Vietnamese Sentence Paraphrase Identification using Pre-trained Model and Linguistic Knowledge

Dien Dinh, Nguyen Le Thanh
Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam

Abstract—The paraphrase identification task identifies whether two text segments share the same meaning, thereby playing a crucial role in various applications, such as computer-assisted translation, question answering, machine translation, etc. Although the literature on paraphrase identification in English and other popular languages is vast and growing, the research on this topic in Vietnamese remains relatively untapped. In this paper, we propose a novel method to classify Vietnamese sentence paraphrases, which deploys both the pre-trained model to exploit the semantic context and linguistic knowledge to provide further information in the identification process. Two branches of neural networks built in the Siamese architecture are also responsible for learning the differences among the sentence representations. To evaluate the proposed method, we present experiments on two existing Vietnamese sentence paraphrase corpora. The results show that for the same corpora, our method using the PhoBERT as a feature vector yields 94.97% F1-score on the VnPara corpus and 93.49% F1-score on the VNPC corpus. They are better than the results of the Siamese LSTM method and the pre-trained models.

Keywords—Paraphrase identification; Vietnamese; pre-trained model; linguistics; neural networks

I. INTRODUCTION

Paraphrase identification, a task that whether two text segments with different wordings express similar meaning, is critical in various Natural Language Processing (NLP) applications, such as text summarization, text clustering, computer-assisted translation, and, especially plagiarism detection [1]. Paraphrases can take place at different linguistic levels, ranging from word and phrase to sentence and discourse. For instance, Neculoiu et al. [2] deployed Siamese recurrent networks to determine similarity among texts, normalizing job titles that are paraphrases at the word level. Meanwhile, to detect paraphrases at the discourse level, Liu et al. [3] calculated semantic equivalence among academic articles published in 2017 to identify documents with similar themes and contents.

Paraphrase corpora are corpora that contain pairs of sentences that convey the same meaning. Regarding Vietnamese, there have been two paraphrase corpora published for the language, one of which is vnPara by Bach et al. [4], while the other is VNPC (Vietnamese News Paraphrase Corpus) by Nguyen-Son et al. [5]. Both of these corpora consist of sentence-level paraphrases. Examples of paraphrases and non-paraphrases extracted from vnPara and VNPC are shown in Tables I and II, respectively.

While string matching is the simplest solution to the paraphrase identification question in theory, it does not yield high accuracy rates in practice. Two segments of text that are constructed with different strings can still be paraphrases. On the contrary, various text segments that have overlapping substrings can denote different interpretations, and thus they are non-paraphrases.

According to Suzuki et al. [1], these two types of paraphrases and non-paraphrases are categorized as a non-trivial class, whose instances hold a key role in the paraphrase identification task. Table III presents examples of non-trivial instances extracted from VNPC. The WOR column in this table represents the word overlap rate of two given sentences, which is calculated using Jaccard index [7], where X and Y denote the set of words of those two sentences:

<table>
<thead>
<tr>
<th>Paraphrase</th>
<th>Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASA sẽ tìm thấy người hành tinh trong 20 năm tới.</td>
<td>vnPara</td>
</tr>
<tr>
<td>ASA sẽ tìm thấy người hành tinh trong 20 năm tới.</td>
<td>vnPara</td>
</tr>
<tr>
<td>ASA will find aliens in the next 20 years</td>
<td>VNPC</td>
</tr>
<tr>
<td>ASA says that it’s possible to find aliens in the next 20 years</td>
<td>VNPC</td>
</tr>
<tr>
<td>Đừng chỉ yêu, mà đọc nấy chà hơi động mà ở chế độ “ngủ đông”, chờ lệnh tấn công.</td>
<td>VNPC</td>
</tr>
<tr>
<td>Đừng chỉ yêu, mà đọc nấy chà hơi động mà ở chế độ “ngủ đông”, chờ lệnh tấn công.</td>
<td>VNPC</td>
</tr>
</tbody>
</table>

TABLE I. EXAMPLES OF VIETNAMESE PARAPHRASES AND THEIR TRANSLATION INTO ENGLISH

<table>
<thead>
<tr>
<th>Non-paraphrase</th>
<th>Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Các giải dự Trung Quốc đã tạo công hệ thống mạng của một tổ chức nghiên cứu lớn của chính phủ Canada, giới chức Canada ngày 29/7 cho biết:</td>
<td>vnPara</td>
</tr>
<tr>
<td>Các giải dự Trung Quốc đã tạo công hệ thống mạng của một tổ chức nghiên cứu lớn của chính phủ Canada, giới chức Canada ngày 29/7 cho biết:</td>
<td>vnPara</td>
</tr>
<tr>
<td>Chinese spies have attacked the network system of a country in Southeast Asia.</td>
<td>VNPC</td>
</tr>
<tr>
<td>Chinese spies have attacked the network system of a country in Southeast Asia.</td>
<td>VNPC</td>
</tr>
<tr>
<td>Cậu thứ nhất gia thế ba Real từng đào tạo là Alvaro Negredo.</td>
<td>VNPC</td>
</tr>
<tr>
<td>Cậu thứ nhất gia thế ba Real từng đào tạo là Alvaro Negredo.</td>
<td>VNPC</td>
</tr>
<tr>
<td>The third most valuable young player who Real has trained is Alvaro Negredo.</td>
<td>VNPC</td>
</tr>
<tr>
<td>The third most valuable young player who Real has trained is Alvaro Negredo.</td>
<td>VNPC</td>
</tr>
</tbody>
</table>

TABLE II. EXAMPLES OF VIETNAMESE NON-PARAPHRASES AND THEIR TRANSLATION INTO ENGLISH
are relevant to the current study in Section 2, and then propose a novel method to identify sentence paraphrases in Vietnamese in Section 3. Section 4 contains our experiments on evaluating the performance of this method. Section 5 concludes the work and discusses future directions.

II. RELATED WORK

Various paraphrase identification and similarity measurement methods have been proposed for a range of languages. The methods can be categorized into four different groups of approaches: string-based, corpus-based, knowledge-based, and hybrid [15]. In this section, we first present the methods laid out in these four approaches and then discuss previous work conducted for the Vietnamese language.

A. String-based Approach

The advantage of this approach lies in its simplicity, as most of the methods are easy to implement. The main information is derived from the text itself, with little to no reliance on additional resources. However, this also lowers the accuracy of the approach, as all of these methods do not detect semantic similarity effectively, thereby failing to account for non-trivial cases, as discussed earlier.

First, among the similarity measures that are widely used across different applications is the Damerau-Levenshtein distance [16]. This measure considers the minimum number of operations needed to convert one text into the other. An operation can be either an insertion, a deletion, a substitution of a single character or a transposition of two consecutive characters.

Secondly, the n-gram comparison of two texts is also considered a common algorithm. An n-gram is a sequence of n elements of a text sample. These n elements can be characters, phonemes, syllables, or words, depending on the tasks and applications. Alberto et al. define the formula to calculate the text similarity value using n-grams as follows [17]:

\[
\text{Similarity} = \frac{\text{Number of the same n-grams}}{\text{Total number of n-grams}}
\]

Another popular similarity measure in not only this vein of research but also in other fields is the Jaccard similarity index. This measure is calculated by taking the ratio of the number of common words and the total number of distinct words of both texts [7]. Moreover, other methods, such as Euclid, Manhattan, and Cosine, typically represent texts in the form of vectors and then compute text similarity using the distance between these vectors, as shown below:

\[
\text{Euclid distance} = \sqrt{\sum_{i=1}^{n} (X_i + Y_i)^2}
\]

\[
\text{Manhattan distance} = \sum_{i=1}^{n} |X_i - Y_i|
\]

\[
\text{Cosine similarity} = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2} \sqrt{\sum_{i=1}^{n} Y_i^2}}
\]

On the one hand, previous literature on the paraphrase identification task in Vietnamese also depends heavily on the string-based methods. For instance, Bach et al. [4] use nine string-based similarity measures combined with seven-string methods. For instance, Bach et al. [4] use nine string-based similarity measures combined with seven-string methods. These n-gram comparison of two texts is also considered a common algorithm. An n-gram is a sequence of n elements of a text sample. These n elements can be characters, phonemes, syllables, or words, depending on the tasks and applications. Alberto et al. define the formula to calculate the text similarity value using n-grams as follows [17]:

\[
\text{Similarity} = \frac{\text{Number of the same n-grams}}{\text{Total number of n-grams}}
\]

Another popular similarity measure in not only this vein of research but also in other fields is the Jaccard similarity index. This measure is calculated by taking the ratio of the number of common words and the total number of distinct words of both texts [7]. Moreover, other methods, such as Euclid, Manhattan, and Cosine, typically represent texts in the form of vectors and then compute text similarity using the distance between these vectors, as shown below:

\[
\text{Euclid distance} = \sqrt{\sum_{i=1}^{n} (X_i + Y_i)^2}
\]

\[
\text{Manhattan distance} = \sum_{i=1}^{n} |X_i - Y_i|
\]

\[
\text{Cosine similarity} = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2} \sqrt{\sum_{i=1}^{n} Y_i^2}}
\]

On the one hand, previous literature on the paraphrase identification task in Vietnamese also depends heavily on the string-based methods. For instance, Bach et al. [4] use nine string-based similarity measures combined with seven-string methods. These n-gram comparison of two texts is also considered a common algorithm. An n-gram is a sequence of n elements of a text sample. These n elements can be characters, phonemes, syllables, or words, depending on the tasks and applications. Alberto et al. define the formula to calculate the text similarity value using n-grams as follows [17]:

\[
\text{Similarity} = \frac{\text{Number of the same n-grams}}{\text{Total number of n-grams}}
\]

Another popular similarity measure in not only this vein of research but also in other fields is the Jaccard similarity index. This measure is calculated by taking the ratio of the number of common words and the total number of distinct words of both texts [7]. Moreover, other methods, such as Euclid, Manhattan, and Cosine, typically represent texts in the form of vectors and then compute text similarity using the distance between these vectors, as shown below:

\[
\text{Euclid distance} = \sqrt{\sum_{i=1}^{n} (X_i + Y_i)^2}
\]

\[
\text{Manhattan distance} = \sum_{i=1}^{n} |X_i - Y_i|
\]

\[
\text{Cosine similarity} = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2} \sqrt{\sum_{i=1}^{n} Y_i^2}}
\]

On the one hand, previous literature on the paraphrase identification task in Vietnamese also depends heavily on the string-based methods. For instance, Bach et al. [4] use nine string-based similarity measures combined with seven-string methods. These n-gram comparison of two texts is also considered a common algorithm. An n-gram is a sequence of n elements of a text sample. These n elements can be characters, phonemes, syllables, or words, depending on the tasks and applications. Alberto et al. define the formula to calculate the text similarity value using n-grams as follows [17]:

\[
\text{Similarity} = \frac{\text{Number of the same n-grams}}{\text{Total number of n-grams}}
\]

Another popular similarity measure in not only this vein of research but also in other fields is the Jaccard similarity index. This measure is calculated by taking the ratio of the number of common words and the total number of distinct words of both texts [7]. Moreover, other methods, such as Euclid, Manhattan, and Cosine, typically represent texts in the form of vectors and then compute text similarity using the distance between these vectors, as shown below:

\[
\text{Euclid distance} = \sqrt{\sum_{i=1}^{n} (X_i + Y_i)^2}
\]

\[
\text{Manhattan distance} = \sum_{i=1}^{n} |X_i - Y_i|
\]

\[
\text{Cosine similarity} = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2} \sqrt{\sum_{i=1}^{n} Y_i^2}}
\]
In all of these formulas, \( X \) and \( Y \) denote the two representation vectors of two corresponding segments of text.

Furthermore, while these three measures are considered methods within the string-based approach, they are still utilized as objective functions in other methods in other approaches, especially in machine learning models. Given its straight-forwardness implementation, the string-based approach can be found in applications that do not strictly rely on paraphrase identification. Since the processing occurs mainly on the input strings, these methods can be extended to the analyses of texts in a broad range of languages, including Vietnamese.

B. Corpus-based Approach

The methods of this approach exploit information from existing corpora to predict the similarity of input texts. The most common method in this approach is the Latent Semantic Analysis (LSA) [18], which assumes that words with similar meanings are co-occurrence in similar text segments. In this method, a matrix that represents the cohesion between words and text segments is first constructed from one or more given corpora. Then, its dimensions are reduced using the Singular Value Decomposition (SVD) technique. Finally, the similarity is calculated by the cosine similarity between the vectors which are the rows of the matrix.

Some methods use online corpora obtained from websites or search engine results. The advantage of these methods is that the extracted information is not only tremendously large, but it is also regularly updated. For instance, the Explicit Semantic Analysis (ESA) method uses Wikipedia articles as a data source to build representation vectors for texts [19]. Likewise, Cilibrasi et al. calculate the text similarity based on the statistics of results from the Google search engine for a given set of keywords [20].

In recent years, the deep learning technique on machine learning models has become more and more popular because of their efficiency in solving classification problems in various fields. In the paraphrase identification task, deep learning on the Siamese architecture for neural networks is the most popular method. The Siamese networks are dual-branch networks that share the same weights and are merged by an energy function. The Siamese architecture can learn the information about the differences between two input samples. Recently, the Siamese LSTM model is a well-known combination. Each input text is fed into an LSTM’s sequence. The outputs of the LSTM’s sequence are then merged by the Manhattan distance function in [9]. Meanwhile, Neculoiu et al. [2] use another feed-forward neural network which finetunes the output of LSTM layers before they are being merged by the cosine similarity function. Neculoiu et al. also use bi-directional LSTM’s sequence to exploit the bi-directional context instead of the single LSTM’s sequence as in [9].

The Google AI Research team then proposes the Bidirectional Encoder Representations from Transformers (BERT, 2018) model using Transformers as the model’s core [12]. These Transformers are fully connected, which allows it to outperform the state-of-the-art models at that time for some NLP downstream tasks. The model has achieved high results in over six tasks of NLP, including text similarity and paraphrase identification. We implement this BERT model in the proposed method of our study.

The introduction of the BERT model also leads to the emergence of the Siamese BERT model. Reimers et al.’s (2019) Sentence BERT (SBERT) model [21] uses the Siamese architecture to help fine-tune BERT with some corpora, targetting specific tasks to improve sentence representation for each task. The results of this work in downstream tasks are better than those of the representation vectors obtained from BERT.

Based on Transformer-XL, Yang et al.’s (2019) XLNet model [22] is argued to yield better results than BERT. The research team pointed out some shortcomings of the BERT model such as inconsistencies between training and the fine-tuning task and parallel independent word predictions. To overcome these drawbacks, they utilize both Permutation Language Modeling (PLM) and Transformer-XL [23].

Besides the pretrained model BERT, Alexis Conneau et al. (2020) introduce the XML-R model (XML-RoBERTa) [13], which is a generic cross lingual sentence encoder that obtains state-of-the-art results on many cross-lingual understanding (XLU) benchmarks. It is trained on 2.5T of filtered Common-Crawl data in 100 languages, and Vietnamese is one of the supported languages.

Based on RoBERTa, Dat Quoc Nguyen and Anh Tuan Nguyen (2020) introduce the PhoBERT model [14]. PhoBERT outperforms previous monolingual and multilingual approaches, obtaining new state-of-the-art performances on four downstream Vietnamese NLP tasks of Part-of-speech tagging, Dependency parsing, Named-entity recognition and Natural language inference.

While this is a potential approach for Vietnamese, the lack of high-quality and large corpora remains an obstacle to adopt these methods to the language.

C. Knowledge-based Approach

Methods in this approach exploit the linguistic knowledge from knowledge bases such as semantic networks, ontology, etc. WordNet [24], the most popular semantic network, is often used to extract linguistic knowledge at the lexical level to recognize the similarity between texts. Meanwhile, BabelNet [25] is a new semantic network that covers 284 languages. The main disadvantage that comes with BabelNet is that it only provides API in Java in its free edition.

There are six semantic measures, three of which are based on information content, while the remaining three are based on the connection length in the network. The former measures are proposed in Resnik (res) [26], Lin (lin) [27], and Jiang & Conrath (in) [28], while the latter ones can be found in Leacock & Chodorow (Ich) [29], Wu & Palmer (wup) [30], and Path Length (path). These measures are slightly different but can be interchangeable. Path Length is the most commonly used measure.

With the work of Le et al. [31] and BabelNet, we can apply this approach to Vietnamese. However, the Vietnamese semantic networks are not complete and still being updated, implying inconsistent results that would be yielded from the implementation of this approach alone.
D. Hybrid Approach

Mihalcea et al. (2006) combine two methods of the corpus-based approach with six measures of the knowledge-based approach to computing text similarity [32]. The results of the combination are better than those of each of the methods. Meanwhile, Li et al. (2006) calculate text similarity using the semantic vectors built from WordNet and Brown corpus [33]. Besides, the representation vector for word order also involves in the process of calculating the similarity of two sentences.

E. Vietnamese Sentence Paraphrase Identification

The work of Bach et al. in 2015 is among the first attempts to solve the paraphrase identification task for Vietnamese [4]. The important contribution of this work is the Vietnamese paraphrase corpus, vnPara, which is the first Vietnamese paraphrase corpus. The corpus is used to evaluate their proposed method, which is to construct a text representation vector from the combination of multiple similarity measures in the string-based approach for syntactic units such as words, syllables, part-of-speech (POS), nouns, verbs, etc. After this combination of measures and syntactic units with four machine learning methods, Bach et al. has achieved the highest results with the Support vector machine (SVM) when combining nine measures with seven syntax units.

Then, Nguyen-Son et al. (2018) propose a method that matches duplicate phrases and similar words [5]. First, this method matches all identical substrings of two sentences, and then eliminates stop words. Afterwards, WordNet is utilized to calculate the similarity for the remaining words. The experimental results of this method reveal that the vnPara corpus contains multiple paraphrase pairs that have a high rate of word overlap. Therefore, Nguyen-Son et al. introduce the construction of a new corpus, VNPC, which is argued to be more diverse than vnPara. In summary, most research on paraphrase identification for Vietnamese still rely heavily on the string-based approach, which is not ineffective in detecting semantic paraphrase identification.

III. PROPOSED METHOD

Since deep learning methods often require large corpora, the lack of Vietnamese paraphrase corpora creates challenges to researchers who plan to apply this technique to the language. Recently, the emergence of pre-trained models helps researchers overcome this obstacle. The pre-trained models such as BERT, XML-R and PhoBERT are most popular pre-trained models, especially for Vietnamese. Therefore, we take this advantage to construct our method.

Even though in theory, pre-trained models can effectively solve the paraphrase identification task for Vietnamese, we expect that this task can be improved with the addition of linguistic knowledge during the process. Devlin et al. [12] state that there are three ways to improve BERT, which are pre-training from scratch, fine-tuning the pre-trained model, and utilizing BERT as a feature extractor. However, linguistic knowledge cannot be used in the fine-tuning process, and training BERT and other pre-trained models from scratch is extremely costly. Therefore, feature extraction is the most plausible way to implement BERT in our method.

The proposed method follows a hybrid approach. In particular, it is a combination of the corpus-based approach and the knowledge-based approach to fully exploit the information gained from these two approaches.

We built three vectors for each input sentence:
- Feature vector achieved from pre-trained model.
- Semantic vector constructed by using WordNet.
- POS vector represents the POS of words in a sentence.

These three vectors then were joined together to form a sentence representation vector. There were two such vectors for two input sentences. These vectors were fed into a Siamese feed-forward neural network to train or predict. The overview of the proposed method is depicted in Fig. 1.

A. Preprocessing

The input pairs of sentences before being put into the main processing chain were normalized by regular expressions and Heuristic rules.


B. Features Extraction Using Pre-trained Model

Heretofore, simple word embedding models such as Word2Vec [34], GloVe [35], FastText [36], etc. are common methods used by many research groups to represent text in vector form. However, these models represent every word with a unique vector in all contexts. In contrast, the feature vector constructed from pre-trained model contains full information of the bi-directional context, thanks to Transformer blocks’ multi-head self-attention mechanisms and fully connected architecture.

The features extracted from pre-trained model were the output of the Transformer blocks. For instance, BERT-Base with 12 Transformer blocks provided 12 real vectors for each token and BERT-Large with 24 Transformer blocks provided 24 vectors. The dimension of each vector was the number of hidden units of each layer. There were 768 dimensions for BERT-Base and 1,024 dimensions for BERT-Large.

C. Semantic Vector Construction

We followed the method of [33] to construct semantic vectors for a sentence pair. A semantic vector contained information on the semantic relatedness of the words in these sentences. These vectors were constructed by using a semantic network and statistical information of a corpus. In our work, we use Vietnamese WordNet which was constructed by Le et al. in 2016 [31] and the statistical information from [37].

From the list of words of the sentences pair, a set of unique words was constructed. The order of these words was preserved in the order of the words in the sentences.

Let M be a two-dimensional matrix containing the relatedness of each pair of words. The matrix M has n rows corresponding to n words of the considered sentence and m columns corresponding to m words in the set of unique words. The relatedness between w1 (line r) and w2 (column c) is calculated using the formula:

\[
M(r,c) = \begin{cases} 
1 & \text{if } w1 = w2 \\
\text{PathLength}(w1,w2) & \text{if } w1 \neq w2 \\
0 & \text{if } w1 \text{ or } w2 \text{ is not in Wordnet}
\end{cases}
\]

(6)

where WordNet.PathLength(w1, w2) is the Path Length similarity in WordNet of word w1 and word w2. Each element of the lexical vector s is the maximum value on a column of the matrix M:

\[
s[c] = \text{Max}(M(r,c)), \ c = [1,m]
\]

(7)

Finally, the semantic vector semVec is calculated using the formula:

\[
\text{semVec}[c] = s[c] \times I(W1) \times I(W2)
\]

(8)

where W1 and W2 are words that have the greatest relatedness s[c] on column c; I(W1) and I(W2) are the information content of the two corresponding words. The information content I(w) of word w is calculated by the frequency of w in a corpus:

\[
I(w) = \frac{\log(p(w))}{\log(N + 1)} = 1 - \frac{\log(n + 1))}{\log(N + 1)}
\]

(9)

where p(w) is the relative frequency of w in a corpus, N is the number of words in the corpus and n is the frequency of the word w in the corpus.

Table IV shows the process of constructing the semantic vector for the first sentence in this sentences pair:

- Sentence 1: anh ấy là giáo viên (he is a teacher)
- Sentence 2: anh ấy là nhà giáo (he is an educator)

The semantic vector must be padded with zero-value to have a fixed length. According to the statistics in [37] about the average length of sentence (in words), we assume that the longest sentence may have a length of 50 words. Thus, we construct the semantic vector with a fixed length of 100.

D. Parts-of-speech (POS) Vector Construction

WordNet only accepts four simple parts-of-speech which are noun, verb, adjective, and adverb so that the semantic vector does not contain full information of the sentence’s parts-of-speech. Therefore, we also used the POS vector to provide more information to the model. To construct a POS vector, each word in a sentence was tagged with its parts-of-speech to create a list of parts-of-speech for each sentence. These POS lists were then represented as real vectors by using the FastText model [36]. To train this model, we used the Vietnamese Treebank corpus [38] with 10,000 POS tagged sentences. An output vector of the FastText model had a fixed length of 100.

E. Siamese Feed-forward Neural Network (SFFNN)

For each input sentence, the feature vector obtained from pre-trained model, the semantic vector, and the POS vector were concatenated to form the representation vector (Fig. 3). Sentence paraphrase identification task has an input of two sentences. Therefore, we generated two representation vectors.

To make the neural network learn the similarity between two sentences, we applied the Siamese architecture to the feed-forward neural network. Fig. 2 depicts a Siamese feed-forward neural network. The feed-forward neural network was constructed by multiple dense (fully connected) layers. The number of layers and hidden units will be presented in

---

**TABLE IV. PROCESS OF CONSTRUCTING THE SEMANTIC VECTOR FOR THE FIRST SENTENCE**

<table>
<thead>
<tr>
<th>vector</th>
<th>anh</th>
<th>ấy</th>
<th>là</th>
<th>giáo viên</th>
<th>nhà giáo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>I(anh)</td>
<td>I( ấy)</td>
<td>I( là)</td>
<td>I( giáo viên)</td>
<td>I( nhà giáo)</td>
</tr>
<tr>
<td>s</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.33</td>
</tr>
<tr>
<td>Semantic vector</td>
<td>0.1026</td>
<td>0.1474</td>
<td>0.0654</td>
<td>0.2597</td>
<td>0.1065</td>
</tr>
</tbody>
</table>

---
The activation function of hidden layers. The input layer was the ReLU function, and the output layer was the sigmoid function. The ReLU function is often used as the activation function in hidden layers due to its simplicity. Besides, the constant gradient of ReLUs results in faster learning and allows ReLU to overcome the defect of sigmoid function when the absolute value of the input is great.

The neural network was trained using the backpropagation algorithm and the training only stopped when the value of the energy function no longer changed. We used the Mean Squared Error (MSE) function as an error function for the gradient descent method. There are three similarity functions commonly used as energy functions in the text similarity task. They are Euclid, Manhattan, and Cosine. The study of Chopra et al. [39] shows that the Euclid and Cosine functions using the normalized function $l_2$ instead of $l_1$ in the similarity function can lead to undesirable plateaus in the overall objective function. Therefore, we used the Manhattan similarity function as the energy function in our model.

IV. Experiment

A. Corpora

The experiments in this paper were conducted on two main corpora: vnPara [4] and VNPC [5].

1) vnPara: VnPara has become a common Vietnamese paraphrase corpus in various studies [5][6]. To construct vnPara corpus, Bach et al. [4] first collected articles from online newspaper websites such as dantri.com.vn, vnexpress.net, thanhnien.com.vn, and so on. As shown in Table V, sentences extracted from the articles were paired if they have multiple words in common. These sentence pairs were labeled manually by two people.

2) VNPC: VNPC was constructed by Nguyen-Son et al. when they experimented with their proposed method in [5]. According to their experiment result, VNPC was argued to be more diverse than vnPara.

To build this corpus, first of all, the pairs of sentences were extracted from 65,000 pages of 15 Vietnamese news websites. Nguyen-Son et al. used their proposed method to measure the similarity of the obtained pairs. 3,134 candidates were selected using a predefined threshold. As shown in Table VI, these sentences formed paraphrase pairs, which were manually labeled.
was divided into 5 folds randomly to perform a 5-fold cross validation test. We also used the same metrics which were accuracy and F1 score as follows:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (11)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (12)
\]

\[
\text{Recall} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)
\]

where TP is true positive (a correct prediction of paraphrase), TN is true negative (a correct prediction of non-paraphrase), FP is false positive (a wrong prediction of paraphrase), and FN is false negative (a wrong prediction of non-paraphrase).

2) Configuration of Feed-forward Neural Network: The configuration of feed-forward neural network includes 12 hidden layers, 768 hidden units. We chose this configuration for experiments on the vnPara corpus and VNPC corpus.

C. Experimental Results

The experiments with our proposed method were performed with some different configurations of stop words, BERT’s output layer, and BERT’s output pooling strategies. We achieved the best result when testing our method with the configuration in which we kept stop words, used the second-to-last layer of pre-trained output, and utilized an average pooling strategy to get the feature vector. When experimenting with the Siamese LSTM model in the article [9], we used the pre-trained Vietnamese Word2Vec model of Vu et al. [40].

Tables VII and VIII show the results of experiments we conducted on vnPara and VNPC with several methods. The result of each method is presented by each row in the tables. Each method is evaluated by the accuracy and the F1 score. Each table shows the available results from previous studies for Vietnamese, the results of the Siamese LSTM model [9], the results of original pre-trained models, and the results of our method. The results of our method are presented in three rows according to three different configurations of additional vectors: adding semantic vector, adding the POS vector, and adding both semantic vector and POS vector.

We also compute the F1 score on each word overlap rate range with the proposed method as figures similar to the 4 and 5. The calculation of the F1 score is divided into two cases: paraphrase cases and non-paraphrase cases to assess the effect of non-trivial cases on model training.

Fig. 6 shows that the proposed method results above 80% on all word overlap rate ranges in the VNPC corpus. For the vnPara corpus, the proposed method’s result is below 80% for sentence pairs with a word overlap rate of 0.3 or less. In the word overlap rate range [0.1; 0.2], the proposed method...
TABLE VII. EVALUATION RESULTS OF DIFFERENT METHODS ON THE VNPARA CORPUS

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>F1 score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnPara [4]</td>
<td>89.10</td>
<td>86.70</td>
</tr>
<tr>
<td>Siamese LSTM [9]</td>
<td>65.64</td>
<td>64.29</td>
</tr>
<tr>
<td>BERT</td>
<td>73.14</td>
<td>73.56</td>
</tr>
<tr>
<td>Our method (using BERT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature vector (BERT) + Semantic vector</td>
<td>94.12</td>
<td>94.28</td>
</tr>
<tr>
<td>Feature vector (BERT) + POS vector</td>
<td>72.45</td>
<td>72.22</td>
</tr>
<tr>
<td>Feature vector (BERT) + Semantic vector + POS vector</td>
<td>94.27</td>
<td>94.38</td>
</tr>
<tr>
<td>XLM-R</td>
<td>74.57</td>
<td>75.22</td>
</tr>
<tr>
<td>Our method (using XLM-R)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature vector (XLM-R) + Semantic vector</td>
<td>93.58</td>
<td>93.76</td>
</tr>
<tr>
<td>Feature vector (XLM-R) + POS vector</td>
<td>75.12</td>
<td>75.51</td>
</tr>
<tr>
<td>Feature vector (XLM-R) + Semantic vector + POS vector</td>
<td>93.67</td>
<td>93.85</td>
</tr>
<tr>
<td>PhoBERT</td>
<td>76.33</td>
<td>75.80</td>
</tr>
<tr>
<td>Our method (using PhoBERT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature vector (PhoBERT) + Semantic vector</td>
<td>94.71</td>
<td>94.83</td>
</tr>
<tr>
<td>Feature vector (PhoBERT) + POS vector</td>
<td>75.83</td>
<td>75.20</td>
</tr>
<tr>
<td>Feature vector (PhoBERT) + Semantic vector + POS vector</td>
<td>94.86</td>
<td>94.97</td>
</tr>
</tbody>
</table>

TABLE VIII. EVALUATION RESULTS OF DIFFERENT METHODS ON THE VNPC CORPUS

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>F1 score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching duplicate phrases and similar words [5]</td>
<td>87.68</td>
<td>Not available</td>
</tr>
<tr>
<td>Siamese LSTM [9]</td>
<td>65.64</td>
<td>64.29</td>
</tr>
<tr>
<td>BERT</td>
<td>86.85</td>
<td>92.77</td>
</tr>
<tr>
<td>Our method (using BERT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature vector (BERT) + Semantic vector</td>
<td>86.72</td>
<td>92.74</td>
</tr>
<tr>
<td>Feature vector (BERT) + POS vector</td>
<td>86.14</td>
<td>92.37</td>
</tr>
<tr>
<td>Feature vector (BERT) + Semantic vector + POS vector</td>
<td>87.05</td>
<td>92.90</td>
</tr>
<tr>
<td>XLM-R</td>
<td>86.61</td>
<td>92.65</td>
</tr>
<tr>
<td>Our method (using XLM-R)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature vector (XLM-R) + Semantic vector</td>
<td>86.80</td>
<td>92.77</td>
</tr>
<tr>
<td>Feature vector (XLM-R) + POS vector</td>
<td>87.28</td>
<td>93.01</td>
</tr>
<tr>
<td>Feature vector (XLM-R) + Semantic vector + POS vector</td>
<td>87.70</td>
<td>93.30</td>
</tr>
<tr>
<td>PhoBERT</td>
<td>86.97</td>
<td>92.64</td>
</tr>
<tr>
<td>Our method (using PhoBERT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature vector (PhoBERT) + Semantic vector</td>
<td>87.45</td>
<td>93.12</td>
</tr>
<tr>
<td>Feature vector (PhoBERT) + POS vector</td>
<td>86.39</td>
<td>92.25</td>
</tr>
<tr>
<td>Feature vector (PhoBERT) + Semantic vector + POS vector</td>
<td>88.02</td>
<td>93.49</td>
</tr>
</tbody>
</table>

achieves F1 score of 28.57% for the vnPara corpus and 89.16% for the VNPC corpus.

In general, the F1 score is in Fig. 7 for the non-paraphrase cases of the VNPC is lower than vnPara corpus, due to the small number of non-paraphrase cases compared with the paraphrase cases in the VNPC corpus. The F1 score has a value of 0 in the range of overlap [0.7; 1] for the VNPC corpus and in [0.5; 1] for the vnPara corpus. This is because both corpora almost do not contain non-paraphrase cases in these two ranges. At the word overlap rate [0; 0.1), the test with the VNPC is 0% and the test with the vnPara reaches 99.71%.

To further demonstrate the universality of the proposed method’s improvement over the pre-trained model, an experiment was performed on another corpus. Apart from the vnPara and the VNPC, almost no Vietnamese paraphrase corpora has been published. Therefore, the proposed model will be tested further on a Vietnamese translation of a well-known paraphrase corpus MSRP [41]. The evaluation results are shown in Table IX.

D. Discussion

The Siamese LSTM model produces mediocre results, because the training process of this model requires great corpora. For English, this model is trained with over 300,000 sentence pairs and achieves an accuracy rate of 82.29%. Meanwhile, existing Vietnamese paraphrase corpora contain only about 3,000 pairs of sentences.
The pre-trained models are used passively, not yet tagger can entail the mistakes when building the semantic depends much on the POS tagger. The mistakes of the POS model itself contains some disadvantages. First of all, the translated documents.

This shows that our proposed method still achieves higher results with the F1 score of our proposed method are much better. Meanwhile, the re-proposed method achieves.

paraphrase cases on both corpora. Thus, it can be seen that the vnPara and VNPC. Fig. 5 clearly shows the lack of non-trivial cases having the word overlap rate is in the range [0.7; 1], though this corpus contains very few paraphrase sentence pairs variation across all word overlap ranges for the VNPC, even

proposed method for paraphrase cases does not have great

for the proposed method to achieve higher F1 score on the vnPara than on the VNPC when considering the paraphrase cases.

The number of duplicate sentences has a certain influence on the results of the proposed method. The number of duplicate sentences of VNPC is more than twice the number of duplicate sentences of vnPara. This means that the sentence diversity of vnPara is higher than that of VNPC, affecting the process of training deep learning models. This is also one of the reasons for our proposed method to achieve higher F1 score on the vnPara than on the VNPC when considering the paraphrase cases.

Fig. 6, 7 and the descriptions of these two figures partly show the importance of the non-trivial sentence pairs on the training process of the proposed method. The F1 score of our proposed method for paraphrase cases does not have great variation across all word overlap ranges for the VNPC, even though this corpus contains very few paraphrase sentence pairs in the range [0; 0.2] and [0.7; 1]. For the non-paraphrase cases having the word overlap rate is in the range [0.7; 1], our proposed method could not detect these cases on both vnPara and VNPC. Fig. 5 clearly shows the lack of non-trivial paraphrase cases on both corpora. Thus, it can be seen that the properties of the two corpora greatly affect the results that the proposed method achieves.

With testing on the Vietnamese-translated MSRP corpus, the results obtained from the proposed method are still higher than the results of the pre-trained models. Meanwhile, the results with the F1 score of our proposed method are much better. This shows that our proposed method still achieves higher results than the original pre-trained models when processing translated documents.

Although we achieve good results with our method, the model itself contains some disadvantages. First of all, the model requires big resources to operate, so it is not ready to work in practice. To build the semantic vector, the model depends much on the POS tagger. The mistakes of the POS tagger can entail the mistakes when building the semantic vector. The pre-trained models are used passively, not yet involved in the training process. Therefore, they have not been best exploited for this task.

V. CONCLUSION AND FUTURE WORK

The paraphrase identification task is a crucial core task of several NLP tasks and applications. There are various studies for popular languages but a few for Vietnamese. The great challenge in the research for the Vietnamese paraphrase identification task is the lack of good and large corpora. The emergence of the pre-trained models enable us to propose a novel method that does not require large corpora for training but is still highly effective. The proposed method uses three vectors: feature vector achieved from pre-trained model, semantic vector constructed by using WordNet, POS vector represents the POS of words in a sentence. They are joined to form a sentence representations vector that contains rich context information. Explicit linguistic knowledge helps the method yield 94.97% F1-score on the VnPara corpus and 93.49% F1-score on the VNPC corpus, which is better than the pre-trained models for the paraphrase identification task in Vietnamese. These results also show that using a pre-trained model is a feasible way for studies of text similarity as well as other NLP tasks in resource-poor languages such as Vietnamese.

Although the method proposed in this paper achieves positive results, we realize that there are still potential improvements to achieve better results. We plan to fine-tune the pre-trained models in the training process to make the pre-trained models learn information from the input samples to have better sentence representation vectors. Using linguistic knowledge has proved to be effective. However, the resources for the proposed method to work are quite high. Hence, we need to create a method that uses fewer resources but still guarantees high accuracy rates.

Whereas the proposed method can solve the corpora lacking problem for deep learning, it is still necessary to have Vietnamese paraphrase corpora for fine-tuning, improvement, or evaluation. Meanwhile, the two existing Vietnamese paraphrase corpora still have some shortcomings such as class imbalance, the lack of non-trivial instances, and various duplicate sentences, etc. Therefore, the need to construct good-quality Vietnamese paraphrase corpora remains as pressing as ever.

ACKNOWLEDGMENT

This research is funded by University of Science, VNU-HCM under grant number CNTT 2020-07. This research is supported by Computational Linguistics Center, University of Science, Vietnam National University, Ho Chi Minh City, and uses the Vietnamese Treebank, which was developed by the VLSP project.

REFERENCES

