation on the embedding stage to generate new samples [1]. In this case, Deep Pyramid Convolutional Neural

DOI: 10.54254/2755-2721/21/20231140

Networks, which use multiple convolution and pooling operations to build a pyramid-shaped network structure after embedding, are used for training [18]. The DPCNN model could capture the relationship between the context of the sentence, which makes it easier for the model to identify whether new patterns are generated in the augmenting samples.

3. Experimental results and analysis

3.1. Experimental setting and evaluation index

This paper takes an average of 10% of the data under each label in the IMDB dataset as a raw dataset to simulate an inadequate dataset.

To evaluate the effect of the Mixup method and RelGAN model, this paper first using a pre-processed raw dataset (called the origin dataset) to train a DPCNN baseline model, and then uses the datasets by combining the raw dataset with dataset augmenting through the Mixup and GANs model (called Mixup augmenting dataset and GAN augmenting dataset separately) to train the model for comparing with the baseline one. Unified hyperparameters, as shown in Table 1, are used to train the models.

Table 1. The hyperparameters used in training.

Hyperparameter	Value	Explanation
num_classes	2	The number of categories in the original dataset
minlen	10	Minimum review length generated by GAN
maxlen	400	Maximum length of each review
embedding_dim	50	Dimensions of word vectors
hidden_dim	250	The dimension of hidden layers
batch_size	32	Size of each batch
num_pre_epoch	40	The number of epochs in pretraining GAN
num_epochs	10	The number of training epochs
generated_num	5000	The number of reviews generated by GAN
alpha	0.3	Interpolation coefficient in wordMixup
lr	0.01	The learning rate

When comparing the performance of the models, this paper utilizes the cross-entropy loss function and accuracy as the evaluation criteria for the model, based on their practicality and effectiveness. The cross-entropy loss function, which calculates the difference between the predicted value and the actual label, makes it a frequently employed method for classification tasks. Furthermore, it aligns with the maximum likelihood estimation approach [19], [20]. Meanwhile, accuracy is an intuitive metric that indicates the proportion of correctly predicted samples out of the total number of samples, and it can reflect the performance of the model on different categories [21]. Therefore, this paper uses the following formulas to calculate the cross-entropy loss function and accuracy:

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p_i) + (1 - y_i)\log(1 - p_i) \tag{2}$$

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

where N stands for the sample counts, y_i stands for the sample's actual label, p_i stands for the sample's predicted probability, TP stands for true positives counts, TN stands for true negatives counts, FP stands for false positives counts, and FN stands for false negatives counts.

3.2. Performance of the baseline model

This paper uses the hyperparameters in Table 1 to train the DPCNN baseline model, and during each training epoch, the loss of both training set and validation set are recorded as well as the accuracy. Table 2 demonstrates the details of how the performance of the baseline model changes during training, the loss and accuracy of training set and validation set of each training epoch are listed.

DOI: 10.54254/2755-2721/21/20231140

Table 2. Detail of the training and validation of the DPCNN baseline model trained by the origin dataset.

Epoch	Val. Acc. (%)	Train Acc. (%)	Val. Loss	Train Loss
1	88.64	80.55	0.2732	0.3994
2	89.32	91.03	0.2661	0.2297
3	86.61	93.84	0.3329	0.1629
4	88.41	95.83	0.3008	0.1152
5	87.33	97.36	0.3851	0.0728
6	88.20	98.00	0.4121	0.0555
7	88.19	98.39	0.4134	0.0444
8	88.13	98.61	0.4303	0.0387
9	88.05	98.91	0.5059	0.0310
10	88.03	98.97	0.5099	0.0304

During the training process, the loss on the training set decreased rapidly. However, the loss on validation set starts to increase after 4 epochs of training. Also, the accuracy on validation set indicates a decreasing tendency.

3.3. Performance of model training by augmenting datasets

3.3.1. The performance of wordMixup. To evaluate the performance of wordMixup augmenting, linear interpolation was applied after representing all the text data into vectors. Then, the data are input into the classification model for training. Table 3 indicates the performance of the model after using wordMixup augmenting algorithm.

Table 3. Detail of the training and validation of the DPCNN baseline model trained by the Mixup augmenting dataset.

Epoch	Val. Acc. (%)	Train Acc. (%)	Val. Loss	Train Loss
1	64.80	28.14	0.6406	0.6854
2	72.23	31.81	0.5821	0.6602
3	74.41	32.99	0.5292	0.6423
4	75.22	32.90	0.5193	0.6268
5	73.58	32.78	0.5222	0.6168
6	78.24	33.95	0.4644	0.6020
7	79.17	33.57	0.4403	0.5962
8	80.10	34.29	0.4224	0.5885
9	78.92	33.76	0.4566	0.5870
10	80.17	33.92	0.4203	0.5780

According to the results, there is a downward trend in the loss on both training set and validation set. Yet the accuracy on the validation set only remains about 80%.

3.3.2. The performance of RelGAN. This paper first uses positive data in the dataset to pre-train the generator, negative data in the dataset to pre-train the discriminator, and then uses the pre-training generator and discriminator for adversarial training to obtain the generator that generates positive

DOI: 10.54254/2755-2721/21/20231140

reviews. Then, the pre-training data sets are exchanged to get another generator that generates negative reviews. Finally, use the two generators to generate 5000 new comments each to augment the raw dataset. Table 4 shows the training performance using the GAN augmented dataset.

Table 4. Detail of the training and validation of the DPCNN baseline model trained by the GAN augmenting dataset.

Epoch	Val. Acc. (%)	Train Acc. (%)	Val. Loss	Train Loss
1	64.80	28.14	0.6406	0.6854
2	72.23	31.81	0.5821	0.6602
3	74.41	32.99	0.5292	0.6423
4	75.22	32.90	0.5193	0.6268
5	73.58	32.78	0.5222	0.6168
6	78.24	33.95	0.4644	0.6020
7	79.17	33.57	0.4403	0.5962
8	80.10	34.29	0.4224	0.5885
9	78.92	33.76	0.4566	0.5870
10	80.17	33.92	0.4203	0.5780

Accordingly, the accuracy on the training set reached 99.70% after 10 epochs of training, whereas the average accuracy is slightly below the baseline model. Moreover, the loss on validation set still shows an upward trend.

3.4. Comparison and analysis

Comparing the model performance of using origin and augmenting data as the training dataset, the model performance on the training set outweighs that on the validation set, with a training loss of 0.0304 and a validation loss of 0.5099. In other words, the model performs well on the training set, but shows a poor performance on the validation set. Therefore, training the model with the origin dataset would have a high risk of overfitting. Nevertheless, if wordMixup augmenting algorithm is added before applying classification algorithm, the performance on the training set becomes very poor, with an average accuracy below 40% and loss above 0.5. When considering the performance on the validation set, the loss decreases continuously throughout the training progress. To be specific, adding wordMixup augmenting could effectively reduce the risk of overfitting. A possible explanation for this could be that the introduction of wordMixup augmenting increases the difficulty of training the model. To clarify, using wordMixup as an augmenting of the insufficient dataset could increase the diversity of the sample, make the features of the target more difficult to be identified. Thus, Mixup is a strategy that can be considered under the condition of insufficient data. In contrast, applying the Mixup method has the risk of introducing patterns not included in origin datasets. To rephrase it, since Mixup randomly, the Mixup method applies random linear interpolation on samples. This process generates new samples that may have patterns that the original samples did not possess, and the model learns these patterns during training, which affects the performance of the model. Also, the decreased performance on validation set could be explained by this factor.

In contrast to the progress made by Mixup methods, there is no significant improvement using GAN to augment the dataset. Compared with the model performance, which used the origin data for training, the model becomes overfitted earlier after applying the GAN algorithm. The loss on validation set expands rapidly after 3 epochs of training. This result may be explained by the fact that the GAN model requires more data to reach a good Nash equilibrium, and avoid mode collapse problem [7]. As a result, using the data generated by GAN model training with a small sample as an augmenting would not increase sample diversity. Therefore, the generated data will strengthen the model's ability to recognize certain patterns, which leads the overfitting of the model. Also, the target of the GAN model is to generate data that matches the true distribution of the data. If the small sample data used for training is biased, the GAN model will provide biased data as well.

DOI: 10.54254/2755-2721/21/20231140

4. Conclusions

This study set out to compare the two ways of data augmenting under the condition that the samples are not sufficient. Through comparing the models' performance by training them with a dataset applying various augmenting algorithms, the finding indicates that the Mixup method is an effective way to augment a small dataset. However, using Generative Adversarial Networks to generate augmenting datasets does not demonstrate superior performance on a small dataset. When the data are insufficient, using GAN to generate new data could expand the amount of data, but the diversity of data could not be improved. Put differently, the GAN model trained in the case of small samples, the generated data is similar to the replication of the original data, which could not actually improve the performance of the model, instead it makes the model become over-fitting even faster. The results of this paper support the idea that the Mixup method is a better data augmenting method than using the GAN model when the number of samples is small. This finding indicates that various noise-adding operations on the dataset may be a more effective way to enhance the robustness of the trained model than using generative models under a small sample situation. On the other hand, several limitations of this paper need to be acknowledged. The current study only examined the performance of the Mixup and GAN model on a binary classification dataset. Consequently, these results may not be applicable to multiple classification tasks. Further experiments, using a broader range of datasets with various labels and amounts, could shed more light on exploring the potential of the Mixup method and GAN model in the field of data augmentation.

References

[1] Guo H, Mao Y and Zhang R 2019 Augmenting Data with Mixup for Sentence Classification: An Empirical Study

[2] Brown T B, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D M, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I and Amodei D 2020 Language Models are Few-Shot Learners

[3] Şahin G G and Steedman M 2018 Data Augmentation via Dependency Tree Morphing for Low-Resource Languages Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (Brussels, Belgium: Association for Computational Linguistics) pp 5004–9

[4] Yang S, Xiao W, Zhang M, Guo S, Zhao J and Shen F 2022 Image Data Augmentation for Deep Learning: A Survey

[5] Wei J and Zou K 2019 EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) EMNLP-IJCNLP 2019 (Hong Kong, China: Association for Computational Linguistics) pp 6382–8

[6] Li B, Hou Y and Che W 2022 Data augmentation approaches in natural language processing: A survey AI Open 3 71–90

[7] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2020 Generative adversarial networks Commun. ACM 63 139–44

[8] Nie W, Narodytska N and Patel A 2018 RelGAN: Relational Generative Adversarial Networks for Text Generation International Conference on Learning Representations

[9] TensorFlow 2020 IMDb Movie Reviews Dataset

[10] Hartmann J, Huppertz J, Schamp C and Heitmann M 2019 Comparing automated text classification methods Int. J. Res. Mark. 36 20–38

[11] Yang Y 1999 An Evaluation of Statistical Approaches to Text Categorization Inf. Retr. 1 69–90

[12] Marivate V and Sefara T 2020 Improving short text classification through global augmentation

DOI: 10.54254/2755-2721/21/20231140

methods International Cross-Domain Conference for Machine Learning and Knowledge Extraction (Springer) pp 385–99
[13] Nie W, Narodytska N and Patel A 2019 RelGAN: Relational Generative Adversarial Networks for Text Generation International conference on learning representations (ICLR)
[14] Zhang H, Cisse M, Dauphin Y N and Lopez-Paz D 2018 Mixup: Beyond Empirical Risk Minimization
[15] Yu L, Zhang W, Wang J and Yu Y 2017 SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient Proc. AAAI Conf. Artif. Intell. 31
[16] Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M and Gao J 2021 Deep Learning Based Text Classification: A Comprehensive Review
[17] Wang G, Li C, Wang W, Zhang Y, Shen D, Zhang X, Henao R and Carin L 2018 Joint Embedding of Words and Labels for Text Classification
[18] Johnson R and Zhang T 2017 Deep Pyramid Convolutional Neural Networks for Text Categorization Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) ACL 2017 (Vancouver, Canada: Association for Computational Linguistics) pp 562–70
[19] Jadon S 2020 A survey of loss functions for semantic segmentation 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) pp 1–7
[20] Mao A, Mohri M and Zhong Y 2023 Cross-Entropy Loss Functions: Theoretical Analysis and Applications
[21] Fei R, Yao Q, Zhu Y, Xu Q, Li A, Wu H and Hu B 2020 Deep Learning Structure for Cross-Domain Sentiment Classification Based on Improved Cross Entropy and Weight Sci. Program. 2020 e3810261