Semantic Preservation and Natural Language Data Augmentation via Variational Autoencoder

Yuchul Shin[1], Kyohoon Jin[2], Juhwan Choi[3], Junho Lee[4], Soojin Jang[2], and Youngbin Kim[2]*

[1] *School of Game, Chungkang College of Cultural Industries, Icheon, Korea*

[2] *Department of Image Science and Arts, Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University/ Seoul, Korea*

[3] *School of Electrical and Electronics Engineering, Chung-Ang University/ Seoul, Korea*

[4] *Department of Artificial Intelligence, Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University/ Seoul, Korea*

**Corresponding Author (ybkim85@cau.ac.kr)*

Abstract: Text augmentation, unlike image augmentation, is challenging because text modifications directly affect labels. Studies on text augmentation using generative and pretrained language models (PLMs) have been conducted; however, their application has limitations. This study proposes a PLM-based data augmentation technique using a variational autoencoder (VAE) structure. Latent variables were used to better understand the semantics, and the VAE was used to assign randomness. The PLM was placed in the encoder and decoder to improve the augmentation performance. We evaluated our proposed method on two benchmark datasets and demonstrated its augmentation effect.

Keywords: Text augmentation, Latent variable, Natural Language Processing.

Received Sep. 22, 2022; accepted for publication Oct. 5, 2022; published online Oct. 31, 2022. DOI: 10.15323/techart.2022.10.9.3.24 / ISSN: 2288-9248.

1. Introduction

Deep learning techniques have improved the processing of various data types, such as images and voices. The development of such deep learning techniques is the same as the processing of natural language data. Many deep learning architectures, from recurrent neural networks (RNNs) to transformers, perform outstanding natural language processing (NLP) tasks. Recent language models pretrained with a large amount of data, such as BERT, outperform humans on such tasks. The performance of these models is influenced by the amount and diversity of the training data.

Because these deep learning technologies use data-based learning methods, the quantity and quality of the data have a significant influence. When insufficient data are available for training, the model is optimized only for the given data. Consequently, overfitting does not show good performance on new data that has not been observed during training. To solve this problem, various data augmentation techniques have been developed. Data augmentation refers to increasing the amount of training data. Data augmentation aims to improve the generalization performance of the model by obtaining additional data that can be used for training and thereby prevent overfitting.

Data augmentation in the image-processing field has been primarily performed using a simple and intuitive method. Only a part of a given image has been proven to be useful as additional data by cropping, inverting the image left or right, and even rotating the image to increase the performance [1]. In addition, performance improvements have been shown in natural language processing, e.g., by deleting some words in a given sentence, replacing them with random comments, inserting new words between sentences, and reordering words between words [2]. However, data augmentation in this manner can damage the fundamental meaning of a sentence, unlike correcting an image. For example, the result of flipping a picture of a cat to the left or right is not a new non-cat object. However, if a verb that plays a key role in a sentence is deleted or replaced, the new sentence may have a different meaning from its original purpose or even lose its meaning. Consequently, it may have a different sense from the classified label of the original sentence, which may lower the model's performance.

Moreover, a technique for augmenting data by grafting a pretrained model using a large amount of data has been proposed [3]. However, because the parameters of the

pretrained model are extensive in this method, retraining is costly and time-consuming. In addition, a large amount of GPU resources ae required to train the model, which increases the cost.

In this study, we propose a technique that can efficiently augment data without compromising the fundamental meaning of natural language data using latent variables. The features learned through the encoder are input into a variational autoencoder [4] to understand the fundamental meaning of natural language. Data are augmented by reconstructing the features. Data augmentation is relatively simple compared with pretrained models, without compromising the underlying sense using the latent variables. The experiment confirms that a maximum increase of 5.13% was achieved when our method was applied in the natural language classification task.

2. Related work

Data augmentation is a common strategy used to prevent performance degradation caused by model overfitting or bias [5]. Simple methods such as mixing images or features [6] and generative models [7] have been used in the image domain. In contrast, text data require different approaches, as the syntax and semantics can change when the text is directly modified, as in the abovementioned methods.

A simple method involves changes at the word level; for example, new words are injected into a sentence by replacing one letter with a neighboring letter on the keyboard [8] or randomly deleting, swapping, or adding noise to create new sentences [2]. These methods improve the text classification performance by allowing models to be generalized to words. However, preserving the label and semantics of a sentence is challenging when words are transformed by random deletions and swaps [9].

In addition, pretrained language models (PLMs) are advantageous because they learn rich text expressions from large-scale data; therefore, various studies have conducted augmentation using PLMs. Conditional- BERT consists of a BERT structure modified by altering the segment embedding [3]. In addition to BERT, autoregressive models such as GPT have been used with various methods [5]. However, these methods have certain limitations. Conditional-BERT relies on modifying the embedding of BERT and cannot be generalized to other PLMs without segment embeddings [10]. In addition, these methods are limited in that the same performance cannot be maintained when applying other language models [11].

In this study, we propose a strategy based on a variational autoencoder (VAE). The semantics of the sentences are preserved using the latent variables. Because no significant restrictions exist on using the encoder and decoder structures, implementing the proposed method in a PLM is relatively simple.

3. Method

In this section, we introduce the proposed VAE-based data augmentation technique used to improve the text classification performance.

A. Variational Autoencoder

The autoencoder takes the ith data point $x^{(i)}$ as input, which is a part of dataset X consisting of N sentences, and converts the corresponding latent variable $z^{(i)}$ through the encoder $q_\phi(z|x)$. By providing the generated latent variable $z^{(i)}$ as input for the decoder $p_\theta(x^{(i)}|z)$, the original input $x^{(i)}$ is reconstructed. Here, ϕ and θ are the mean parameters of the encoder and decoder, respectively. To train these parameters, we maximize the following optimization function. That is, the possibility that $x^{(i)}$ can be decoded again in the latent variable z that encodes a given $x^{(i)}$ is maximized.

$$\mathcal{L}_{AE}(\theta, \phi; x^{(i)}) = \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] \quad (1)$$

The VAE extends the above autoencoder structure, by assuming the existence of the posterior probability distribution $q(z|x^{(i)})$ of the latent variable z for a given input data $x^{(i)}$. In addition, learn it to be close to a predetermined prior probability distribution $p_\theta(z)$. For P_z, the standard normal distribution with a mean of 0 and standard deviation of 1 was used. To train $Q(z|x^{(i)})$ and P_z to be near each other, we introduced the Kullback-Leibler (KL) divergence, which represents the difference between the two probability distributions.

$$\begin{aligned}&\mathcal{L}_{VAE}(\theta, \phi; x^{(i)}) = \\ &-D_{KL}\left(q_\phi(z|x^{(i)})||p_\theta(z)\right) \\ &\qquad + \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] \\ &\qquad \leq \log p_\theta(x^{(i)})\end{aligned} \quad (2)$$

The newly defined optimization function corresponds to the lower bound of the probability distribution $p_\theta(x^{(i)})$ in which the decoder generates $x^{(i)}$. To maximize this, the value of the newly added KL divergence term should be reduced as much as possible. That is, this type of function maximizes the possibility that $x^{(i)}$ is generated from the latent variable z, such as in the existing autoencoder, and simultaneously increases the posterior probability distribution of z for $x^{(i)}$. The parameters of the encoder and decoder were trained such that they could be estimated.

Unlike the existing autoencoder, which uses the encoder output value as the latent variable z, the VAE extracts the latent variable from the probability distribution. However, because randomness is reflected in the removal of values

from the probability distribution, the gradient value is not transmitted; therefore, the error backpropagation technique cannot be used. The reparameterization trick is used such that the error backpropagation technique can be used to optimize the VAE model by solving these problems. This method estimates the mean and variance of the posterior probability distribution from the output of the encoder, multiplies the standard deviation by the random value extracted from the standard normal distribution, and adds the mean value as a latent variable. The value obtained in this manner does not differ in terms of distribution from directly extracting a value from the original posterior probability distribution. However, because the value estimated by the model is the average and variance values of the distribution, it can be learned using the error backpropagation technique.

B. Transformer Architecture

The transformer architecture enables parallel data processing using the attention operation based on matrix multiplication, thereby solving the problem the inability to perform parallel operations and the long-term dependency problem of the existing recurrent neural network (RNN) model, including natural language and image processing. Therefore, they are widely used in deep learning [15]. Because each attention operation can be performed in parallel, the transformer architecture exhibits high efficiency when used with a GPU optimized for parallel operation.

C. Adoption of the Pretrained Language Model

PLMs are pretrained in various manners using various types of data. BERT is typically trained using a denoising autoencoding objective called masked language modeling. BERTs trained in this manner perform better with natural language understanding (NLUs) than with natural language generation (NLGs). Unlike this autoencoding objective, an autoregressive objective performs better in NLG [12]. Because each PLM has different strengths and weaknesses, we propose using different PLMs depending on the domain or learning environment.

A model consisting of an encoder and decoder, such as BART [13] and T5 [14], can be used separately for the encoder and decoder in our proposed VAE, or combined with two other models, such as BERT + GPT, which uses BERT for the encoder and GPT for the decoder. Our method is domain-free, because we can use any PLM for the encoder and decoder in the VAE architecture. For example, FinBERT [15] and GPT can be combined when augmenting text in the financial domain. The PLM used in this study was BART.

4. Experiment and discussion

A. Experiment Settings

Eight layers were used for each of the transformer's encoders and decoders and were taken from BART. AdamW [16] was used as the optimizer. The batch size was 32, the learning rate was 1e-5, and learning was performed for ten epochs. BERT [17] was used as the text classification model to verify the performance, and AdamW was used as the optimizer. In the verification task, the batch size was 32, the learning rate was adjusted using the warmup scheduler, and learning was performed for three epochs.

B. Datasets

IMDB, Yelp Full, ProsCons, and MR were used as text classification datasets to verify model performance. The IMDB dataset collects movie reviews, classifies them as positive or negative, and contains 50,000 sentences. The Yelp Full dataset consists of 650,000 sentences and categorizes the evaluation of various places from one to five points. The ProsCons dataset classifies approximately 40,000 product evaluations as positive or negative. The MR dataset classifies approximately 10,000 movie reviews as positive or negative sentences.

Table I lists the results of additional learning using augmented data from the baseline and proposed model. Other learning results used the same amount of augmented data as the proposed model by applying the EDA technique for comparison with existing data augmentation techniques.

Table. I
Comparison of performances according to data augmentation technique. The baseline is the result of the BERT model without data augmentation.

Model / Dataset	**IMDB**	**Yelp_5**
Baseline	91.95%	65.52%
EDA	90.98% (-0.97%p)	67.92% (+2.40%p)
Proposed Model	92.27% (+0.32%p)	68.95% (+3.43%p)
Model / Dataset	**ProsCons**	**MR**
Baseline	93.65%	84.05%
EDA	94.21% (+0.56%p)	84.18% (+0.13%p)
Proposed Model	95.16% (+1.51%p)	84.27% (+0.22%p)

The experiment demonstrated that when additional data were provided using the proposed model on the IMDB dataset, the performance improved by 0.32% compared with when the text classification model was trained using only the original data, and the performance was enhanced by 3.43% on the Yelp Full dataset. Performance improvements of 1.51% on the ProsCons dataset and of 0.22% on the MR dataset were confirmed. The range of performance improvement differs for each dataset, which confirms that the performance improvement was more significant on the dataset with an enormous amount of given data. This ensured that the data could be augmented more effectively because the VAE model generated new data by estimating the distribution of the provided data. Moreover, for data

augmentation using the EDA method as the comparison group, the performance improvement was slight compared with the proposed model; for the IMDB dataset, the performance was lower than before data augmentation. This indicates that data augmentation using EDA methods may be ineffective.

Table. II
Comparison of generated data according to data augmentation technique.

Original Data	I loved this movie since I was 7 and I saw it on the opening day. It was so touching and beautiful. I strongly recommend seeing for all. It's a movie to watch with your family by far.
Augmented Data w/ EDA	I this movie since I was and I saw it on the opening day. It was so touching and beautiful. I strongly recommend seeing disastor. It's some movie to watch with your family by far.
Augmented Data w/ Proposed Model	I loved this movie since I was 9 and I saw it on the opening day. It was so touching and beautiful. I recommend seeing for all. It's a movie to watch with your family by far.

Table II compares the data generated by EDA, which is widely used in natural language data augmentation, and the proposed method. With EDA, the word "loved" was removed, and the word "disaster" was inserted. Although the original text praised the film, a word simply disappeared, and the fundamental meaning was damaged although the word was replaced. However, the proposed method produces data with well-preserved fundamental meanings, with only relatively less critical age-related information being incorrect.

5. Conclusion

In this study, we proposed a VAE-based data augmentation technique to improve the performance of text classification tasks. We compared the training results of the model, including newly generated data, by applying this technique with the proposed method. A performance improvement of up to 5.13% was observed in text classification tasks. In the future, we intend to further study whether the proposed data augmentation technique can be applied to tasks, such as machine translation, as well as manners of improving the current model structure. To this end, by injecting the label information of the original data in the augmentation process, we aim to study a model that better maintains the label and creates suitable sentences.

References

[1] L. Taylor and G. S. Nitschke, "Improving deep learning with generic data augmentation," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1542-1547, 2018.

[2] J. Wei and K. Zou, "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), pp. 6383-6389, 2019.

[3] X. Wu, S. Lv, L. Zang, J. Han, and S. Hu, "Conditional BERT contextual augmentation," in International Conference on Computational Science, pp. 84–95, 2019.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in the 2nd International Conference on Learning Representations (ICLR), 2014.

[5] A. Anaby-Tavor et al., "Do not have enough data? Deep learning to the rescue!" in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, 2020.

[6] S. Yun et al., "Cutmix: regularization strategy to train strong classifiers with localizable features," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019.

[7] C. Chadebec et al., "Data augmentation in high dimensional low sample size setting using a geometry-based variational autoencoder," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[8] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," International Conference on Learning Representations, 2017

[9] S. T. Luu, K. V. Nguyen, and N. L.-T. Nguyen, "Empirical study of text augmentation on social media text in Vietnamese," 2020. arXiv preprint arXiv:2009.12319

[10] V. Kumar, A. Choudhary, and E. Cho, "Data augmentation using pre-trained transformer models," 2020. arXiv preprint arXiv:2003.02245

[11] M. Bayer et al., "Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers," *International Journal of Machine Learning and Cybernetics*, pp. 1-16, 2022.

[12] A. Yang et al., "Enhancing pre-trained language representations with rich knowledge for machine reading comprehension," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019.

[13] M. Lewis et al., "Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880, 2020.

[14] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1-67, 2020.

[15] Z. Liu et al., "Finbert: a pre-trained financial language representation model for financial text mining," in Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021.

[16] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017. arXiv preprint arXiv:1711.05101

[17] J. Devlin et al., "Bert: pre-training of deep bidirectional transformers for language understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers), vol. 1, pp. 4171–4186, 2019.

Biographies

Yuchul Shin received a B.S. degree in Computer Science from Kookmin University, Seoul, South Korea, in 2002 and M.S. degree in image engineering with the Graduate School of Advanced Imaging Science, Multimedia and Film, Chung-Ang University, Seoul, South Korea, in 2020. He is currently an Assistant Professor with the School of Game, Chungkang College of Cultural Industries. His research interests include game development and deep learning.

Kyohoon Jin received a M.S. degree in Imaging Engineering from the Graduate School of Advanced Imaging Science, Multimedia and Film at Chung-Ang University, Korea in 2021. He is currently a Ph.D student with the Imaging Engineering Department at the Graduate School of Advanced Imaging Science, Multimedia and Film at Chung-Ang University. His research interests include deep learning and natural language processing.

Juhwan Choi is currently pursuing a B.S. degree in Electrical and Electronics Engineering at Chung-Ang University. He plans to study graduate school further. His research interests include natural language processing and understanding based on deep-learning approaches.

Junho Lee received a B.S degree in Computer Science from Chung-Ang University, Seoul, South Korea, in 2020. He is currently pursuing a M.S. degree with the Department of Artificial Intelligence, Chung-Ang University, Seoul. His interests include deep learning and natural language processing.

SooJin Jang is currently a Ph.D. student at Graduate School of Advanced Imaging Science, Multimedia and Film at Chung-Ang University. She received a B.S. degree in Information and Communication Engineering from Sunmoon University in 2018, and her M.S. degree in Digital Imaging from Chung-Ang University in 2020. Her research interests include deep learning and computer vision.

YoungBin Kim is currently an assistant professor with the Graduate School of Advanced Imaging Science, Multimedia and Film at Chung-Ang University. He received his B.S. and M.S. degrees in Computer Science from Korea University in 2010 and 2012, respectively. He completed a Ph.D. in Visual Information Processing from Korea University in 2017. From August 2017 to February 2018, he has been a principal research engineer at Linewalks. His current research interests include data mining and deep learning.