Spelling Error Correction using XLM-RoBERTa

Hao Huynh Nhat

University of Information Technology, Ho Chi Minh City, Vietnam

18520714@gm.uit.edu.vn

***Abstract*—Spelling error correction is an important task that has many real-life applications, such as helping language learners to detect his/her own spelling mistakes, or automatic checking and correcting errors in many formal articles. There have many approaches for this task, especially in English, but not in other languages like Vietnamese. In this work, we explore 2 new methods for Vietnamese spelling error correction called Hard-Masked XLM-R and Soft-Masked XLM-R. Both methods are base on XLM-RoBERTa, which is a recent state of the art multilingual language representation model, and achieve impressive performance on the task with the Hard-Masked XLM-R.**

***Index Terms*—Spelling error correction, XLM-R, Soft-Masked XLM-R, Hard-Masked XLM-R.**

I. Introduction

The goal of the spelling error correction system is to detect and correct potential error words in a given sentence. The task can be decomposed into 2 sub-tasks: (1) detect the wrong words in the sentence (if exist) and (2) replace the wrong words with the appropriate words. The former is an easy one, but the latter need human-level language understanding ability to perform well. There are many approaches for this task, at either word-level or character-level, including using some heuristics to detect the wrong words and Edit Distance, SoundEx algorithm to correct it [4] (Vietnamese). Machine translation approach has been explored for the task [6] (English), a character-level encoder-decoder neural network combine with attention mechanism [5] (English), a Soft-Masked BERT model for Chinese spelling error correction [7] (Chinese), which uses a system of 2 networks to address the problem: A detector network and a corrector network. In this work, inspired by the Soft-Masked BERT, we will explore 2 new models similar to the Soft-Masked BERT for Vietnamese spelling error correction: Soft-Masked XLM-R and Hard-Masked XLM-R, both bases on the XLM-RoBERTa language representation model [1]. Unfortunately, we do not have enough computation power to train the Soft-Masked XLM-R model to see if it work, but we can train the Hard-Masked XLM-R and show that it gives a very impressive result on detecting and correcting the miss-spelled words.

The contributions of this work include (1) explain the Soft-Masked XLM-R and the Hard-Masked XLM-R architectures, then discuss the advantages and disadvantages of each architecture (2) the results of the Hard-Masked XLM-R on a synthesized dataset.

Identify applicable funding agency here. If none, delete this.

II. Model Architecture

Given an input sentence $X = x_1, x_2, ..., x_{T_x}$, we wish to map to an output sentence $Y = y_1, y_2, ..., y_{T_y}$ with $T_x = T_y$ is the length of the input sentence. We looking for a system $f(x)$ that map from X to Y, where the miss-spelled words, abbreviated words, teencode, and other errors in X will be corrected in Y. The system we discussed here has 2 components: A detector network and a corrector network. The detector network can be either a Bi-GRU or a Bi-LSTM network, and the corrector is base on a language representation model, such as BERT [2], XLM-R,... depend on the language it supports. The following subsections describe in detail the model architecture of Soft-Masked XLM-R and Hard-Masked XLM-R.

A. Soft-Masked XLM-R

The Soft-Masked XLM-R is inspired by the Soft-Masked BERT. It works exactly like Soft-Masked BERT, but replace the BERT model with the XLM-R model. Soft-Masked BERT uses a Bi-GRU as the detector, but we use a Bi-LSTM as the detector because there is not much difference in the performance of Bi-GRU or Bi-LSTM as the detector. The system uses its language representation model's encoder combine with a fully connected network and a soft-max function as its corrector network. Fig. 1 describes the detail of how it works.

The input text first go through an embedding layer and become embedding vectors $E = (e_1, e_2, ..., e_n)$, then the detector take in these vectors and give probabilities $P = (p_1, p_2, ..., p_n)$ of each token that the detector thinks it belongs to a wrong word. Then the soft-masked embedding vectors $e_i^{'}$ for the i^{th} token is created using the previous embedding vectors, the probabilities given by the detector as followed:

$$e_i^{'} = p_i * e_{mask} + (1 - p_i) * e_i \tag{1}$$

Where e_i is the input embedding to the detector, e_{mask} is the pre-trained embedding vector of mask token of the language representation model. If the probability p_i is high, then $e_i^{'}$ will close to e_{mask}, otherwise it will close to e_i.

The correction network is a sequential multi-class labeling model based on XLM-R. It is responsible for replacing wrong tokens with appropriate tokens base on the context of the

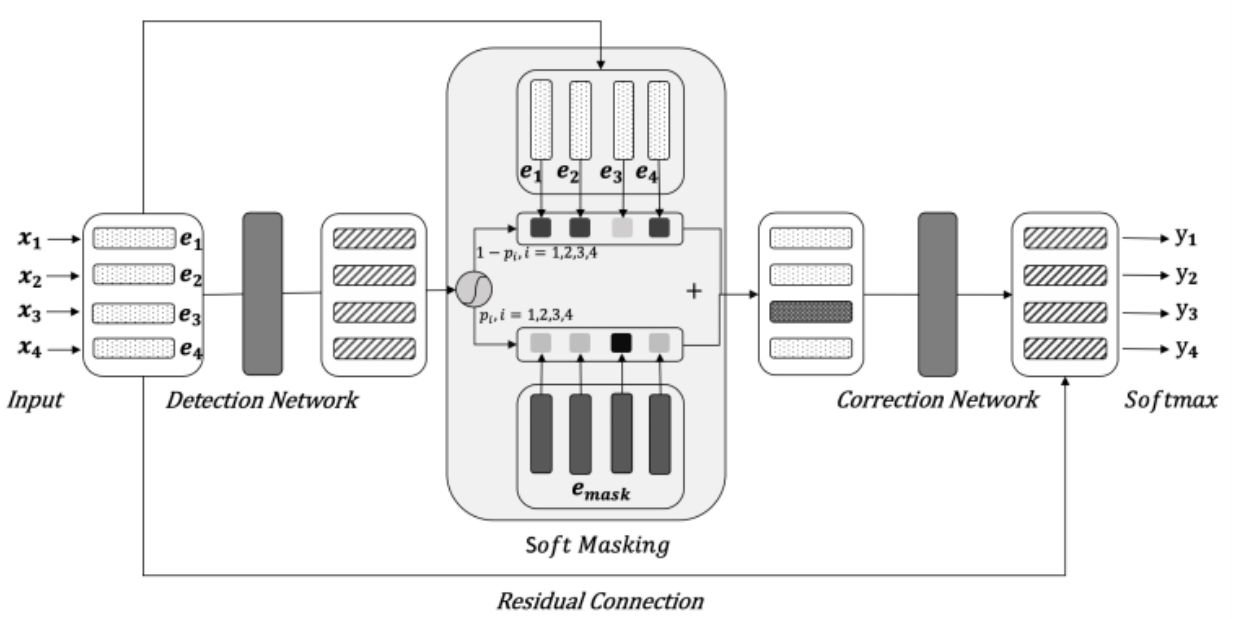

Fig. 1. Architecture of Soft-Masked BERT [7]

sentence. It uses the XLM-R encoder to get a new vector representation for each token, then uses a fully connected network combine with a soft-max layer to predict the probability of each token in the vocabulary, then choose the token which has the highest probability to be the candidate token.

Unfortunately, the number of parameters that need to be trained and fine-tune are more than 480M parameters which are too large and we do not have enough computation power to train it to see if it works. So we have created a new version of Soft-Masked XLM-R called Hard-Masked XLM-R which has 293M parameters, but we only need to train the detector and the total parameters that need to be trained are only more than 15M parameters.

B. Hard-Masked XLM-R

The Hard-Masked XLM-R has a similar architecture as Soft-Masked XLM-R, include a detector and a corrector. But its detector and corrector are independent. Each component uses its own embedding layer and have the separate vocabulary for the embedding layers.

The detector is a Bi-LSTM network with 2 layers, hidden size of 512. It uses an embedding layer converts from a vocabulary of 10000 tokens using the sentencepiece tokenizer [3] to an embedding size of 512. The sentencepiece tokenizer was trained on all the crawled data, which is more than 3M Vietnamese sentences.

The corrector is a pre-trained masked language model base on XLM-R called XLMRobertaForMaskedLM, which is publicly released by huggingface. This model will predict a new token to replace the mask token in the sentence. It uses its own embedding layer, which has a size of 768 and uses a vocabulary of 250002 tokens. We do not fine-tune this model because it already works very good at choosing the right word candidates for the mask token.

The Hard-Masked XLM-R will works as followed:

1) The detector takes in a sentence, tokenize it using the sentencepiece tokenizer, convert it into embedding vectors with its own embedding layers, feed-forward these vectors through the Bi-LSTM network to get the probabilities of belonging to a wrong word for each token. Then use the 0.5 thresholds to decide if a token belongs to a wrong word or not. Then detokenize these tokens and replace all the words that have a probability greater than 0.5 with the mask token and give the new sentence to the corrector.
2) The corrector receives the new sentence from the detector, again tokenize and convert it to embedding vectors using its embedding layers and vocabulary. Then feed-forward these vectors through the XLM-R encoder to get the new vector representations. Then a fully connected neural network combine with a soft-max function will use these vector representations to predict the candidates for the masked tokens.

C. Compare the Soft-Masked XLM-R and Hard-Masked XLM-R

The Soft-Masked XLM-R combines its detector and corrector into one seamless system, which only tokenizes sentence into tokens and converts it into embedding one time. While the Hard-Masked XLM-R has two independent networks and each uses a separated embedding layer and vocabulary. In the detector network, using the separated only-Vietnamese vocabulary size of 10000 tokens will help it to reduce its number of parameters and avoid irrelevant tokens compare

to the XLM-R vocabulary which has 2500002 tokens of 100 languages. But a separated system like that comes with a price: the corrector is heavily dependent on the detector. If the detector can not recognize an error, then the corrector will ignore it as well. We also have to make a hard decision on which token is wrong and which is not, base on a hard threshold. On the other hand, the Soft-Masked avoids making this hard decision by using a soft way to mask its embedding vectors.

III. Experiment result

A. Creating data

At the time we experimenting on these networks, there is no publicly released dataset or benchmark dataset for the Vietnamese language on the task. So we only evaluate our model on the synthesized dataset. To create data for experiments, we crawl more than 3 million sentences from many online news websites such as vietnamnet.vn, thanhnien.vn, vnexpress.net,... All these sentences are in Vietnamese and have the journalistic writing style, which we expect that the model trained on this dataset will perform a little worse if we test it on other writing styles compare to the journalistic writing style.

There are two types of errors that can be created by the synthesis function: non-word errors and real-word errors. Non-word errors include typos, abbreviation,... real-word errors include using the not appropriate word in the sentence context, or mistake by homophone words,... because the synthesis function's nature is random, the same sentence goes through the function two times will create two different samples with different errors at a different positions. Using this random nature, we can synthesize more than 3M samples with 3M original sentences. For the training set, we only use 2M sentences with different types of errors, and the development set has 20000 sentences randomly chosen from the original set of sentences to go through the synthesis function.

B. Result on development set

The result we show here only for the Hard-Masked XLM-R model. We trained its detector on 2 epochs with 2M samples, batch size 128, learning rate 0.001. At this point, the detector f1 score on the dev set is 0.96. Then we reduce the learning rate to 0.0005 and continue to train the model until there is no improvement and achieve 96.7 f1 scores. Note that this result is on the synthesized dataset, which is created also by the synthesize function. The lack of labeled real-world datasets for the task in the research community has led us to only using synthesize data for evaluation.

We do not fine-tune the corrector network, because of its performance already very good in the Vietnamese language. We do not have any evaluation method on the performance of the corrector, because it always suggests either the original label word or an appropriate word to the masked token. The next step to choose the candidate to fill in the masked token is a function that will compute the edit distance between the miss-spelled word and the candidates, and choose the closet edit distance candidate to be the replaced word.

IV. Conclusion

In this work, we have explored two new model architecture, inspired by the Soft-Masked BERT model, for the Vietnamese spelling error correction base on the XLM-R language representation model. The Soft-Masked XLM-R and Hard-Masked XLM-R both have a detector-corrector architecture, where the detector is a sequential binary labeling model which is responsible for detecting wrong words, and the corrector is a sequential multi-class labeling model which is responsible for replacing the wrong words with appropriated words. As future work, we plan to extend the ability of the detector to recognize more error types in texts by train it on more diverse text data.

References

[1] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[3] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018.

[4] P. H. Nguyen, T. D. Ngo, D. A. Phan, T. P. T. Dinh, and T. Q. Huynh. 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*, pages 96–102, 2008.

[5] Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. Neural language correction with character-based attention. *CoRR*, abs/1603.09727, 2016.

[6] Zheng Yuan and Ted Briscoe. 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*, pages 380–386, San Diego, California, June 2016. Association for Computational Linguistics.

[7] Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. Spelling error correction with soft-masked bert, 2020.