Paraphrase Identification in Vietnamese Documents

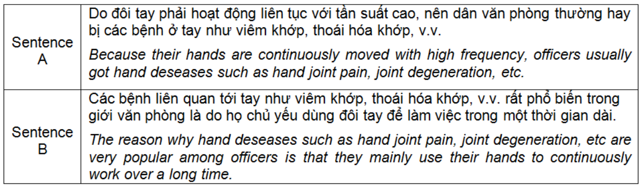
Ngo Xuan Bach*†, Tran Thi Oanh‡, Nguyen Trung Hai*, Tu Minh Phuong*†
*Department of Computer Science, Posts and Telecommunications Institute of Technology, Vietnam
{bachnx,haint,phuongtm}@ptit.edu.vn
†Machine Learning & Applications Lab, Posts and Telecommunications Institute of Technology, Vietnam
‡International School, Vietnam National University, Hanoi
oanhtt@isvnu.vn

***Abstract*—In this paper, we investigate the task of paraphrase identification in Vietnamese documents, which identify whether two sentences have the same meaning. This task has been shown to be an important research dimension with practical applications in natural language processing and data mining. We choose to model the task as a classification problem and explore different types of features to represent sentences. We also introduce a paraphrase corpus for Vietnamese, vnPara, which consists of 3000 Vietnamese sentence pairs. We describe a series of experiments using various linguistic features and different machine learning algorithms, including Support Vector Machines, Maximum Entropy Model, Naive Bayes, and k-Nearest Neighbors. The results are promising with the best model achieving up to 90% accuracy. To the best of our knowledge, this is the first attempt to solve the task of paraphrase identification for Vietnamese.**

***Keywords*—Paraphrase Identification, Semantic Similarity, Support Vector Machines, Maximum Entropy Model, Naive Bayes Classification, K-Nearest Neighbor.**

I. Introduction

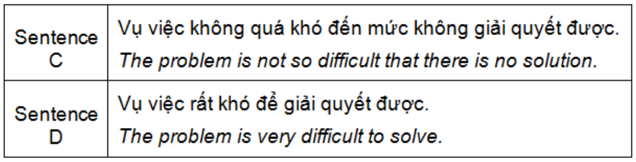
Paraphrase identication is the task of deciding whether two text fragments are paraphrases of each other. In this paper, we focus on sentential paraphrases. To give an example, we show below a pair of sentences from our manually built corpus, *vnPara* [1], in which sentence A is a paraphrase of sentence B and vice versa (see Figure 1).



Sentence A	Do đôi tay phải hoạt động liên tục với tần suất cao, nên dân văn phòng thường hay bị các bệnh ở tay như viêm khớp, thoái hóa khớp, v.v. *Because their hands are continuously moved with high frequency, officers usually got hand deseases such as hand joint pain, joint degeneration, etc.*
Sentence B	Các bệnh liên quan tới tay như viêm khớp, thoái hóa khớp, v.v. rất phổ biến trong giới văn phòng là do họ chủ yếu dùng đôi tay để làm việc trong một thời gian dài. *The reason why hand deseases such as hand joint pain, joint degeneration, etc are very popular among officers is that they mainly use their hands to continuously work over a long time.*

Fig. 1: An example of two Vietnamese paraphrase sentences and its translation into English.

Paraphrase identification is important in a number of applications such as text summarization, question answering, machine translation, natural language generation, and plagiarism detection. For example, detecting paraphrase sentences would help a question answering system to increase the likelihood of finding the answer to the user's question. As a further example, in text summarization, a paraphrase identification system can be used to avoid adding redundant information.



Sentence C	Vụ việc không quá khó đến mức không giải quyết được. *The problem is not so difficult that there is no solution.*
Sentence D	Vụ việc rất khó để giải quyết được. *The problem is very difficult to solve.*

Fig. 2: An example of two Vietnamese non-paraphrase sentences and its translation into English.

Paraphrase identication is not an easy task. Considering the first sentence pair of sentence A and sentence B above, this pair is a paraphrase although the two sentences only share a few words, while the second one (sentence C and sentence D in Figure 2) is not a paraphrase even though the two sentences contain almost all the same words.

Paraphrase identication has been extensively explored for documents written in English and some other popular languages, most notably by Kozareva and Montoyo [9], Fernando and Stevenson [7], etc. However, to the best of our knowledge, there is no effort done for Vietnamese. A main reason might be the lack of annotated corpora.

In this paper, we focus on Vietnamese paraphrase identification, in which we model the task as a binary classification problem and train a statistical classifier to solve it. Our method employs string similarity measures applied to different abstractions of the input sentence pair. We investigate the task regarding both learning model and linguistic feature aspects. The contributions of this paper are two-fold:

1) We build a corpus annotated with paraphrase identification labels by extracting paraphrased sentences in online articles referring to the same topics, followed by manual annotation and statistical verification.
2) We investigate the impact of different features, including linguistic ones on the classification performance using different machine learning methods.

The rest of the paper is organized as follows. Section 2 presents previous research on paraphrase identification. Section 3 introduces in detail our proposed system for Vietnamese paraphrase identification. Section 4 describes our corpus and experimental setups. Experimental results are presented in Section 5. In Section 6, we conduct an analysis of our system's misclassification on the vnPara corpus. Finally, Section 7 concludes the paper and discusses our plans for the future.

[1] This vnPara corpus will be made available by the authors at publication time.

978-1-4673-8013-3/15 $31.00 © 2015 IEEE
DOI 10.1109/KSE.2015.37

II. Related Work

Various studies on paraphrase identification have been conducted in different languages, especially in English. Finch et al., [8] investigate the utility of applying standard MT evaluation metrics, including BLEU [19], NIST [6], WER [17], and PER [10], to building classifiers to predict paraphrase relations. Mihalcea et al., [16] use pointwise mutual information, latent semantic analysis, and WordNet to compute an arbitrary text-to-text similarity metric. Wan et al., [25] show that dependency-based features in conjunction with bigram features improve upon the previously published work to give us the best reported classification accuracy on the PAN corpus [15]. Kozareva et al., [9] propose a machine learning approach based on lexical and semantic information, e.g. a word similarity measure based on WordNet. They also model the problem of paraphrasing as a classification task. Their model uses a set of linguistic attributes and three different machine learning algorithms, i.e. Support Vector Machines, k-Nearest Neighbors, and Maximum Entropy, to induce classifiers. The classifiers are built in a supervised manner from labeled training data.

Fernando and Stevenson [7] present an algorithm for paraphrase identification which makes extensive use of word similarity information derived from WordNet. Rus et al. [20] adapt a graph-based approach for paraphrase identication by extending a previously proposed method for the task of text entailment. Das and Smith [5] introduce a probabilistic model which incorporates both syntax and lexical semantics using quasi-synchronous dependency grammars for identifying paraphrases. Socher et al., [22] introduce a method for paraphrase detection based on recursive autoencoders. This unsupervised method is based on a novel unfolding objective and learn feature vectors for phrases in syntactic trees. Madnani et al., [15] present an investigation of the impact of MT metrics on the paraphrase identication task. They examine 8 different MT metrics, including BLEU, NIST, TER, TERP, METEOR, SEPIA, BADGER, and MAXSIM, and show that a system using nothing but some MT metrics can achieve state-of-the-art results on this task.

Recently, Bach et al., [1] present a new method named EDU-based similarity, to compute the similarity between two sentences based on elementary discourse units. They also show the relation between paraphrases and discourse units, which plays an important role in paraphrasing.

All previous works, except for Nguyen et al. [18], were performed for English and other popular languages such as Chinese, Japanese, and Korea. Nguyen et al. [18] present a method for measuring semantic similarity of two Vietnamese sentences based on concepts. The overall semantic similarity is a linear combination of word-to-word similarity, word-order similarity, and concept similarity. Their work, however, focuses on measuring semantic similarity, not on predicting paraphrases. Compared with previous work, our work makes the first effort to solve the task of paraphrase identification for Vietnamese. In order to conduct experiments, we also build a corresponding corpus for this task, which includes 3000 Vietnamese sentence pairs.

III. Our method

In this section, we present our method for Vietnamese paraphrase identification. The main idea of the method is to calculate the similarity between two sentences based on various abstractions of the input sentences. The method is described in more detail as follows:

In general, given a set of n labelled sentence pairs $\{\langle S_{1,1}, S_{1,2}, y_1\rangle, \ldots, \langle S_{n,1}, S_{n,2}, y_n\rangle\}$, where $S_{i,1}$ and $S_{i,2}$ are the i^{th} sentence pair, y_i receives value 1 if the two sentences are paraphrases, 0 otherwise. Each sentences pair $\langle S_{i,1}, S_{i,2}\rangle$ is converted to a feature vector $\vec{v_i}$, whose values are scores returned by similarity measures that indicate how similar $S_{i,1}$ and $S_{i,2}$ are at various levels of abstraction. The vectors and the corresponding categories $\{\langle \vec{v_1}, y_1\rangle, \ldots, \langle \vec{v_n}, y_n\rangle\}$ are given as input to the supervised classifiers, which learns how to classify new vectors $\vec{v}$, corresponding to unseen pairs of sentences $\langle S_1, S_2\rangle$.

In this paper, nine string similarity measures are used, including Levenshtein distance (edit distance), Jaro-Winkler distance, Manhattan distance, Euclidean distance, co-sine similarity, n-gram distance (with $n = 3$), matching coefcient, Dice coefficient, and Jaccard coefficient [14]. For each pair of input sentences, we form seven new string pairs $\langle s_1^i, s_2^i\rangle$ which correspond to seven abstraction levels of the two input sentences. The seven new sentence pairs are:

1) Two strings consisting of the original syllables[2] of S_1 and S_2, respectively, with the original order of the tokens maintained.
2) As in the previous case, but now the tokens are replaced by their words.
3) As in the previous case, but now the words are replaced by their part-of-speech tags.
4) Two strings consisting of nouns, verbs, and adjectives of S_1 and S_2, as identified by a POS tagger, with the original order of the nouns, verbs and adjectives maintained.
5) As in the previous case, but keep only nouns.
6) As in the case 4, but keep only verbs.
7) As in the case 4, but keep only adjectives.

In total, 9 string similarity measures combined with 7 string pairs give 63 values.

Figure 3 presents the framework of our method to solve the Vietnamese paraphrase identification task. The framework consists of two main phases: the training and the testing phase. In the training phase, labeled sentence pairs are preprocessed and are used to extract corresponding feature vectors by calculating nine string similarity measures applied to seven different abstract levels of the input sentences (as shown above). These feature vectors are then used to train a model using some strong machine learning methods. In the testing phase, the obtained model is used to classify a raw sentence pair after preprocessed and feature-extracted as in the training step into paraphrase or non-paraphrase labels.

[2] Unlike English words, words in Vietnamese cannot be delimited by white spaces. Vietnamese words may consist of one or more syllables, and syllables are delimited by white spaces.

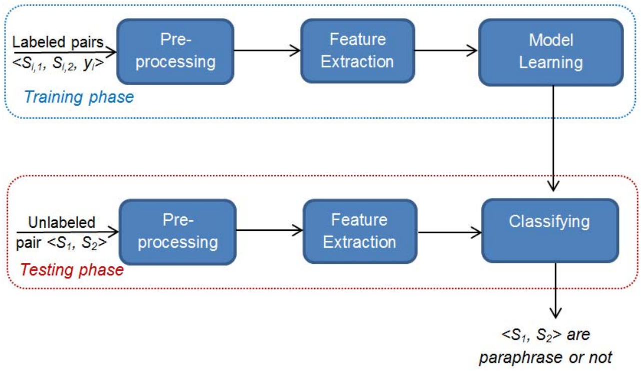


Fig. 3: A proposed method to solve Vietnamese paraphrase identification.

A. Similarity Measures

We now describe the nine string similarity measures used in this paper. The measures are applied to a string pair of $\langle s_1, s_2 \rangle$.

1) Jaro-Winkler distance: The Jaro-Winkler distance [26] is a measure of similarity between two strings. The higher the Jaro-Winkler distance for two strings is, the more similar the strings are. The Jaro distance d_j of two given strings s_1 and s_2 is computed as follows:

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases}$$

where m is the number of matching characters; and t is half the number of *transpositions*. Jaro–Winkler distance uses a prefix scale p which gives more favourable ratings to strings that match from the beginning for a set prefix length l . Given two strings s_1 and s_2 , their Jaro-Winkler distance d_w is:

$$d_w = d_j + (lp(1 - d_j)), \text{where:}$$

- d_j is the Jaro distance for strings s_1 and s_2.
- l is the length of common prefix at the start of the string up to a maximum of 4 characters.
- p is a constant scaling factor for how much the score is adjusted upwards for having common prefixes. p should not exceed 0.25, otherwise the distance can become larger than 1. The standard value for this constant in Winkler's work is $p = 0.1$.

2) Levenshtein Distance: The Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two strings is the minimum number of single word (token) edits (i.e. insertions, deletions or substitutions) required to change one string into the other. It is named after Vladimir Levenshtein, who considered this distance in 1966 [11].

3) Manhattan Distance: The Manhattan distance [13] function computes the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components.

The formula for this distance between a point $X = (X_1, X_2, ...)$ and a point $Y = (Y_1, Y_2, ...)$ is:

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

where n is the number of distinct words (tokens) that occur in any of the two strings, and x_i and y_i show how many times each one of these distinct words occurs in each of these two strings, respectively.

4) Euclidean Distance: Similarly to previous case, we also represent two strings in n-dimensional vector space and the Euclidean distance [13] between two strings is calculated as follows:

$$L_2(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

5) Cosine Similarity: Cosine similarity [13] is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. It is defined as follows:

$$cos(\vec{x}, \vec{y}) = \frac{\vec{x}.\vec{y}}{\|\vec{x}\| \times \|\vec{y}\|}$$

In our system $\vec{x}$ and $\vec{y}$ are as above, except that they are binary, i.e., x_i and y_i are 1 or 0, depending on whether or not the corresponding word (or tag) occurs in the first or the second string, respectively.

6) N-gram distance: This is the same as the Manhattan distance, but instead of words we use all the (distinct) character n-grams in two strings. In experiments, we used $n = 3$.

7) Matching coefficient: This simple matching coefficient counts how many common words (tags) that two strings have.

8) Dice coefficient: The Dice coefficient [13] is a statistic used for comparing the similarity of two samples and is calculated as follows:

$$\frac{2 \times |X \cap Y|}{|X| + |Y|}$$

where X and Y are the sets of (unique) words (or tags) of two strings, respectively.

9) Jaccard Coefficient: The Jaccard coefficient [13] measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where again X and Y are as in the Dice coefficient.

IV. Data and Experimental Setup

A. Data

To build the Vietnamese paraphrase corpus, we first collected articles from online news websites such as dantri.com.vn, vnexpress.net, thanhnien.com.vn, etc. Each document was preprocessed through natural language processing steps including, sentence separator (VnSentDetector[3]), word

[3] http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnSentDetector

segmenter(VnTokenizer[4]), and POS tagger (VnTagger[5]). After that, we extracted pairs of two sentences in two different documents, which refer to the same topics, if the two sentences contain several similar words. Obtained sentence pairs were then labeled as paraphrases or non-paraphrases depending on whether they bear the almost same meaning or not. We had two people performing this labeling step. They worked independently. Then, we used Cohen's kappa coefficient [3] to measure inter-annotator agreement for labeling paraphrases between two annotators. The Cohen's kappa coefficient was calculated as follows:

$$k = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}$$

where $Pr(a)$ is the relative observed agreement between two annotators, and $Pr(e)$ is the hypothetical probability of chance agreement. The Cohens kappa coefficient of our corpus was 0.9 in this case. This means that the agreement between these two annotators was high, and could be interpreted as almost perfect agreement. As a result, a complete corpus was built. This corpus includes 3000 sentence pairs, 1500 of which were labeled as paraphrases (labeled as 1) and the other 1500 sentence pairs were not (labeled as 0).

B. Experimental Setup

1) The method for conducting experiments: We randomly divided the corpus into 5 folds and conducted 5-fold cross-validation test. We report results using two widely-used performance metrics, which are accuracy, and the F_1 score as follows:

$$Accuracy = \frac{\text{\#of correctly identified pairs}}{\text{\#of pairs}}, \text{and}$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

where,

$$Precision = \frac{\text{\#of correctly identified pairs}}{\text{\#of identified pairs}}, \text{and}$$

$$Recall = \frac{\text{\#of correctly identified pairs}}{\text{\#of gold pairs}}.$$

The accuracy was the percentage of correct predictions over all the test set, while the F_1 score was computed only based on the paraphrase sentence pairs (label 1). All scores were averaged over five folds.

2) Feature Selection: Our feature extraction method is based on sentences pairs. Their values are scores returned by similarity measures that indicate how similar the two input sentences are at various levels of sentence abstraction. Corresponding to 7 representations of the input sentences, which are based on sets of words, syllables, part-of-speech tags, nouns, verbs, adjectives, and a combination of nouns, verbs, and adjectives, we form 7 kinds of feature sets. In other word, each kind of features corresponds to a type of representations of the input sentence. For each kind of features, we calculate 9 similarity measures as described in Section 3.1.

TABLE I: The experimental results using different feature sets

Feature Sets	Accuracy (%)	F_1
(1) words	**89.03**	**87.06**
(2) syllables	88.73	86.71
(3) Part-of-speech tags	88.63	85.96
(4) Combination of n, v, and a	88.33	86.38
(5) nouns	85.90	83.89
(6) verbs	82.50	81.13
(7) adjectives	75.37	72.75

TABLE II: The experimental results using different combinations of feature sets

Feature Sets	Accuracy (%)	F_1
(1)	89.00	86.71
(1)+(2)	88.97	**87.06**
(1)+(2)+(3)	88.93	87.01
(1)+(2)+(3)+(4)	**89.10**	86.77
(1)+(2)+(3)+(4)+(5)	88.90	86.89
(1)+(2)+(3)+(4)+(5)+(6)	88.83	86.90
(1)+(2)+(3)+(4)+(5)+(6)+(7)	88.77	86.69

3) Learning Algorithms: To conduct experiments, we used four classification algorithms, including SVM [24], MEM [2], k-NN [21], and Naive Bayes classifiers [21]. These four methods are also successfully applied for this task in other languages.

4) Experimental Purposes: In experiments, we performed three types of experiments and their purposes are as follows:

- To investigate the effectiveness of each kind of features.
- To conduct feature selection in order to find the best feature set.
- To investigate different machine learning methods.

In the first two experiments, we chose the SVM as the learning method.

V. Experimental Results

A. Different feature types

This section describes experimental results of paraphrase identification using seven different feature types separately. Table I presents the experimental results using these 7 feature sets. The results show that features extracted based on the word representation of sentence pairs yielded the highest performance. The second best results were achieved with features extracted based on the syllables representation. This is reasonable because the words and syllables keep original meaning of the input sentences.

B. Combinations of different feature sets

Based on the experimental results of the previous section, we gradually combined feature sets using the performance of each feature set. Feature sets which yielded the higher performance will have higher priorities. Table II presents the experimental results using these combinations.

The results show that the combination of all representation levels of sentences pairs, including words, syllables, pos tags, nouns, verbs, and adjectives, yielded the highest performance.

[4] http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnTokenizer

[5] http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnTagger

TABLE III: The experimental results using different machine learning methods

ML Methods	Accuracy (%)	F_1
SVM	89.10	86.77
Maximum Entropy	88.60	86.01
Naive Bayes	88.59	85.62
k-NN (k = 10)	88.43	85.82
k-NN (k = 5)	87.93	86.33

TABLE IV: Some statistics of the experimental results on the corpus

#of models predicted correctly	#of sentence pairs	Percentage(%)
0	168	5.6
1	115	3.8
2	30	1.0
3	27	0.9
4	29	1.0
5	95	3.2
6	634	21.1
7	1902	63.4

We achieved 89.10% accuracy and 86.77% in the F_1 score. This means that the more information the model integrated, the better its performance was.

C. *Different Machine Learning Methods*

We also conducted experiments to investigate performance of different machine learning methods for this task. We chose the combination of feature sets yielding the highest performance according to the previous experimental results. That was the combination of feature sets of (1)+(2)+(3)+(4). Table III presents the experimental results using this combination on different machine learning methods. Here is the list of the software tools used in this experimental setting:

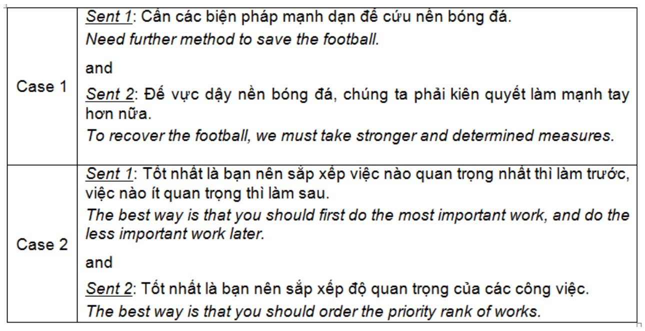
- For the SVM method, we chose LibSVM[6] written by Chih-Chung Chang and Chih-Jen Lin [4].
- For the three remaining classifying methods, we chose WEKA software[7] to perform experiments.

Experimental results showed that the SVM method performed slightly better than other learning methods, including MEM, Naive Bayes, and K-Nearest Neighbor, on the Vietnamese paraphrase identification task.

VI. Error Analysis

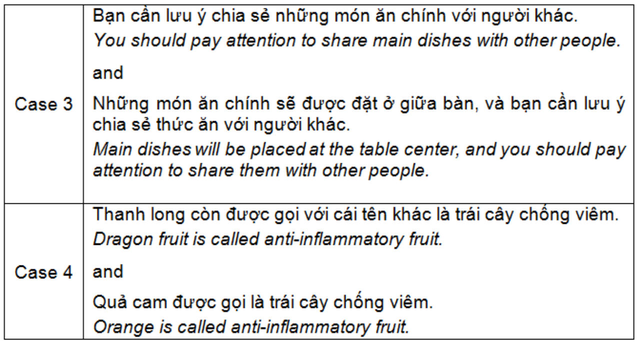
In this section, we analyze main types of errors that our system made. First, we perform statistics using 7 kinds of base models, which corresponds to 7 different feature sets (as presented in Section 5.1). Table IV presents some figures of:

- With each sentence pair in corpus, how many models among seven base models produced a correct output?
- How many sentence pairs were predicted correctly by at least one base model? And therefore, how many sentence pairs were unable to be predicted correctly by base models?



Case 1	*Sent 1*: Cần các biện pháp mạnh dạn để cứu nền bóng đá. *Need further method to save the football.* and *Sent 2*: Để vực dậy nền bóng đá, chúng ta phải kiên quyết làm mạnh tay hơn nữa. *To recover the football, we must take stronger and determined measures.*
Case 2	*Sent 1*: Tốt nhất là bạn nên sắp xếp việc nào quan trọng nhất thì làm trước, việc nào ít quan trọng thì làm sau. *The best way is that you should first do the most important work, and do the less important work later.* and *Sent 2*: Tốt nhất là bạn nên sắp xếp độ quan trọng của các công việc. *The best way is that you should order the priority rank of works.*

Fig. 4: Examples of two types of errors caused by our system that wrongly predict paraphrase sentences as non-paraphrases.



Case 3	Bạn cần lưu ý chia sẻ những món ăn chính với người khác. *You should pay attention to share main dishes with other people.* and Những món ăn chính sẽ được đặt ở giữa bàn, và bạn cần lưu ý chia sẻ thức ăn với người khác. *Main dishes will be placed at the table center, and you should pay attention to share them with other people.*
Case 4	Thanh long còn được gọi với cái tên khác là trái cây chống viêm. *Dragon fruit is called anti-inflammatory fruit.* and Quả cam được gọi là trái cây chống viêm. *Orange is called anti-inflammatory fruit.*

Fig. 5: Examples of two types of errors caused by our system that wrongly predict non-paraphrase sentences as paraphrases.

We also observe the output of the final system (the best model which uses a SVM classifier as the machine learning method and the combination of the first four types of feature sets) and analyze errors based on two main types: the first type contains some main causes that lead the system wrongly identifies paraphrase sentence pairs as non-paraphrases, and the second type lists some main causes that lead the system wrongly identifies non-paraphrase sentence pairs as paraphrases.

A. *Paraphrases (predicted as non-paraphrases)*

- *Using totally different words*: two sentences in a pair using very different words (or rewritten using lots of new words). An example is the case 1 as shown in Figure 4.
- *Complex or compound sentences*: rewrite a sentence using multiple clauses. An example is the case 2 as shown in Figure 4.
- *Typing errors*: There exist some sentences in the corpus, that contain typos and spelling errors that make the system cannot judge correctly.

B. *Non-paraphrases (predicted as paraphrases)*

- *Containing*: These sentence pairs consist of two sentences in which one of them contains the other one but has additional parts. This is similar to the relation

[6] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[7] http://www.cs.waikato.ac.nz/ml/weka/

of textual entailment. An example is given by the case 3 in Figure 5.

- *Misleading lexical overlap*: These sentence pairs consist of two sentences which have large lexical overlap. They share a lot of words and contain only a few different words. However, these few different words make the meaning change. An example is given by the case 4 in Figure 5.

Therefore, the system needs to use more semantic features such as ontology, dictionary of synonyms and asynonym, etc.

VII. Conclusion and Future Work

Although the role of paraphrase identification has been proved to be important in many NLP and DM applications for English and other popular languages, there exists no research on this field for Vietnamese. This paper marks our first work to this interesting research direction.

Throughout the paper, we have presented a method to recognize paraphrases given pairs of Vietnamese sentences. The method uses nine string similarity measures applied to seven different abstract levels of the input sentences. We also introduced a corpus built manually, which consists of 3000 paraphrase-labeled Vietnamese sentence pairs to conduct experiments. Experiments were performed in a supervised manner, in which we combine different feature sets using strong machine learning methods. The experimental results showed that the proposed method got the highest performance of 89.10% accuracy, and 86.77% in the F_1 score when using the combinations of four feature sets (including words, syllables, pos tags, and the combination of nouns, verbs, and adjectives) and a single SVM classifier.

To improve the performance of our method, in the future we plan to integrate more features to include semantic information from synonym dictionaries. Another aspect is that our current method works at lexical levels, therefore, we also add some other features that operate on the grammatical relations such as the information extracted from dependency tree, etc. Further improvements may be possible by including in our system additional features such as MT scores, Brown clustering information, exploit the resources from other languages, etc.

Acknowledgment

This work was partially supported by "KHCN_2015_09 Research Grant", International School, Vietnam National University, Hanoi.

References

[1] N.X. Bach, N.L. Minh, A. Shimazu. Exploiting discourse information to identify paraphrases. *Journal of Expert systems with applications*, Volume 41, Issue 6, pp. 2832–2841, 2014.

[2] A.L. Berger, V.J.D. Pietra, S.A.D. Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, Volume 22, 1996.

[3] J. Carletta. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Journal of Computational Linguistics*, Volume 22 Issue 2, pp. 249–254, 1996.

[4] C. Chih-Chung and L. Chih-Jen. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (ACM TIST)*. 2(3):1–27.

[5] D. Das, N.A. Smith. Paraphrase Identification as Probabilistic Quasi-synchronous Recognition. In *Proceedings of ACL-IJCNLP*, pp. 468–476, 2009.

[6] G. Doddington. Automatic Evaluation of Machine Translation Quality using N-gram Co-occurrence Statistics. In *Proceedings of HLT*, pp. 138–145, 2002.

[7] S. Fernando, M. Stevenson. A Semantic Similarity Approach to Paraphrase Detection. In *Proceedings of the computational linguistics UK (CLUK)*, 2008.

[8] A. Finch, Y.S. Hwang, E. Sumita. Using Machine Translation Evaluation Techniques to Determine Sentence-level Semantic Equivalence. In *Proceedings of IWP Workshop*, 2005.

[9] Z. Kozareva, A. Montoyo. Paraphrase Identification on the Basis of Supervised Machine Learning Techniques. In *Proceedings of the fifth international conference on natural language processing (FinTAL)*, pp. 524–533, 2006.

[10] G. Leusch, N. Ueffing, H. Ney. A Novel String-to-string Distance Measure with Applications to Machine Translation Evaluation. In *Proceedings of MT Summit*, pp. 182–190, 2003.

[11] V.I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Doklady Akademii Nauk SSSR, 163(4), pp. 845–848, 1965 (Russian). English translation in Soviet Physics Doklady, 10(8):707–710, 1966.

[12] M. Lintean, V. Rus. Dissimilarity Kernels for Paraphrase Identification. In *Proceedings of (FLAIRS)*, pp. 263–268, 2011.

[13] P. Malakasiotis, I. Androutsopoulos. Learning Textual Entailment using SVMs and String Similarity Measures. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Association for Computational Linguistics, pp. 42-47, 2007.

[14] P. Malakasiotis. Paraphrase Recognition Using Machine Learning to Combine Similarity Measures. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pp. 27–35, 2009.

[15] N. Madnani, J. Tetreault, M. Chodorow. Re-examining Machine Translation Metrics for Paraphrase Identification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 182–190, 2012.

[16] R. Mihalcea, C. Corley, C. Strapparava. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In *Proceedings of AAAI*, pp. 775–780, 2006.

[17] S. Niessen, F. Och, G. Leusch, H. Ney. An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In *Proceedings of LREC*, 2000.

[18] H.T. Nguyen, P.H. Duong, V.T. Vo. Vietnamese Sentence Similarity Based on Concepts. In *Proceedings of International Conference on Computer Information Systems and Industrial Management Applications*, pp. 243–253, 2014.

[19] K. Papineni and S. Roukos and T. Ward and W.J. Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*, pp. 311-318, 2002.

[20] V. Rus, P.M. McCarth, M.C. Lintean, D.S. McNamara, A.C. Graesser, Paraphrase Identification with Lexico-Syntactic Graph Subsumption. In *Proceedings of FLAIRS*, pp. 201–206, 2008.

[21] A. Smola, S.V.N. Vishwanathan. *Introduction to Machine Learning*, Cambridge University Press, 2008.

[22] R. Socher, E.H. Huang, J. Pennington, Y.Ng. Andrew, C.D. Manning. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In *Proceedings of NIPS*, pp. 801-809, 2011.

[23] N.M.J. Tetreault. Re-examining Machine Translation Metrics for Paraphrase Identification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 182-190, 2012.

[24] V.N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[25] S. Wan, R. Dras, M. Dale, C. Paris. Using Dependency-based Features to Take the Para-farce out of Paraphrase. In *Proceedings of the 2006 Australasian language technology workshop*, pp. 131-138, 2006.

[26] W.E. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods (American Statistical Association)*, pp. 354-359, 1990.