



words to diacritics words. This method is not effective because many words in the dictionary do not appear frequently in real texts. The accuracy of this tool is about from 60% to 85% depending on the domain of texts. VietEditor<sup>2</sup> toolkit alleviate the weakness of VietPad by building the phrase dictionary and uses it after mapping words to find the most appropriate outputs.

The second approach is using machine learning methods to handle this problem. They apply some common machine learning models such as conditional random field (CRF), support vector machine (SVM), and N-gram language models to restore diacritics for Vietnamese texts. [2] proposes viAccent toolkit that is a combination of N-grams, structured perception and CRF. N-grams is used as features for CRF to label diacritics for input sentences. They achieve an accuracy of 94.3% on a newspaper dataset. [3] combines both language model and co-occurrence graph to capture information from non-diacritic texts. For inference, they apply dynamic programming to find the best output sequences based on information from input texts. [4] proposes a pointwise approach for automatically recovering diacritics, using three features for classification including N-grams of syllables, syllable types, and dictionary word features. They achieve an accuracy of 94.7% by using SVM classifier. [5] gives an empirical study for Vietnamese diacritic restoration by investigating five strategies: learning from letters, learning from semi-syllables, learning from syllables, learning from words, and learning from bi-grams. They combine AdaBoost and C4.5 algorithms to get better results. Their best accuracy is 94.7% when using letter-based feature set. [6] formulates this task as sequence tagging problem and use CRF and SVM models to restore diacritics. They achieve the accuracy of 93.8% on written texts by using CRF at syllable level.

Machine translation-based approach has emerged as the best way to handle this problem. [1] and [7] formulate this task as machine translation problem and apply phrase-based translation method by using Moses toolkit. Both of them report the accuracy of 99% on their dataset but the size of this dataset is relatively small. For this reason, in this paper, we experiment this method on a large dataset to get fair comparisons.

### III. METHODOLOGY

We treat the diacritic restoration problem as a machine translation problem and apply phrase-based and neural-based machine translation methods for this task. In particular, non-diacritic and diacritic texts are considered as source and target languages respectively, and machine translation models are trained to learn how to restore diacritics.

#### A. Phrase-Based Machine Translation

Phrase-based machine translation is one type of statistical machine translation that translate phrases in source language to phrases in target language [8], [9]. The main idea of this approach is an input sentence is segmented into a number of sequences of consecutive words (phrases). After that, each phrase in source language is translated to one phrase in target language that might be reordered.

The phrase translation model is based on the noisy channel model. In particular, it tries to maximize the translation probability from source sentence  $\mathbf{f}$  to target sentence  $\mathbf{e}$ . Applying Bayes rule, we can reformulate this probability as

$$\arg \max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \arg \max_{\mathbf{e}} p(\mathbf{f}|\mathbf{e})p(\mathbf{e})\omega^{\text{length}(\mathbf{e})} \quad (1)$$

where  $\omega$  is added to calibrate the output length.

During decoding, the source sentence  $\mathbf{f}$  is segmented into a sequence of  $N$  phrases  $\mathbf{f}_i$ . After that, each source phrase  $\mathbf{f}_i$  is translated to target phrase  $\mathbf{e}_i$  by the probability distribution  $\phi(\mathbf{f}_i|\mathbf{e}_i)$ . The sequence of target phrases might be reordered by a relative distortion probability distribution  $d(\text{start}_i, \text{end}_{i-1})$  where  $\text{start}_i$  denotes the start position of the source phrase that was translated into the  $i^{\text{th}}$  target phrase, and  $\text{end}_{i-1}$  denotes the end position of the source phrase that was translated into the  $(i-1)^{\text{th}}$  target phrase. As sum,  $p(\mathbf{f}|\mathbf{e})$  can be calculated as

$$\prod_{i=1}^N \phi(\mathbf{f}_i|\mathbf{e}_i)d(\text{start}_i, \text{end}_{i-1}) \quad (2)$$

Figure 1 presents an example of phrase-based machine translation system.

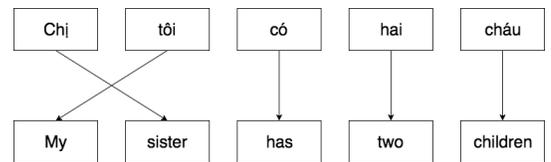


Figure 1: Phrase-based machine translation system that translates source sentence “Chị tôi có hai cháu” to target sentence “My sister has two children”

#### B. Neural-Based Machine Translation

In the last few years, deep neural network approaches have achieved state-of-the-art results in many natural language processing (NLP) task. There are a lot of research that applied deep learning methods to improve performances of their NLP systems. For machine translation problem, [10], [11] proposed a sequence-to-sequence model that achieved best results for many bi-lingual translation tasks. The general architecture of this model is the combination of two recurrent neural networks. One network encodes a sequence of words in source language into a fixed-length vector representation, and the other decodes this vector into another sequence of words in target language. Both of these two networks are jointly trained to maximize the conditional probability of a target sentence given a source sentence. In particular, the conditional probability  $p(\mathbf{e}|\mathbf{f})$  is computed as

$$\log(\mathbf{e}|\mathbf{f}) = \sum_{j=1}^m \log p(\mathbf{e}_j|\mathbf{e}_{<j}, \mathbf{s}) \quad (3)$$

where  $\mathbf{s}$  is representation vector produced by encoder module.

In decoding stage, the conditional probability of a word given previous words is computed as

$$p(\mathbf{e}_j|\mathbf{e}_{<j}, \mathbf{s}) = \text{softmax}(g(\mathbf{h}_j)) \quad (4)$$

<sup>2</sup>[http://irc.quangbinhuni.edu.vn:8181/dspace/bitstream/TVDHQB\\_123456789/264/3/Themdautiengviet.pdf](http://irc.quangbinhuni.edu.vn:8181/dspace/bitstream/TVDHQB_123456789/264/3/Themdautiengviet.pdf)  
(Vietnamese)

where  $\mathbf{h}_j$  is the hidden state at time step  $j$  of recurrent neural network that computed by previous hidden state and representation vector  $\mathbf{s}$ , and  $g$  is a function that transforms the hidden state to vocabulary-sized vector. Figure 2 present the architecture of sequence-to-sequence model.

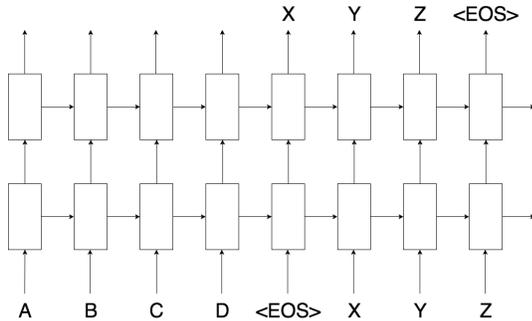


Figure 2: The architecture of sequence-to-sequence model

#### IV. EXPERIMENT

##### A. Dataset

To evaluate these machine translation approaches for this problem on the large dataset, we first collect 10,000 news articles from the web and then remove non-standard characters and diacritics from the original text to build a parallel corpus of about 180,000 sentence pairs. We use 80% of this dataset as a training set, 10% of this dataset as a development set, and the remaining as a testing set. Table II shows the statistics of this dataset.

Table II: Numbers of sentences and words in each part of the dataset

	#sentence	#word
Training	140474	3391193
Development	17559	424309
Testing	17559	423729

##### B. Evaluation Method

We utilize Moses<sup>3</sup> and OpenNMT<sup>4</sup> toolkits as representatives for phrase-based and neural-based machine translation approaches respectively. To evaluate performances of these toolkits, we use an accuracy score that calculates the percentage of correct words restored by these systems and a BLEU score that evaluates results of translation systems.

##### C. Results and Discussions

We train these two toolkits on the training set and use the development set for tuning. In particular, we use the development set to adjust parameters of Moses toolkit and to early stop the training of OpenNMT toolkit. Finally, we restore diacritics of texts in the testing set. We use the standard setting in these two toolkits when training and inference. For Moses toolkit, we use KenLM<sup>5</sup> to build 3-gram language model and GIZA++<sup>6</sup> for word alignment. For OpenNMT toolkit, we use

the sequence-to-sequence model described in [11]. This model consists of encoder and decoder modules that are recurrent neural network models. Table III shows the accuracy and BLEU scores of these two systems.

Table III: Accuracy and BLEU scores of each system on our testing set

	Accuracy	Bleu
Phrase-based (Moses)	97.32	94.11
Neural-based (OpenNMT)	96.15	91.59

The main purpose of BLEU score is evaluating the quality of the machine translation system [12]. It is not suitable to use this score to assess the performances of these systems for diacritic restoration task. We, therefore, focus only the accuracy score. Both of these systems achieve the state-of-the-art results for Vietnamese diacritic restoration task. In particular, the Moses toolkit achieves an accuracy of 97.32%, which is slightly higher than an accuracy of 96.15% of OpenNMT toolkit. The reason for this result may be the size of the training set. Neural-based approach often requires a large amount of training data to get a good performance while our training set has only 140,000 sentence pairs. Moreover, previous works show that using pre-trained word embeddings help to improve greatly the performance of the neural machine translation, but in this task, we do not use any pre-trained word embeddings.

While the performance of OpenNMT toolkit is not better than Moses toolkit in this dataset, we realize that OpenNMT toolkit requires less time for training and has a higher speed when restore diacritics for input sentences. Table IV shows the training time and the average speed at the inference stage of these two systems.

Table IV: Training times (hours) and Testing speeds (#sentence/second) of Moses and OpenNMT toolkits

	Training	Testing
Phrase-based (Moses)	12 hours	10 sent/s
Neural-based (OpenNMT)	8 hours	22 sent/s

In particular, we train and evaluate two system at the same setting. The details of hardware configuration are Intel Xeon E5-2686, 60GB of RAM, and Tesla K80 12GB. OpenNMT toolkit takes 8 hours for training while Moses toolkit needs 12 hours. For inference stage, OpenNMT toolkit is likely to handle 22 input sentences per second which is twice as fast as Moses toolkit. The reason is that OpenNMT toolkit can take advantage of the performance of GPU that has many CUDA cores to parallel processing.

#### V. CONCLUSION

In this paper, we present the empirical study of machine translation-based approaches for Vietnamese diacritic restoration. In particular, we conduct experiments to compare two common approaches for machine translation including phrase-based and neural-based methods. Both of two systems achieve state-of-the-art results for Vietnamese diacritic restoration task. While the phrase-based method has a slightly higher accuracy,

<sup>3</sup><http://www.statmt.org/moses/>

<sup>4</sup><http://opennmt.net/>

<sup>5</sup><http://kheafield.com/code/kenlm/>

<sup>6</sup><http://www.statmt.org/moses/giza/GIZA++.html>

the neural-based method requires less time for training and has much faster inference speed.

In the future, on the one hand, we plan to improve neural-based approach by enlarging our corpus so as to provide more data for training. On the other hand, we would like to incorporate pre-trained Vietnamese word embeddings to boost the accuracy of this approach.

## REFERENCES

- [1] T. N. D. Do, D. B. Nguyen, D. K. Mac, and D. D. Tran, "Machine translation approach for vietnamese diacritic restoration," in *Proceedings of the 2013 International Conference on Asian Language Processing*, Urumqi, China, 2013, pp. 103–106.
- [2] T. T. Truyen, D. Q. Phung, and S. Venkatesh, "Constrained sequence classification for lexical disambiguation," in *PRICAI 2008: Trends in Artificial Intelligence: 10th Pacific Rim International Conference on Artificial Intelligence*. Hanoi, Vietnam: Springer, 2008, pp. 430–441.
- [3] N. H. Trong and P. Do, "A new approach to accent restoration of vietnamese texts using dynamic programming combined with co-occurrence graph," in *Proceedings of the 2009 International Conference on Computing and Communication Technologies*. Da Nang, Vietnam: IEEE, 2009, pp. 1–4.
- [4] T. A. Luu and K. Yamamoto, "A pointwise approach for vietnamese diacritics restoration," in *Proceedings of the 2012 International Conference on Asian Language Processing*, Hanoi, Vietnam, 2012, pp. 189–192.
- [5] K.-H. Nguyen and C.-Y. Ock, *Diacritics Restoration in Vietnamese: Letter Based vs. Syllable Based Model*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 631–636.
- [6] M. T. Nguyen, Q. N. Nguyen, and H. P. Nguyen, "Vietnamese diacritics restoration as sequential tagging," in *Proceedings of the 2012 International Conference on Computing Communication Technologies, Research, Innovation, and Vision for the Future*, Ho Chi Minh, Vietnam, 2012, pp. 1–6.
- [7] L.-N. Pham, V.-H. Trab, and V.-V. Nguyen, "Vietnamese text accent restoration with statistical machine translation," in *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation*, Taipei, Taiwan, 2013, pp. 423–429.
- [8] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 48–54.
- [9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 177–180.
- [10] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014.
- [11] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods on Natural Language Processing*, Lisbon, Portugal, 2015.
- [12] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Pennsylvania, United States: Association for Computational Linguistics, 2002, pp. 311–318.