A Combination of BERT and Transformer for Vietnamese Spelling Correction

Trung Hieu Ngo(✉), Ham Duong Tran, Tin Huynh, and Kiem Hoang

The Saigon International University, Ho Chi Minh City, Vietnam
{ngotrunghieu,tranhamduong,huynhngoctin,hoangkiem}@siu.edu.vn

Abstract. Recently, many studies have shown the efficiency of using **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (BERT) in various Natural Language Processing (NLP) tasks. Specifically, English spelling correction task that uses Encoder-Decoder architecture and takes advantage of BERT has achieved state-of-the-art result. However, to our knowledge, there is no implementation in Vietnamese yet. Therefore, in this study, a combination of Transformer architecture (state-of-the-art for Encoder-Decoder model) and BERT was proposed to deal with Vietnamese spelling correction. The experiment results have shown that our model outperforms other approaches as well as the Google Docs Spell Checking tool, achieves an 86.24 BLEU score on this task.

Keywords: Vietnamese spelling correction · BERT · Transformer

1 Introduction

A spelling error is a word written in a wrong spelling standard, including various forms: homophone, acronym, uppercase, or fairly the phenomenon of wrong-written words. Usually, there are many groups of origination causing the spelling errors to happen in Vietnamese: typing, semantic confusion, local pronunciation, rules, and standards in the written text, not mastery in grammar and influence of social network language, etc. [29].

Spelling correction is a Natural Language Processing task that focuses on correcting spelling errors in text or a document. The spelling correction task keeps a critical role in enhancing the user typing experience and guarantees the information integrity of Vietnamese. Besides, one primary application is the ability to incorporate with other tasks. For example, when using the spelling correction attached to the last phase of the Scene Text Detection / Optical Character Recognition (OCR) task, the results are improved significantly [1,11]. Consider the chatbot task, if spelling correction is applied to preprocess user inputs, the chatbot will have better accuracy and performance in understanding the user requests [27].

Frequently, the spelling correction can be divided into two steps, including spell checking and spell correcting [7]. In the first phase, mistakes are investigated if there are any in the given input and then try to transform the wrong words into corrected words in the second phase.

N. T. Nguyen et al. (Eds.): ACIIDS 2022, LNAI 13757, pp. 545–558, 2022.
https://doi.org/10.1007/978-3-031-21743-2_43

Contrary to English and other languages, the Vietnamese possess up to six complex diacritic marks and uses them as a discrimination sign. Therefore, a word that combines with different diacritic marks can create up to six written forms, and each of them also has independent meaning and usage. For instance, the word "ma" (ghost) can be written in 5 more ways with 5 different diacritic marks: "má" (mother), "má" (nevertheless), "mả" (tomb), "mã" (code), "mạ" (rice seedlings). All the originations and elements described above have made the Vietnamese spelling correction problem a very challenging task.

There are many initial approaches to the Vietnamese Spelling Correction task that has been carried out such as applying rule-based methods [10], using edit-distance Algorithm [20], collating with dictionaries, using n-gram/big-gram language model [19], etc. However, most of these approaches neither adapted to out-of-vocabulary words nor did they take the contextualized word embeddings into account. In order to deal with these gaps, many deep learning models using Recurrent Neural Network (RNN) or Long Short-term Memory (LSTM) networks have been proposed and achieved impressive performance [21].

Recently, spelling correction studies that took advantage of the Encoder-Decoder model have attracted much attention and achieved state-of-the-art in the English spelling correction task [14,30]. This is a novel approach, which is notably potential because of the optimal utilization of the parallelism calculation ability and the strength of powerful pre-trained language models. One of the most attention is the usage of the Transformer architecture [28] with the language model known as BERT [2]. Despite its success in English [31], there is still no implementation in Vietnamese that can be used in practice. Therefore, this paper aims to apply these architectures and techniques to improve the performance of correcting Vietnamese spelling errors. The experiment results show that the proposed solution achieves considerable efficiency and is able to integrate with practical services. The main contributions of this study could be summarized as follows:

- Applying the Transformer architecture and leveraging the pre-trained BERT to provide a solution to the Vietnamese spelling correction problem.
- Constructing a large and creditable dataset based on the most common practical Vietnamese spelling errors. The evaluation dataset is published for the Vietnamese NLP community using in related works.

The remainder of this paper is organized as follows. In Sect. 2, related works are presented and discussed. Section 3 describes the proposed methods in detail. The dataset, experimental results, and discussion are provided in Sect. 4. Section 5 summarizes, concludes and gives future orientation.

2 Related Works

Spelling correction is not a new problem in NLP tasks. Earlier there have been many approaches for this problem, from straightforward approach using probability, such as implementing the Naive Bayes algorithm (Peter Norvig[1]). The

[1] https://norvig.com/spell-correct.html.

large N-gam-based language modeling approach of both left and right side has improved the performance of spelling correction tasks [19]. After training with a large corpus, this model can predict the probabilities of multiple N-gram candidates for correcting words. Large N-gram LM is a pure probability approach. It expects high memory resources to store all pre-calculated probabilities of N-gram pairs and can not handle a not-pre-trained error, which leads to all of the probability of N-gram pairs to zero.

The advantages of contextual embedding in word presentation model, likes Word2Vec [16], Glove [24], etc., is being taken into the spelling correction task [4]. An edit-distance algorithm generates the candidates, then each candidate's score is calculated by the cosine similarity between the candidate vector and the target word vector, the highest score ranking candidate will be selected. This method has shown significant results in the spelling correction task and is suitable for many languages, especially in Vietnamese, English, etc. On the other hand, this approach requires many resources to represent the rich context embeddings of a language accurately. Also, out-of-vocabulary (OOV) is a large major problem to the ranking system.

Another approach to using deep learning has been developed through the use of LSTM network [18]. A LSTM network [8] is constructed that encodes the input sequence and then decodes it to the expected correct output sequence, respectively. The accuracy of their model makes a significant gap compared to the current state-of-the-art model [19]. Studies have reported that spelling correction can be beneficial from Encoder-Decoder architecture [9,25]. A state-of-the-art approach in English is implemented by the Encoder-Decoder architecture [26] and also makes use of the powerful pretrained BERT model [9]. They first fine-tuned the BERT model and then used its last hidden presentation output as additional features to the an error correction model, which is a customized Transformers [28] architecture. A similar method for Vietnamese grammatical error correction using OpenNMT framework [13] instead of the Transformer architecture [25]. This method, respectively, depends on using the Microsoft Office spelling tool to check and detect the incorrect text before the correction step.

Through previous related works, deep learning approaches to spelling correction are our focus. The approach is receiving much concern, gains state-of-the-art performance is the Encoder-Decoder architecture with prominent pre-trained MLM. Both well-known pre-trained Google Multilingual [2] and vinai/phobert [17] are used to extract hidden presentations and implement the transformer architecture into a specific Vietnamese spelling correction task.

3 Our Approach

3.1 Introduction to the Vietnamese Language

This section briefly presents the characteristics and differences from English of the Vietnamese language. Unlike neighbor countries, the Vietnamese does not use hieroglyphic letters, but a modified Latin (Roman) alphabet. The Vietnamese alphabet uses 29 letters, unlike the English alphabet, it does not use 4 letters 'w', 'f', 'j', 'z' and uses 6 more vowel letters (with special mark): 'ă', 'â', 'ê', 'ô', 'ơ', 'ê', and the letter 'đ' [3,6]. Along with the above 6 types of diacritics, it forms up to 67 separate letters (nearly 3 times larger than the number of letters in English). Therefore, spelling mistakes are much more common in Vietnamese than in English.

3.2 Analyzing of Vietnamese Common Spelling Error

In this section, the concept of common error type in the Vietnamese language are presented. Due to the lacks of scientific public research or national survey constructed on this topic, various types of Vietnamese error type from previous related work [18–20] are summarized and divided them into six groups:

- **Abbreviation:** There are a wide variety of abbreviation for common words in Vietnamese writing. Despite its convenience, this style of writing may raise misunderstanding, make the writing less formal and not accepted by most people. To determine this error cases, a list of most common abbreviation substitutions in Vietnamese is compiled from the Internet.
- **Region:** The region error type is the most complicated type to analyze owing to its variety of happening contexts. The region error type comes from different region pronunciation across the Vietnam territory. When people tend to write a word the same way they pronounce it, this error occurs. Many adults may mistake this type of error if not a native speaker or do not have enough knowledge of the Vietnamese language. An incorrect word with region type stands alone, may still have meaning. Some examples of region error type are described in Table 1.

Table 1. Some examples of region error type

Original	Usually mistake for	Original	Usually mistake for
ch-	tr-	-nh	-n
tr-	ch	c-	k-
-n	-ng	k-	c-
-ng	-n	ngh-	ng-
g-	gi-	gi-	g-
...	...	...	...

- **Teencode:** Teencode (or Teen-code) is a method of writing used by teenagers on social media or through messaging. Those teenagers put words into special encryption so the adults can not understand.
- **Telex:** Telex is a convention for encoding Vietnamese text in plain ASCII characters, used initially for transmitting Vietnamese text over telex systems. Forgetting to turn on the language encoder or entering the wrong Vietnamese Telex rules leads to this type of error.

- **Fat Finger** Fat Finger, also known as the clumsy finger, means when typing through a cell phone or computer keyboard, the user's finger mistypes the surrounding key instead of the target key, causing the wrong words.
- **Edit Distance** Edit Distance is a pseudo error generation strategy in which several characters equal to a 'distance' to the original are randomly replaces. Although this error rarely happens logically, a low percentage amount is still generated in our data set.

For the convenience of observation, a list of examples corresponding to the type of error is presented in Table 2.

Table 2. A summary about error types

Error type	When it happends	Examples	
		Correct	Incorrect
Abbreviation	To make writing faster and more convenient, people use abbreviation instead of full words.	Không (No) Mọi người (Everyone) Bình thường (Normal)	**Kg** **Mn** **Bt**
TeenTeencode	When teenagers try to encode their text in order to hide information.	Ví dụ (Example) Chồng (Husband) Điện thoại (Phone)	V**j** d**u** **Cho'ng** **Dj3n tk04j**
Fat-Finger	While typing with a virtual keyboard, the user's finger mistypes the surroundings instead of the target key.	Xin chào (Hello) Trường học (School) Điện thoại (Phone)	**Xim** chào Trường h**ịc** Điện th**ọak**
Telex	Forgetting to turn on the language encoder or entering the wrong Vietnamese Telex rules.	Xin chào (Hello) Trường học (School) Điện thoại (Phone)	Xin ch**afo** Tr**uowng** h**ojc** **Ddieejn thoaji**
Region	Different region pronunciation across the Vietnam (as people tend to write the same as as they pronounce).	Tranh (Painting) Lạnh (Cold) Nghỉ (Rest)	**Ch**anh (Lemon) **N**ạnh Ngh**ĩ**(Think)
Edit-Distance	A common pseudo misspelling error generation strategy.	Tranh (Painting) Lạnh (Cold) Mưa (Rain)	**Th**anh (Bar) L**h**oạnh M**t**ưa

3.3 BERT

BERT is a language representation model based on multi-layer bidirectional transformers encoder architecture. There is a wide variety of challenging natural language tasks that BERT can handle and achieve state-of-the-art performance, from classification, question answering, and sequence-to-sequence learning task, etc. BERT can represent sentence effectively by its encoding mechanism, including various embedding step as token embeddings, sentence embeddings, and transformers positional embeddings. Then, this BERT's last hidden presentation output from BERT is used as the input into the transformers architecture.

In this study, two well pre-trained BERT on the Vietnamese: the Google Multilingual BERT[2] (bert-base-multilingual-cased) and VinAI/phoBERT[3] are

[2] Github: https://github.com/google-research/bert.

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

considered. They are both trained with extensive Vietnamese corpus, while the multilingual BERT is the BERT base model, and phoBERT is a RoBERTa model [15] (which is a modified version from the base model).

3.4 Transformers

Before the transformers architecture, Encoder-Decoder architecture using RNN/ LSTM/ GRU (Gated Recurrent Unit) cell is used widely in machine translation and sequence-to-sequence tasks. This Encoder-Decoder architecture, also known as the Seq2Seq model, uses several RNN cells to encode the input tokens to hidden states and then sum all hidden states up before sending them to the decoder. Thanks to this hidden state, the decoders receive all previously encoded information and use it for the output token prediction task. Despite large capacities in handling sequence-to-sequence tasks, the Seq2Seq decoder may fail to fully capture the meaning and context of the last hidden presentation from the encoder. That means the more extended and more complex the input sequence, the less effective the hidden presentation can represent, which is known as the bottleneck problem.

While the attention mechanism takes several inputs simultaneously, construct weight matrices captured from each hidden presentation input to calculate a weighted sum of all the past encoder states. The decoder will then take the inputs and the provided attention weights, and through that, the decoder knows how to 'pay much attention' to which hidden presentation and vice versa. Another limitation of Seq2Seq architecture is that it handles the input sequentially, which means to compute for the current token at time *t*, we need the previous hidden state *t-1* and so on. Therefore, especially in the spelling correction task, if many erroneous tokens stick together, the correction of the last tokens can be poorly affected. The transformer architecture and attention mechanism come to cross those boundaries of those previous architectures. Based on the Encoder-Decoder architecture [26], the transformers architecture [28] uses stacked multi-head self-attention and fully connected layers. The transformer is designed to allow parallel computation and reduce the drop in performance due to long dependencies. It uses positional embeddings and multi-head self-attention to encode more information about the position of a token and the relation between each token.

As Shun has provided an advanced insights to incorporating pseudo data into the spelling correction task [12]. Consequently, a vast pseudo training dataset can be generated so that not only can the transformer maximize its parallelization ability but also BERT can represent its rich contextual embedding vectors.

3.5 Incorporate BERT into Transformers

As mentioned in Sect. 3.3, BERT is capable of deep language understanding by capturing contextual embedding of different words in a sequence. In addition, the Transformer model has been proved to be more efficient than popular Encoder-Decoder architectures, especially in the Machine Translation problem.

Some recent studies have treated the spelling correction problem as a machine translation job where the error sentence is the source sequence and the corrected sentence is the target sequence. And the combination of BERT and Transformer achieves state-of-the-art results for the English spelling correction task [9]. However, to the best of our knowledge, there has not been any research combining BERT and Transformer for Vietnamese spelling correction problem. The combination can be briefly summarized in the following steps:

- **Step 1**: Let the input sentence notated as X = $(x_1, ..., x_n)$, where n is the number of its tokens; x_i is the i-th token in X. BERT receives the input sequence tokens, and through its layers, BERT extracts them to hidden presentations notated as $H_B = (h_1, h_2, ..., h_n)$, where H_B is the output of the last layers in BERT.
- **Step 2**: The Encoder will take H_B from the previous step and encode the representation of each l layer H_E^l. The final contextual representation of the last encoder layer H_E^L is the output of the Encoder. The Encoder components consist of the multi-head self-attention mechanism, position-wise fully connected feed-forward network. A residual connection around each of the two sub-layers, followed by layer normalization. A Multi-head Attention is a component allowing the model to jointly attend to information from different representations and helps the encoder look at other words in the input sentence as it encodes a specific word for better-capturing contextual embedding.
- **Step 3**: The Decoder receives the representation H_E^L from the Encoder and decodes through its layers into final representation H_D^L. Similar to the Encoder, the Decoder possessed the same components of the Encoder. These Encoder H_E^L are to be used by each Decoder in its "encoder-decoder attention" layer which helps the Decoder focus on appropriate places in the input sequence.
- **Step 4**: Finally, the Decoder final representation H_D^L is mapped via a linear transformation and softmax to get the t-th predicted word ŷ. The decoding process continues until meeting the end-of-sentence token.

An illustration of our proposed method is shown in Fig. 1.

4 Experimental Evaluation

This section includes dataset description, evaluation method, model hyperparameter setting as well as experimental results of applying the Transformer architecture and BERT to Vietnamese spelling correction.

4.1 Experimental Dataset

This section describes the process of creating our training and testing set based on the Binhvq News Corpus[4] which contains 14,896,998 Vietnamese news

[4] Github: https://github.com/binhvq/news-corpus.

crawled from the Internet and preprocessed, including steps like HTML tag removal, duplicate removal, NFC standardization, and sentence segmentation. The corpus is gathered from reputable news and media sites in Vietnam, so the data is very reliable in terms of spelling. For the purpose of training and evaluating spelling correction, our newly constructed dataset must consist of two fields that can be described as a pair of correct and incorrect spelling sentences.

To the best of our knowledge, there is no specific survey as well as assessment on the rate of error types appearing in Vietnamese. However, Vietnamese often has common spelling mistakes: Region, FatFinger, Telex. Besides, some other types of errors are concerned, such as Edit-Distance, Abbreviation, Teencode, but rarely happened in practice. Therefore, error rate is reproduced based on our experience. Details of the rates of error types in the generated data set are listed in Table 3.

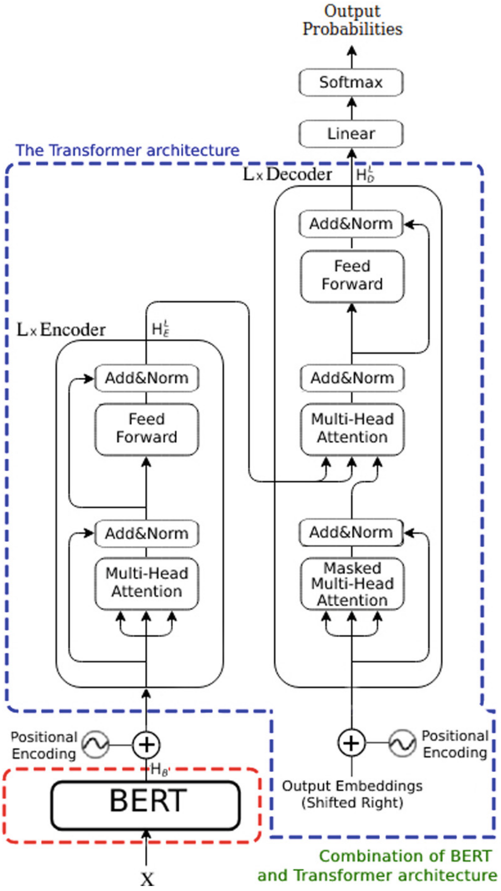

Fig. 1. Proposed combination between BERT and Transformer

Table 3. Training and testing sets

Error type	Error ratio (%)
Acronym	3.0
Teencode sets	3.0
Edit-Distance	3.0
FatFinger	30.0
Telex	30.0
Region	31.0

The training set is composed by randomly selecting 4,000,000 sentences from the Binhvq corpus and then apply the pseudo error generator to these correct ones. All the sentences must have an average word count between 50–60 words per sentence. Similarly, validating set and testing set are generated with the number of correct sentences chosen from the above corpus 20,000 and 6,000 respectively. Details of the dataset can be summarized in Table 4. The testing set is public and can be downloaded at the following link[5]

Table 4. Training and testing sets

Dataset	Size (#Pair of sentence)	Avg. Length per sentence (#token)
Training sets	4,000,000	60
Validating sets	20,000	60
Testing sets	6,000	60

4.2 Evaluating Metric

In the perspective of a spelling correction task, many traditional approaches used Accuracy, Precision, Recall, and F1 for evaluation [18,19]. These metrics require the predictions' words to have the same length as labels' words. Recently, BLEU score is chosen, especially in deep learning models because of its ability to adapt to different prediction lengths [9,31]. Therefore, BLUE [23] is selected for the evaluating task. BLEU, or the Bilingual Evaluation Understudy, is a score for comparing a candidate translation of text to one or more reference translations. Although developed for translation, it can be used to evaluate text generated for a suite of natural language processing tasks. Our BLEU configuration uses four n-grams settings because the spelling correction task critically requires the order of words in the sentence.

$$BP = \begin{cases} 1 & \text{if} \quad \mathrm{r} > \mathrm{c} \\ e^{(1-r/c)} & \text{if} \quad \mathrm{r} \leq \mathrm{c} \end{cases} \tag{1}$$

[5] Github: https://github.com/tranhamduong/Vietnamese-Spelling-Correction-testset.

$$BLEU = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2}$$

where BP stands for Brevity Penalty. c, r is the length of the predictions and labels, respectively. BP will penalty cases where the model failed to propose correction, or the change happens more than allowed (as the number of words need to be corrected must be equal to the actual corrected). p_n stands for modified n-gram precision, using n-grams up to length N and positive weights w_n summing to one. The n-gram precision can be simply understands as 'the number of corrected words which occur in reference sentence (ground-truth)' divided by 'the number of words after sentence transformed'. Therefore, the BLEU metrics has potential to not only to keep track strictly of word ordering by measuring n-gram (up-to-4) overlapping but also evaluate how a sentence has been corrected from the original despite the action (remove, edit, add more words).

4.3 Model Settings

Our models are implemented by fairseq toolkit [22] which is an re-implementation on the base Transformer architecture [28]. To find the appropriate hyperparameters for the proposed model, experiments with multiple model designations has been reviewed and the configuration of Jinhua work [31] are selected. Training details with hyperparameter settings are in the Table 5:

Table 5. Hyperparameters of transformer model

BERT model	Bert-base-multilingual-cased Vinai/phobert-base
Number of epochs	100
Dropout	0.3
Loss function	Labeled smoothed cross-entropy
Optimizer	Adam(0.9,0.98)
Learning rate	0.005
Label smoothing	0.1
Weight decay	0.0001
Beam search	5
Max tokens	1280

4.4 Experimental Results and Discussion

In this phase, we compared with the Google Docs spell checking tool[6] and other methods. From the results showed in Fig. 6, two versions of our model, Trans-

[6] The tool can be found on the Google Docs website (https://docs.google.com/). We collected samples by using a web browser behavior simulator based on Selenium framework that manipulate the Google spell checking tool to correct all of its possible suggestions.

former+vinai/ phoBERT and Transformer+BERT-multi-cased, achieved better results than the previous methods. This partly reinforces our hypothesis that using a pre-trained language model BERT brings two benefits to the spelling correction problem: being applicable in the spelling correction task and taking advantage of contextualized word embeddings. Firstly, as mentioned in the BERT paper, tasks such as Text Classification, Question and Answering, Sentence Tagging, etc., are recommended to be used in the fined-tuning phase but the spelling correction task. Due to our modification, at the first step, BERT is verified to be beneficial for the correction task. Secondly, when correcting an error word, the action of choosing a suitable candidate based on context words is the main characteristic of the spelling correction problem. Concretely, BERT produces contextualized word embedding (the same word for different contexts have different embeddings) helps the models to better utilize word embedding at correcting phase. Besides, the pre-training of BERT on a huge data set also makes fine-tuning for our model easier because of no need to re-train from the beginning, taking advantage of the knowledge from the language model.

Table 6. BLEU scores on models

Model	BLEU score
Google Docs spellchecking tool	0.6829
Transformer + vinai/phobert-base	0.8027
Word2Vec	0.8222
Transformer + bert-multi-cased	**0.8624**

Transformer + vinai/phobert-base: The proposed model based on the Transformer architecture and PhoBERT [17].

Word2Vec: The reimplementation of the Word2Vec approach in spelling correction [5].

Transformer + bert-multi-cased: The proposed model based on the Transformer architecture and BERT multilingual model [2].

For the objective of comparison and practical application, there are a few patterns that our excellent model gain out performance: telex and edit-distance error types, compared to the google docs spellchecking tool. This happened partially because we designated more of these types of error to achieve our goal. More tuning is needed in future work on the error type distribution to improve performance for other types of errors.

The google docs spellchecking tool has another advantage over our model is the ability to restrict unnecessary correction. Additionally, the emergence of proper nouns also makes our model ineffective. When it comes to a proper noun, especially Vietnamese proper names, our model tends to correct them, which should not be the case. To overcome this weakness, some supporting components can be developed to the proposed architecture: Applying a name entity recognition component or an independent spellchecker to determine to correct a word or not.

5 Conclusion

In this paper, a combination of BERT and Transformer architecture is implemented for the Vietnamese spell correction task. The experimental results show that our model outperforms other approaches with a 0.86 BLUE score and can be used in real-world applications. Besides, a dataset is constructed for related works based on a breakdown of the spelling correction problem to define which errors commonly happened and need more attention.

To our concern, despite the improvement in the model's performance, there are late inferences due to large and complex architecture. In addition, due to the different distribution of data in the pre-trained model compared to data for the spelling correction task, we can not fully utilize the representation of pre-trained words, resulting in the model sometimes try to correct the unwanted words.

In the future, our architecture will be experimented with other existing pre-trained language models to see how well the compatibility they are. Moreover, we also evaluate our model's accuracy on a bigger dataset. Finally, investigating and analyzing errors that may happen in practices is our priority in order to create a better error pseudo generator.

References

1. Bassil, Y., Alwani, M.: Ocr post-processing error correction algorithm using google online spelling suggestion. J. Emerging Trends Comput. Inf. Sci. (2012)
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding, pp. 4171–4186 (2019)
3. of Education Vietnam M: Sách Giáo khoa tiếng Việt 1 (Tập Một) Ministry of Education Publisher (2002)
4. Fivez, P., Šuster, S., Daelemans, W.: Unsupervised context-sensitive spelling correction of clinical free-text with word and character n-gram embeddings. In: BioNLP 2017, pp. 143–148. Association for Computational Linguistics, Vancouver, Canada Aug 2017
5. Fivez, P., Suster, S., Daelemans, W.: Unsupervised context-sensitive spelling correction of english and dutch clinical free-text with word and character n-gram embeddings (2017)
6. Hao, C.X.: Tiếng Việt, văn Việt, người Việt Youth Publisher (2003)
7. Hladek, D., Staš, J., Pleva, M.: Survey of automatic spelling correction. Electronics **9**, 1670 (2020)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–80 (1997)
9. Kaneko, M., Mita, M., Kiyono, S., Suzuki, J., Inui, K.: Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4248–4254. Association for Computational Linguistics (2020)

10. Khanh, P.H.: Good spelling of vietnamese texts, one aspect of computational linguistics in vietnam. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL 2000 p. 1–2. Association for Computational Linguistics, USA (2000)
11. Kissos, I., Dershowitz, N.: Ocr error correction using character correction and feature-based word classification. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 198–203. IEEE (2016)
12. Kiyono, S., Suzuki, J., Mita, M., Mizumoto, T., Inui, K.: An empirical study of incorporating pseudo data into grammatical error correction. 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. 1236–1242. Association for Computational Linguistics, Hong Kong, China, Nov 2019
13. Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.: OpenNMT: Open-source toolkit for neural machine translation. In: Proceedings of ACL 2017, System Demonstrations, pp. 67–72. Association for Computational Linguistics, Vancouver, Canada (Jul 2017)
14. Liu, J., Cheng, F., Wang, Y., Shindo, H., Matsumoto, Y.: Automatic error correction on Japanese functional expressions using character-based neural machine translation. In: Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation. Association for Computational Linguistics, Hong Kong, 1–3 Dec 2018
15. Liu, Y., et al.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, 2–4 May 2013. Workshop Track Proceedings (2013)
17. Nguyen, D.Q., Nguyen, A.T.: PhoBERT: Pre-trained language models for Vietnamese. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1037–1042 (2020)
18. Nguyen, H.T., Dang, T.B., Nguyen, L.M.: Deep learning approach for vietnamese consonant misspell correction. In: Nguyen, L.-M., Phan, X.-H., Hasida, K., Tojo, S. (eds.) PACLING 2019. CCIS, vol. 1215, pp. 497–504. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-6168-9_40
19. Nguyen, H., Dang, T., Nguyen, T.T., Le, C.: Using large n-gram for vietnamese spell checking. Adv. Intell. Syst. Comput. **326**, 617–627 (2015)
20. Nguyen, P.H., Ngo, T.D., Phan, D.A., Dinh, T.P., Huynh, T.Q.: Vietnamese spelling detection and correction using bi-gram, minimum edit distance, soundex algorithms with some additional heuristics. In: 2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies, pp. 96–102. IEEE (2008)
21. Nguyen, Q.D., Le, D.A., Zelinka, I.: Ocr error correction for unconstrained vietnamese handwritten text, pp. 132–138 (12 2019)
22. Ott, M., et al.: fairseq: A fast, extensible toolkit for sequence modeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). pp. 48–53. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
23. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting

of the Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002)

24. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (Oct 2014)
25. Pham, N.L., Nguyen, T.H., Nguyen, V.V.: Grammatical error correction for vietnamese using machine translation. In: Nguyen, L.-M., Phan, X.-H., Hasida, K., Tojo, S. (eds.) PACLING 2019. CCIS, vol. 1215, pp. 505–512. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-6168-9_41
26. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS 2014, pp. 3104–3112. MIT Press, Cambridge, MA, USA (2014)
27. Tedjopranoto, M., Wijaya, A., Santoso, L., Suhartono, D.: Correcting typographical error and understanding user intention in chatbot by combining n-gram and machine learning using schema matching technique. Int. J. Mach. Learn. Comput. **9**, 471–476 (2019)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
29. Xuan, P.: Solutions to spelling mistakes in written vietnamese. VNU J. Sci. Educ. Research **33**(2) (2017)
30. Yuan, Z., Briscoe, T.: Grammatical error correction using neural machine translation. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 380–386. Association for Computational Linguistics (Jun 2016)
31. Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., Li, H., Liu, T.: Incorporating BERT into neural machine translation. In: Eighth International Conference on Learning Representations (2020)