Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 301–309

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

A short survey for Vietnamese micro-text augmentation techniques

Huu-Thanh Duong*, Vinh Truong Hoang

Faculty of Information Technology, Ho Chi Minh City Open University,
35-37 Ho Hao Hon, Co Giang Ward, District 1, Ho Chi Minh City and 70000, Vietnam

Abstract

The requisites of a powered-AI system is to have a big enough annotated data. Lack of the datasets is a big challenge to obtain the robustness of the predictive models so that it can broaden the AI ideas to various domains. The predictive models are less generalized and prone to overfit. Although the resources for Vietnamese have been investigated more and more, it has still been a low-resources language which is the biggest barrier in order to leverage the robustness of the AI applications. Building the datasets consumes so much time and money. This paper presents the text augmentation to generate the new annotated training data without user's intervention. This paper has summarized the potential methods, especially for the cross-languages methods to enhance the data, analyzed and evaluated the advantages and disadvantages of each method to apply to Vietnamese language processing. The synthetic presentation shows text augmentation has gained competitive performances and helped to save the time and money to build the data.

Keywords: text augmentation; limited training data; embedding mixup; imbalance data;

1. Introduction

Nowadays, Artificial Intelligence (AI) automatizes the handcrafted works, increases the labour performances, and is able to predict based on its learned stuff. In order to gain robustness of the models and widely apply AI-ideas in various domains, the requisite is to have the big enough annotated data, especially for the modern models such as deep learning (DL) models which inherently required a large amount of the annotated data to struggle with the measurements of the generalization and confidence of the predictive models. The rapid growth of computing power is mainly to motivate the development of DL and has achieved state-of-the-art performances in many tasks of computer vision and natural language processing (NLP) in recent years. The remaining largest barrier is a huge amount of annotated data to gain reliable models.

* Corresponding author. Tel.: +84-0965573460.
E-mail address: thanh.dh@ou.edu.vn

1877-0509
10.1016/j.procs.2023.01.012

The datasets are unavailable and take much time and labor-human to build. Thus, the researchers have instantly studied the alternative ways to leverage the available data without manual annotation. Data augmentation which is the promising solution investigating in recent years aims to construct synthetic training data from the available small data. This has attracted many related studies in NLP and widely been used to enrich the training data leading to enhance the generalization of the training models, avoid overfitting, and also resolve some problems in NLP such as imbalance data of the labels or unknown words. According to Abonizio et al., 2021 [1], text augmentation can be divided into two primary approaches: sentence manipulation is done on raw text and embedding manipulation is done on the representations of embedding space.

This paper investigates the cross-languages methods to gain synthetic samples in order to enhance the training data and boost the accuracy of the predictive model under small data. The main purpose is to leverage the existing resources to auto-generate new annotated data and present the potential methods for several AI-related problems in Vietnamese. The paper discusses these methods in micro-text contexts such as sentiment analysis problems. Sentiment analysis is to detect the polarities (positive or negative) of the reviews or comments on the social media, blogs or e-commerce websites without user intervention. This problem is widely utilized in many domains, specially in e-commerce systems or any systems that take care of the users' feedback so that they can understand the customers' expectations and replies more easily, and also adjust the strategies and supports to make the decisions effectively. The annotated data which trains the predictive models are unavailable in each domain and it is expensive and takes a long time to build them. This is really the big challenge to gain effective models to apply these AI ideas in the real world. Text augmentation emerges as the lifesaver in this case, this enriches the limited training data to automatically generate the new training samples.

This paper aims to summarize the state-of-the-art methods to augment the data under limited training data, analyze the advantages and disadvantages of these methods to determine the approaches for Vietnamese sentiment analysis in particular, text classification in general. This may motivate many next proposals to enhance the training data, especially for the low-resources languages.

The rest of this paper is organized as follows: Sect. 2 presents the methods to augment the synthetic text by directly manipulating the raw text, Sect. 3 presents the embedding-based method to generate the synthetic representations by manipulating on embedding space and the final Sect. 4 is our conclusions and future works.

2. Text-based Augmentation

Text-based augmentation is directly done with the text samples to generate new samples such as synonym substitution based on the wordnet, similar substitution based on the pre-trained word embedding models, sentence shuffling, back translation, syntax-tree transformation, text generation.

2.1. Simple Methods

Wei et al., 2019 [2] presented the easy methods to generate the new samples including synonym replacement, random insertion, random deletion and random swap (calling EDA methods). Firstly, synonym replacement will randomly replace *n* words of the sentence not being stop words by one of their synonym words. For example "tôi **thích** sản_phẩm đó, thiết_kế thật tuyệt_vời" (I like that product, the design is so great), the word "thích" (like) has a set of the synonym words ["yêu", "mến", "chuộng", "ưa", "ưa_chuộng", "yêu_mến", "mến_phục", "mến_mộ", "hâm_mộ", "ngưỡng_mộ"] [1], so it can generate some following new sentences.

- "tôi **ưa** sản_phẩm đó, thiết_kế thật tuyệt_vời"
- "tôi **chuộng** sản_phẩm đó, thiết_kế thật tuyệt_vời"
- "tôi **yêu** sản_phẩm đó, thiết_kế thật tuyệt_vời"

[1] Using a Vietnamese wordnet from https://github.com/zeloru/vietnamese-wordnet

This method is actually simple and easy to implement, but needs an effective thesaurus. Thus, this is only handy for rich-resources languages such as English. Moreover, it will meet some challenges for this approach such as selecting a replacement word in text will affect the quality of the generated text which depends on the context, some formative words are replaced leading to change the meaning of the original data. Similarly, selecting one of its synonym words for replacing may be unsuitable for the context of the original data. For instance, using the word "hâm_mộ" (admire) to replace the word "thích" for the above case is less suitable than using the word "chuộng" in this context. The *tf*×*idf* value is the classic score used widely in NLP; it can be considered to select the replacement word in the sentence. The words usually are informative if this score is low, so replacing them will be less influenced by the original meaning. Yu et al., 2021 [3] only selected the words impacting the label prediction based on the attention score.

Random insertion will insert *n* times in inserting a synonym word of a word into a certain position. For example the sentence "tôi **thích** sản_phẩm đó", some new sentences will gain as follows: "tôi **thích** sản_phẩm **yêu_mến** đó"; "tôi **thích** sản_phẩm đó **yêu**"; "**yêu_mến** tôi **thích** sản_phẩm đó". Similarly, random swap will swap *n* times two random words of the sentence and random deletion will remove a random word of the sentence with a probability *p*. Random deletion can sometimes cause loss of information since it may delete the informative words.

These methods are easy to understand, low cost to implement and independent of the languages. Although they can lead to some meaningless texts, they actually improve the performance of the predictive models. The authors of EDA methods gained the robust models when they conducted the experiments for varying training set size with CNN and RNN models. It conducted some experiments in these approaches in Vietnamese ([4]) and also obtained better results. However, the authors advised to choose the suitable *n* value based on the size of the original data. It's prone to overfitting for the small training size and maybe unhelpful for the larger training size since it tends to generalize properly from real data. Thus, the available training set is larger, the *n* value is smaller, the authors recommended choosing the *n* values as [16, 8, 4] corresponding to the training size as [500, 2000, 5000+]. Karimi, Akbar et al., 2021 [5] proposed the easier method (AEDA) based EDA methods. They simply inserted the punctuation marks into the original text at random. Their approach generated a *n*-random value between 1 and one-third of the length of the sentence, then randomly selected *n* positions in the sentence and inserted the punctuation marks in ['.', ',', '!', '?', ';', ':'] into those positions. The authors also used CNN and RNN models to perform the experiments for both EDA and AEDA and showed the AEDA method outperformed the performances compared to EDA methods.

One more simple method is sentence shuffling. This method shuffles among sentences of the text in the same labels to obtain new samples. For example the sentences "Máy đó đẹp quá. Tôi rất thích." (this phone is so beautiful. I like it) and "Camera tuyệt_vời. Thiết kế cũng xuất_sắc." (The camera is great. The design is also excellent), it can gain some following new samples:

- "Máy đó đẹp quá. **Thiết kế cũng xuất_sắc**."
- "Camera tuyệt_vời. **Tôi rất thích**."

Yoon [6] proposed to perform a mixing operation on the raw input. The synthetic sentence gained by mixing a span of two original sentences and kept more tokens related to the prediction based on saliency information. Their performances outperformed the hidden mixup on a wide range of benchmark datasets.

2.2. Word Embedding

Word embedding is the way to convert the text into a vector, this is an essential tool for machine learning and deep learning models for NLP. Ideally, the words have similar meanings being near each other in vector spaces. It can take advantage of this clue to generate new samples by replacing the words in the sentences by their similar words based on the contextual space. Many pre-trained models have been invented such as Word2vec, Glove. These ones help the machine understand the languages better. In the digital era, it can easily reach a variety of internet resources. Thus, this method can be used in many domains and languages. This also reduces the cost to build the relevant resources such as the thesaurus or wordnets. However, it is a black-box approach and sometimes the opposite words are also near each other in the vector space and tend to the generated sentence having the opposite meaning. This may be the cause to affect the performances of the models. For instance, based in our pre-trained model is built by Word2vec in Vietnamese [7], the word "đẹp" (beautiful) has the similar words ["bắt_mắt", "sang_trọng", "đẹp_mắt", "sáng_sủa", "sạch_sẽ", "lạ_mắt", "lung_linh", "hoành_tráng", "thoáng", "thoáng_mát"] in descending scores. The

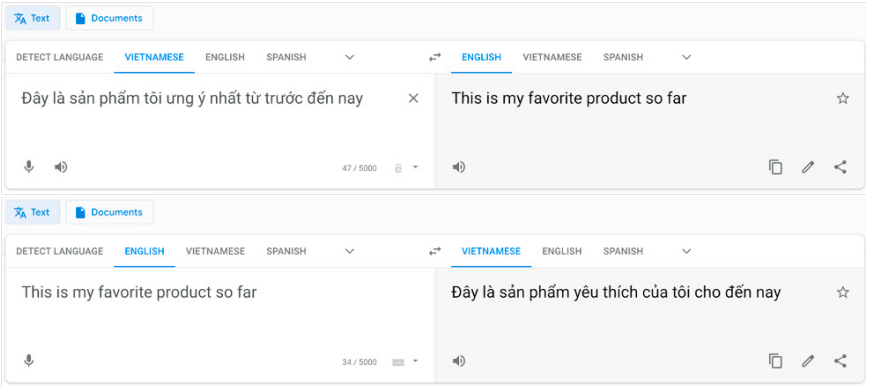

Figure 1. Translates the sentence from Vietnamese to English and Back-translates the translated sentence to Vietnamese.

sentence "điện_thoại rất đẹp" (the phone is beautiful) will generate the following sentences: "điện_thoại rất **bắt_mắt**" or "điện_thoại rất **sang_trọng**".

Another spotlight pre-trained model is BERT (Bidirectional Encoder Representation from Transformer), BERT is the new language representation model being introduced by Google gaining the state-of-the-art performances for many NLP tasks such as sentiment analysis, question answering, sentence inference, etc. This has appeared in Vietnamese, namely PhoBERT. This is a pre-trained language model for Vietnamese presented by Nguyen, Dat Quoc and Tuan Nguyen, Anh [8]. This achieves the state-of-the-art performances for NLP tasks such as part-of-speech tagging, named entity recognition, language inference.

Masked Language Model (MLM) is used to fill a masked blank in the sentence. This a clue to generate the new samples by using the pre-trained model with PhoBERT. For instance, the sentence "đây là sản_phẩm tuyệt_vời trong tầm giá", it can select one of the words in the sentence and feed to MLM to obtain the new sentence. It assumes replacing the word "tuyệt_vời" and feeds the sentence "đây là sản_phẩm **<mask>** trong tầm giá" to MLM.

2.3. Back Translation

Back translation is firstly used to improve the performance of machine translation. This method leverages the power of the machine translators and abundance of the rich-resources languages as the intermediate language to generate the new samples. The first step will translate the text from the source language to the intermediate language, and then it will conduct to back-translate the translated text to the source language, this text will usually be totally exact with the original text. Hence, it gains the new samples. For instance, the sentence "Đây là sản phẩm tôi ưng ý nhất từ trước đến nay" is translated into English as "This is favorite product so far" by Google Translation, this is back-translated into Vietnamese being "Đây là sản phẩm yêu thích của tôi cho đến nay" (see Figure 1), this is the augmented sample.

This method preserves the meaning of the original text. However, this needs an effective translator and the quality of the augmented text depends on the grammar structure of the original text. If the text has a lot of noise information such as wrong spelling, it may tend to the meaning of the augmented text being different from the original meaning. In common, this is still an effective method and can apply to many languages. Our experiments in [4] also improved the accuracy of the predictive models in Vietnamese.

2.4. Syntax-tree Transformation

Syntax-tree transformation uses some syntactic grammar to transform the syntax tree of the original text to another tree to generate new sentences. This method relies on some rules to work on the syntax tree, such as changing from active voice to passive voice. For instance, the sentence "Tôi rất thích sản phẩm đó" (I like that product so much), Figure 2 is the syntax tree of this sentence, it can change to passive voice for the new sentence "sản phẩm đó được thích bởi tôi". This significantly improves the accuracy of the predictive model, but it is expensive in computation, especially for Vietnamese inherently having complex in sentence structures.

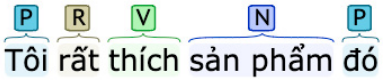

Figure 2. The syntax tree of the sentence is parsed by underthesea [2] (P: pronoun, N=noun, V=verb, R=adverb)

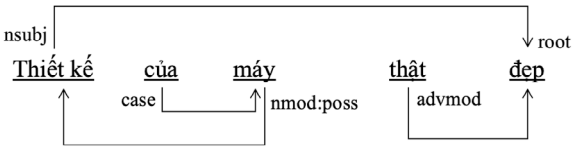

Figure 3. The dependency tree of the sentence (advmod=adverbial modifier, nsubj=nominal topic, case=case-marking, nmod:poss=possessive nominal modifier)

Dependency tree refers to a way to determine the grammatical structure of the sentence based on the dependencies among its phrases. Sahin et al. [9] borrowed the ideas from image classification, including rotating and cropping. They used the dependency tree to remove dependency link (sentence cropping) and move the tree fragment around the root (sentence rotating) for new data. For instance, the sentence "Thiết kế của máy thật đẹp" (The design of that phone is so beautiful) has the dependency tree as Figure 3 by using the dependency parsing of underthesea. Some sentences gains by Sentence cropping: "Thiết kế của máy đẹp"; "thật đẹp" and others by Sentence rotating: "thật đẹp thiết kế của máy"; "đẹp thiết kế của máy thật".

3. Embedding-based Augmentation

Embedding-based augmentation is done on representation vectors of the text instead of the text samples. This approach has frequently been applied in deep learning models which achieve state-of-the-art performances in NLP. Mixup is the latest method for data augmentation techniques by linearly interpolating the input samples and their corresponding labels. The synthetic samples are fed into the models for training to minimize the loss function. This method has been initiated in image classification by Zhang et al. 2017 [10] by mixing the input images and significantly improved the predictive results of the state-of-the-art networks and gained superior performances. The linear mixup method will mix the input embeddings and their corresponding labels, the mixup-ratio coefficient is a scalar value based on Beta distribution.

$$\hat{x} = \lambda x_i + (1 - \lambda) x_j \tag{1}$$

$$\hat{y} = \lambda y_i + (1 - \lambda) y_j \tag{2}$$

Where x_i, x_j are the raw input vectors, y_i, y_j are the corresponding one-hot labels and λ is the mixup-ratio having the value between [0, 1] based on Beta distribution.

This idea has equipped NLP working on word embedding space and also gained promising performances. There are many variants of mixing the text such as embedding mixup, hidden state mixup, sentence mixup. Embedding mixup is performed immediately after word embedding before feeding to the training networks, hidden state mixup is performed prior to the last fully connected layer and sentence mixup is performed before softmax. Mixup approaches only work on embedding space, so it can be applied for any language. The following sections discuss the mixup-based approaches for NLP.

3.1. Linear Mixup

Guo et al. 2019 [11] proposed wordMixup and senMixup based on Zhang's idea [10]. The wordMixup method will linearly interpolate word embedding in the sentence before feeding them to the training networks (see Figure 4). All sentences are zero padded to the same length and the vector of each word in the sentence will be interpolated. A pair of the t^{th} word of two input sentences, is linearly interpolated as below to gain the new sample ($\widetilde{B}_t^{ij}$, $\widetilde{y}^{ij}$):

$$\widetilde{B}_t^{ij} = \lambda B_t^i + (1 - \lambda) B_t^j \tag{3}$$

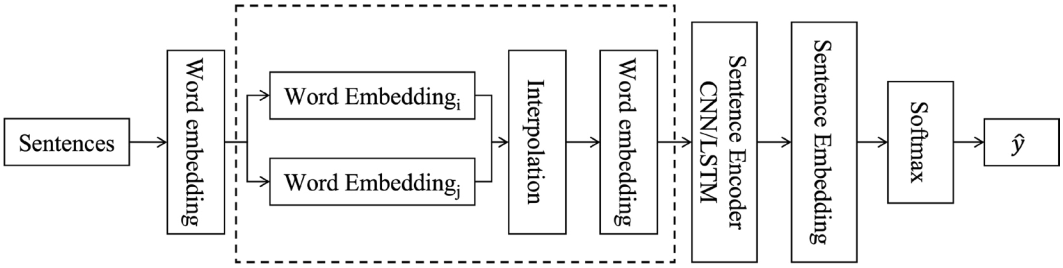

Figure 4. The process of wordMixup

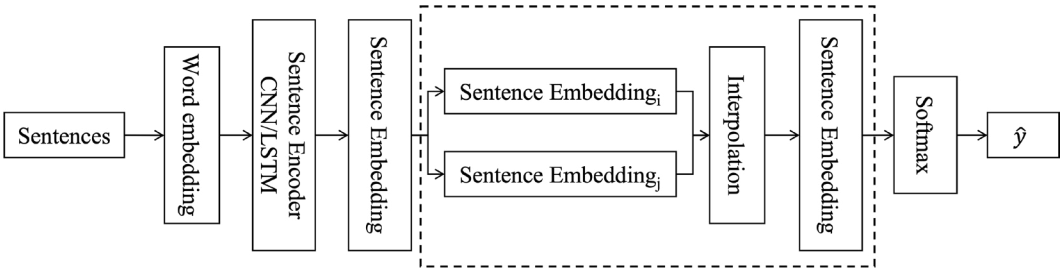

Figure 5. The process of senMixup

$$\widetilde{y}^{ij} = \lambda y^i + (1-\lambda) y^j \tag{4}$$

Where B^i, B^j are the matrix embeddings of the i^{th} and j^{th} sentences, each row of each matrix is an embedding of each word in the corresponding sentences, y^i, y^j is the labels respectively.

The senMixup will mix a pair of sentence embeddings on the final hidden layer before passing them to the softmax layer (see Figure 5). The hidden embeddings of two sentences are generated by the encoders CNN and LSTM. These embeddings will be linearly interpolated to gain new samples as below. Where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the corresponding vectors of two sentences from the encoders, y_i and y_j are their labels respectively. These methods will generate the soft labels being out of the labels of the original training data. The authors empirically show these approaches improve significantly performances on CNN and LSTM models.

$$\widetilde{x} = \lambda x_i + (1-\lambda) x_j \tag{5}$$

$$\widetilde{y} = \lambda y_i + (1-\lambda) y_j \tag{6}$$

3.2. Non-linear Mixup

The synthetic samples of the conventional mixup reside in the straight line between two input representations of the sentences. According to Guo, 2020 [12], it significantly limits the space of the synthetic samples since this requires a convex combination of the inputs and the modeling targets. Thus, they also proposed the non-linear mixup. The mixing weights were randomly gotten towards the Gaussian distribution. Their experiments significantly improve upon the conventional mixup. Unlike linear mixup, this uses the same scalar value (λ) to interpolate a pair of the representations, non-linear mixup creates a individual lambda based on Beta distribution for each dimension of a word embedding, this is similar to wordMixup for non-linear interpolation. For B^i, B^j are the matrix embeddings of the i^{th} and j^{th} sentences, the synthetic samples are computed by the following formula:

$$\widetilde{B}^{ij} = \Lambda \circ B^i + (1-\Lambda) \circ B^j \tag{7}$$

The mixing ratio is the matrix Λ, each element of this matrix is estimated by Beta distribution, $\circ$ symbol is a Hadamard product. One more problem of the non-linear mixup is the synthetic soft labels since the mixing ratio

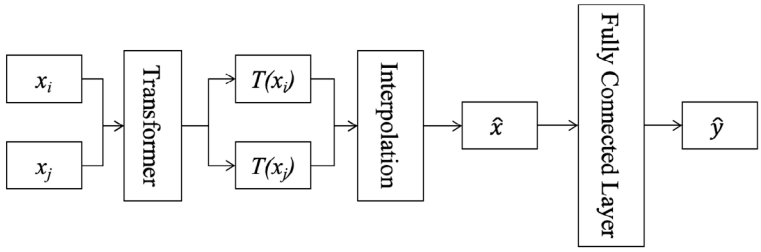

Figure 6. The process of transformers mixup.

is a matrix, so it cannot compute directly on one-hot labels of pair sentences. The authors resolved by using label embedding proposed by Bengio [13], 2010 so that The one-hot labels are encoded to the one-hot vectors. Shortly, the experiments performed on the benchmark sentences show this approach outperformed the conventional mixup since the input space of these synthetic samples is much larger than these ones are created by linear mixup. Thus, they provide better regularization for the training model.

3.3. Transformer Mixup

The traditional methods, the word embeddings gained from the deep neural networks (such as CNN or LSTM), Sun et al. [14] used the transformer-based pretrained model to learn the text features. The authors stacked the mixup layer over the final hidden layer of the transformer-based model before feeding to the fully connected layer to get the output (see Figure 6). Their experiments achieve good performances on the GLUE benchmark.

Yin et al. [15] explored two drawbacks of the mixup methods, namely it generated new points covering pretty limited region in the entire space of the mini-batch and slowed down the training phase because of taking time to generate new points for each a pair of the original points. They proposed the BatchMixup approach by non-linearly interpolating all the samples in the same mini-batch on the level of hidden states generated by RoBERTa instead of a pair of the data points. The mixed data points can cover better the space expressed by the mini batch. Their experiments showed this approach improved significantly performances compared to the conventional mixup.

Table 1. The comparison of the methods of the text augmentation.

	Cross languages	Additional Resources	The meaning of the sentences	Computational cost	Implementation
Synonym replacement	No	Thesaurus	Can retain the meanings	Low	Easy
Random Insertion	No	Thesaurus	May not make senses	Low	Easy
Random Deletion	Yes	No needs	May not make senses	Low	Easy
Random Swap Yes	No needs	May not make senses	Low	Easy	
Sentence Shuffling	Yes	No needs	May not make senses	Low	Easy
AEDA	Yes No needs	Can retain the meanings	Low	Easy	
Back translation	Yes	Translators	Can retain the meanings	Low	Easy
Syntax-tree transformation	No	Syntax parser	Can retain the meanings	High	Complex
Contextual replacement	Yes	Pretrained models	May retain the meanings	Medium	Medium
MLM-based replacement	Yes	Pretrained models with BERT	May retain the meanings	High	Medium
Linear mixup	Yes	No needs	Working in vector space	High	Complex
Non-linear mixup	Yes	No needs	Working in vector space	High	Complex
Transformer mixup	No	Transformer-based pretrained models	Working in vector space	High	Complex

4. Conclusions

Lack of the annotated data tends to the poor performances and the less reliable prediction of the AI systems. In order to gain the big enough annotated data takes a long time, labor-intensive humans and cost a lot. This is a big challenge for low-resources languages like Vietnamese and inspires the AI ideas to many domains. This paper has discussed the potential text augmentation for Vietnamese micro text, selected the cross-languages methods, analyzed the strong and weak points of each method (Table 1 is the short comparison among the methods). The related experiments show text augmentation boost the performances of the predictive models and hope this can motivate more effective research of AI applications for Vietnamese on many various domains. Moreover, text augmentation is helpful for resolving imbalance problems in training data.

In future, it plans to investigate embedding-based approaches, especially for embedding mixup approaches. The abundance of internet resources and the strong development of the pre-trained models show it is feasible to leverage to enrich the annotated data. Thus, this is the most potential method which can apply to the low-resources languages such as Vietnamese so that it increases the generalization of the models and avoids overfitting for deep learning models which is inherently a hot trend to understand the languages better. We will collect to enrich more data for Vietnamese, also exploit the existing Vietnamese pretrained models such as PhoBERT [3] to enhance the training data in order to solve more complex relevant problems in NLP.

Acknowledgment

The study was supported by The Youth Incubator for Science and Technology Programe, managed by Youth Development Science and Technology Center - Ho Chi Minh Communist Youth Union and Department of Science and Technology of Ho Chi Minh City, the contract number is 06/2021/HĐ-KHCNT-VƯ.

References

[1] H. Q. Abonizio, E. C. Paraiso, and S. Barbon Junior, "Toward text data augmentation for sentiment analysis," pp. 1–1. [Online]. Available: https://ieeexplore.ieee.org/document/9543519/

[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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6382–6388. [Online]. Available: https://aclanthology.org/D19-1670

[3] Y. J. Yu, S. J. Yoon, S. Y. Jun, and J. W. Kim, "TABAS: Text augmentation based on attention score for text classification model," p. S2405959521001454. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405959521001454

[4] H.-T. Duong and T.-A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Computational Social Networks*, vol. 8, no. 1, jan 2021. [Online]. Available: https://doi.org/10.1186%2Fs40649-020-00080-x

[5] A. Karimi, L. Rossi, and A. Prati, "AEDA: An easier data augmentation technique for text classification," in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2748–2754. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.234

[6] S. Yoon, G. Kim, and K. Park, "SSMix: Saliency-based span mixup for text classification." [Online]. Available: http://arxiv.org/abs/2106.08062

[7] H.-T. Duong and T.-K. Tran, "The impacts of the contextual substitutions in vietnamese micro-text augmentation," in *Nature of Computation and Communication*, P. Cong Vinh and N. Huu Nhan, Eds. Cham: Springer International Publishing, 2021, pp. 32–39.

[8] D. Q. Nguyen and A. Tuan Nguyen, "PhoBERT: Pre-trained language models for vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, pp. 1037–1042. [Online]. Available: https://www.aclweb.org/anthology/2020.findings-emnlp.92

[9] G. G. Şahin and M. Steedman, "Data augmentation via dependency tree morphing for low-resource languages," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 5004–5009. [Online]. Available: https://aclanthology.org/D18-1545

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[11] H. Guo, Y. Mao, and R. Zhang, "Augmenting data with mixup for sentence classification: An empirical study," number: arXiv:1905.08941. [Online]. Available: http://arxiv.org/abs/1905.08941

[3] https://github.com/VinAIResearch/PhoBERT

[12] H. Guo, "Nonlinear mixup: Out-of-manifold data augmentation for text classification," vol. 34, no. 4, pp. 4044–4051. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/5822

[13] S. Bengio, J. Weston, and D. Grangier, "Label embedding trees for large multi-class tasks," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010. [Online]. Available: https://proceedings.neurips.cc/paper/2010/file/06138bc5af6023646ede0e1f7c1eac75-Paper.pdf

[14] L. Sun, C. Xia, W. Yin, T. Liang, P. Yu, and L. He, "Mixup-transformer: Dynamic data augmentation for NLP tasks," in *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, pp. 3436–3440. [Online]. Available: https://www.aclweb.org/anthology/2020.coling-main.305

[15] W. Yin, H. Wang, J. Qu, and C. Xiong, "BatchMixup: Improving training by interpolating hidden states of the entire mini-batch," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, pp. 4908–4912. [Online]. Available: https://aclanthology.org/2021.findings-acl.434