A Hybrid Approach to Vietnamese Word Segmentation

Tuan-Phong Nguyen
Faculty of Information Technology
VNU University of Engineering and Technology
No. 144 Xuan Thuy Street
Dich Vong Hau Ward, Cau Giay District
Hanoi, Vietnam
Email: phongnt_570@vnu.edu.vn

Anh-Cuong Le*
Faculty of Information Technology
Ton Duc Thang University
No. 19 Nguyen Huu Tho Street
Tan Phong Ward, District 7
Ho Chi Minh City, Vietnam
Email: leanhcuong@tdt.edu.vn

***Abstract*—Word segmentation is the very first task for Vietnamese language processing. Word-segmented text is the input of almost other NLP tasks. This task faces some challenges due to specific characteristics of the language. As in many other Asian languages such as Japanese, Korean and Chinese, white spaces in Vietnamese are not always used as word separators and a word may contain one or more syllables. In this paper, we propose an efficient hybrid approach to detect word boundary for Vietnamese texts using logistic regression as a binary classifier combining with longest matching algorithm. First, longest matching algorithm is used to catch words that contain more than two syllables in input sentence. Next, the system utilizes the classifier to determine the boundary of 2-syllable words and proper names. Then, the predictions having low confidence conducted by the classifier are verified by a dictionary to get the final result. Our system can achieve an F-measure of 98.82% which is the most accurate result for Vietnamese word segmentation to the best of our knowledge. Moreover, the system also has a high speed. It can run word segmentation for nearly 34k tokens per second.**

I. Introduction

In linguistics, word is the smallest meaningful unit of speech that can stand by itself. Vietnamese, an Austroasiatic language, uses a Latin alphabet with additional diacritics and certain letters. However, unlike many occidental languages using Latin alphabets, Vietnamese has similar characteristics to other East Asian languages such as Japanese, Korean, Chinese and Thai in which white spaces are not always word separators and a word may consist of more than one syllable with many ambiguous cases. This leads to some challenges in Vietnamese word segmentation.

Studies on Vietnamese word segmentation used either dictionary-based algorithms, statistical models or hybrid approaches. Recent studies using hybrid approaches such as [1], [2], [3] can provide state-of-the-art results at approximately 97%.

In this study, we propose an efficient hybrid approach to solve this task. In our approach, word segmentation is represented as a binary classification problem in which we have to determine the label of each white space in input text. These two labels are SPACE (separator of two syllables which belong to two different words) and UNDERSCORE (separator of two syllables inside a word). Our system is mainly based on three steps. First, we use a forward longest matching algorithm to determine the boundary of all words having at least three syllables. Next, the classifier using logistic regression helps to detect the boundary of 2-syllable words and proper names. Finally, we continue to use the dictionary to recheck the predictions having low confidence produced by the machine learning process and return final labels for white spaces. For experiments, we evaluate our approach using 10-fold cross-validation on Vietnamese Treebank corpora [4] of 75k manually word-segmented sentences. Our system can yield an F-measure of 98.82% which is the best result for Vietnamese word segmentation known to us. Furthermore, the system can also perform at a high speed of nearly 34k tokens per second when running on a personal computer.

The rest of this paper is organized as follows. In Section II, we talk about the difficulties in Vietnamese word segmentation. In Section III, the methods used in other studies to resolve word segmentation task are discussed. Section IV provides details of our approach. We report and discuss about the experimental results of our system in Section V. Finally, we make some conclusions on this work in Section VI.

II. Difficulties in Vietnamese word segmentation

Vietnamese is an inflexionless language in which every word never changes its form. Vietnamese words are made of one or more syllables. A word which contains only one syllable is called single word. On the other hand, a word which is composed of more than one syllable is called compound word. Frequency of each kind of word is different from others. We tried to make some statistics on the dictionary provided by VLSP project[1] and the frequency analysis told us some useful knowledge. Almost of the words (71%) in this dictionary are 2-syllable words. Single words account for 17.67% of total words. Therefore, the percentage of over-2-syllable words is just under 12%. Due to this low frequency, it leads us to a simple idea that we can just use the dictionary to cover those words and then find an efficient way to deal with the other words in the input text. There are two kinds of the

[1] http://vlsp.hpda.vn:8080/demo/?page=home

978-1-5090-4134-3/16/$31.00 ©2016 IEEE

remaining words that we have to care about, which are 2-syllable words and proper names (in Vietnamese, names of people and locations are considered as lexical units).

One simple method to deal with proper names is to compose all consecutive upper-case syllables into a word. This method is obviously not good in many cases such as when two proper names appear consecutively.

For 2-syllable words, the easiest way is to scan through the input sentence and connect all of two consecutive syllables that can compose a word in the dictionary. There are many ambiguous cases where this method produce wrong results. One of the most frequent cases is called *overlap ambiguity* in which a sentence has three consecutive syllables $s_is_{i+1}s_{i+2}$ where both s_is_{i+1} and $s_{i+1}s_{i+2}$ are words in the dictionary but in the current context, only one of them is the right word. In another common situation, a word composed of two consecutive syllables s_is_{i+1} is in the dictionary, but in the current context, these two syllables are actually two single words. Other significant case that this method cannot handle is out-of-vocabulary problem in which two consecutive syllables s_is_{i+1} actually compose a right word in its context but it has not appeared in the dictionary.

Taken together, it is necessary to have more effective techniques to deal with those problems. In the next section, we talk about studied approaches to word segmentation Vietnamese and other languages' texts.

III. Related works

There are many effective approaches that have been studied to resolve word segmentation task [5], [6]. The first and traditional approach is based on dictionary. There are two common techniques of this approach, namely maximum matching (MM) and longest matching (LM). While MM algorithm aims to find the segmentation candidates by segmenting input sentence into a sequence with the smallest number of words, LM algorithm tends to scan through the sentence and at each syllable, it finds the longest word composed of this syllable and the next consecutive ones. Systems using this kind of approach for Chinese can gain very promising results [7], [8]. However, for Vietnamese, this simple approach seems to be unable to deal with out-of-vocabulary problem and overlap ambiguity.

The second one is statistical approach. As in many other core NLP tasks, this approach has proved to be good for word segmentation too. For instance, the methods using Conditional Random Fields (CRFs) and Support Vector Machines (SVMs) in [9] can reach results of over 94% while evaluating on a small corpus of 7800 Vietnamese sentences. Other studies using CRFs [10], SVMs [11], Hidden Markov Model (HMM) [12], [13], n-gram model [14], Maximum Entropy (MaxEnt) [15], [16] and probabilistic ensemble learning [17] also produces high accuracy for Vietnamese and other East Asian languages. Statistical approaches help to gain good result for Thai [18], too.

Although statistical algorithms can provide a good way to deal with ambiguous problems, both of those approaches still have their own limitations. Thus, some studies combined these two approaches into their systems. Some hybrid approaches for Vietnamese word segmentation were presented to use Weighted Finite State Transducer (WFST) with Neural Network [3], or combine MM and n-gram language model [1], or use MM combining with stochastic models using part-of-speech information [2]. These approaches are able to reach state-of-the-art results at approximately 97%. For Chinese, the study in [19] proposes a lattice-based framework for joint Chinese word segmentation, POS tagging and parsing which helps to significantly improve the accuracy of the three subtasks. Joint model of word segmentation and POS tagging was also used for Japanese [20].

IV. Our approach

In this section, we first talk about how we represent word segmentation task. Next, we describe three main components of our segmentation system before proposing its architecture.

A. Problem representation

The two main ways of problem representation for Vietnamese word segmentation are syllable-based and white-space-based.

The first one can be described as a sequential tagging task. For example, in the approach presented in [9], there are three labels for syllables, which are B_W (Begin of a Word), I_W (Inside of a Word) and O (Outside of a word). This approach is implemented in JVnSegmenter [21], a toolkit for Vietnamese word segmentation.

The second way is to cast Vietnamese word segmentation as a binary classification problem for white spaces. It should be repeated that in Vietnamese, there are two kinds of white space. The first one is separator of two syllables which belong to two different words (SPACE) and the second one is separator of two syllables inside a word (UNDERSCORE). PELSegmenter [17] and DongDu[2] are toolkits that use this problem representation.

We use the second way of problem representation for our system because of its simplicity. Moreover, in this way, it is possible to modify the label of a white space but does not affect the labels of other ones beside it. In the next section, we describe the simplest component of our system, longest matching algorithm.

B. Longest matching

Due to the low frequency of over-2-syllable words and in our observation, ambiguity is inappreciable for them, we just use such a dictionary to deal with those words. The dictionary-based technique in our system is longest matching. The dictionary is the one used in Section II. The work after longest matching is to handle the 2-syllable words and proper names efficiently. Our binary classifier using logistic regression is responsible for this task.

[2] https://github.com/rockkhuya/DongDu

C. Logistic regression as binary classification

Logistic regression is used to construct a binary classifier for white spaces in our system. From training data, we have a training set $D = \{(X, Y)\}$ where X denotes feature vector and Y denotes the corresponding label of white space. To be convenient, we denote the two values for Y as 1 and 0 corresponding to UNDERSCORE and SPACE labels respectively. Based on this training set, logistic regression assumes a parametric model and learns the conditional distribution $P(Y|X)$. The assumed parametric model is presented in equation 1 and equation 2, in which w_i denotes weight (or parameter).

$$P(Y=1|X) = \frac{1}{1+\exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)} \quad (1)$$

$$P(Y=0|X) = 1 - P(Y=1|X) \quad (2)$$

The rule of our binary classifier is that we assign UNDERSCORE label for a white space given its feature vector X if $P(Y=1|X) > P(Y=0|X)$ (or $P(Y=1|X) > 0.5$) and otherwise, we assign SPACE label for it if $P(Y=1|X) < 0.5$.

This statistical method seems to be able to handle proper names and many ambiguous problems well if we have a good feature set and large training data. However, it still has serious limitations as we map a continuous domain of probability P to a discrete domain of binary variable. But that is also the reason why we choose logistic regression instead of other methods. Obviously, no machine learning method can perform perfectly in all cases and it is necessary to have verification for its outcome. Logistic regression provides a simple way to detect those low-confident predictions. It is clear that predictions with probabilities P in a narrow boundary around 0.5 have low confidence of precision. Additionally, it is also possible that in the overlap ambiguity case, this classifier may connect all three syllables to compose a word. We will propose our simple techniques to resolve these problems in the following section.

D. Post-processing for binary classifier

We use the dictionary to handle the low-confident predictions and the results in overlap ambiguity cases produced by the binary classifier. First, we define that a prediction for label Y of a white space given its feature vector X is a low-confident prediction if the following condition holds:

$$|P(Y=1|X) - 0.5| < r, r \text{ is a threshold}$$

Assume that we have a sequence of syllables and labeled white spaces after the binary classification using logistic regression in the form of:

$$\dots s_{i-1}[\ ]s_i[*]s_{i+1}[\ ]s_{i+2}\dots$$

where s_j denotes syllable; [ ] denotes SPACE label; [_] denotes UNDERSCORE label and $[*]$ is the label that has low-confident precision. Our solution is to verify whether the word $s_i s_{i+1}$ is in the dictionary or not. The result of this operation is the final label for $[*]$.

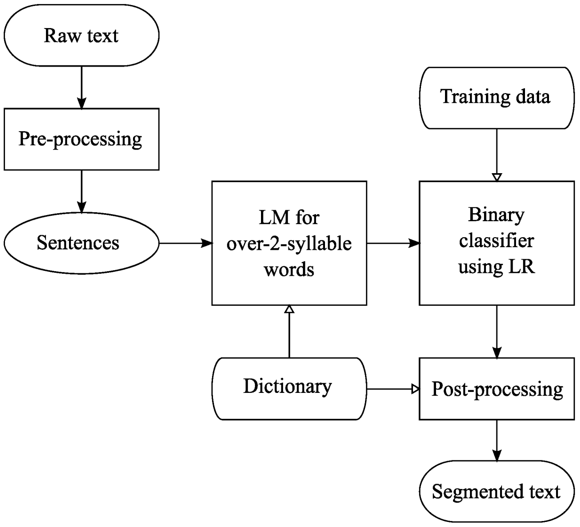

Figure 1. Architecture of our segmentation system.

In another case, the conducted sequence looks like that:

$$\dots s_{i-2}[\ ]s_{i-1}[\_]s_i[\_]s_{i+1}[\ ]s_{i+2}\dots$$

In this case, $s_{i-1}s_i s_{i+1}$ is not a word in the dictionary. That means it is much likely a wrong word because of the low frequency of 3-syllable words. We divided this case into four possibilities:

- word $s_{i-1}s_i$ is in dictionary but word $s_i s_{i+1}$ is not
- word $s_i s_{i+1}$ is in dictionary but word $s_{i-1}s_i$ is not
- both of them are not in dictionary
- both of them are in dictionary

For the first and second cases, we only keep up the word that appears in the dictionary. For the third case, we change both labels of two white spaces into SPACE. The last case is corresponding to overlap ambiguity. In this case, we keep up UNDERSCORE label of white space that has higher probability conducted by the classifier and change the other one to SPACE.

E. Proposed segmentation system

Combine all the above components with the pre-processing step for raw input data, we have the architecture of our system as presented in Figure 1.

In the pre-processing step, we first standardize the raw text, then use regular expressions to recognize regular patterns such as numbers, times and dates, then separate punctuation marks, parentheses and quotation marks at the end of words, and then utilize some simple heuristic rules to split the text into sentences. Next, each sentence is passed into the LM component to detect boundary for words having at least three syllables. Continuously, the remaining white spaces will be labeled by the classifier using LR. This classifier was trained on training data before. The post-processing then handles the low-confident predictions conducted by the classifier to return the final segmented text.

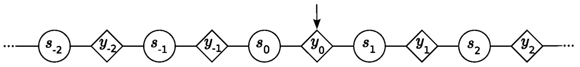

Figure 2. A 5-syllable window.

V. Experiments

In this section, we present the feature templates used for logistic regression and the performances of different systems compared to our system. We also talk about the affection of threshold r to accuracies of our segmentation system.

A. Features

Performance of any statistical technique is based on the quality of feature set. For the classifier using logistic regression of our system, to generate feature vector of each white space, we capture a window of size 2 for it as depicted in Figure 2, where s denotes syllable; y denotes white space and the subscript is the index of corresponding syllable or white space.

Table I represents all feature templates for logistic regression. In Table I, f_i denotes the lowercase-simplified form of syllable s_i; t_i is the type of syllable s_i; (f_i, f_j) is a combination feature; $isVNFamilyName(s_i)$ returns true if and only if s_i is a Vietnamese family name; $isVNSyllable(s_i)$ returns true if and only if s_i is a valid Vietnamese syllable. There are five types of syllable we defined in our system, namely LOWER, UPPER, ALLUPPER, NUMBER and OTHER corresponding to the cases that the syllable has all lowercase letters, the syllable has upper-case initial letter, the syllable has all upper-case letters, the syllable is a number or the other cases, respectively.

For n-gram features, we use both the lowercase form of syllables and their types. We do not use the original form of syllables from input text to extract n-gram features because we found that this way of feature extraction may produce profitless features and they are not good for logistic regression. Moreover, we only add features of syllable's types to feature vector if the type is different from LOWER. Taking all features of LOWER type to feature vectors can make the regression model confused and draw its performance because almost syllables are of LOWER type. These techniques can reduce a large number of useless features. The sixth feature template in Table I is used for full-reduplicative words. The seventh one catches information of Vietnamese people's name and the last one is used for detecting two consecutive proper names.

B. Results

We analyze affection of each component to our whole system. In our experiments, we use Vietnamese Treebank corpora of 75k manually word-segmented sentences which is one of the largest annotated corpora for Vietnamese. The corpus is randomly splitted into ten equal partitions for 10-fold cross-validation. F-measure is used, in which precision ratio (P) is computed as number of right segmented words over total number of words conducted by the segmentation system; recall ratio (R) is computed as number of right segmented words over total number of words in the golden test set. The average accuracies of systems over ten folds are presented in Table II. We utilize LIBLINEAR L2-regularized logistic regression [22] to implement the classifier for our experiments.

Table II
Accuracies of sub-systems (%).

Sub-system	P	R	F
LM	97.11	97.31	97.21
LR	97.95	98.29	98.12
LM + LR	98.11	98.16	98.14
LR + Post	98.59	**98.99**	98.79
LM + LR + Post	**98.77**	98.87	**98.82**

Our baseline system is LM which uses only longest matching algorithm and the rule to compose all consecutive UPPER syllables into a word. Longest matching algorithm is only used for phrases which have LOWER syllable(s) and do not consist of any NUMBER or OTHER syllable. This system can gain an F-measure of 97.21%, however, it obviously cannot resolve overlap ambiguity and out-of-vocabulary problems. Moreover, the rule for proper names is too much greedy and fails in many cases. Meanwhile, if we only utilize the classifier using logistic regression in system LR, the result is much better. The regression model can handle many cases of overlap ambiguity and out-of-vocabulary and provide a better way to detect proper names. Combining these two components to LM + LR system provides a slightly increased accuracy compared to LR system. The precision ratio is higher because LM + LR is able to cover all the over-2-syllable words that the LR system fails to catch. However, its recall ratio is decreased because of the inconsistency of those words in the training data and rarely ambiguous cases.

Post-processing for LR makes a significant impact on the result of segmentation. LR + Post system, which adds post-processing after LR system, can reach the highest recall ratio of 98.99%. Our whole system which is composed of all components (LM + LR + Post) makes the best result at 98.82% for F-measure. Obviously, performance of post-processing is mainly based on threshold r. In this experiment, we use $r = 0.33$ which helps to gain the best result. In the next section, we take a deeper look into how to choose a proper threshold r and discuss about its affection to the final result.

C. Discussion on threshold r and post-processing

Choosing a proper threshold r depends on the quality of dictionary and how well the machine learning process performs. It can be described that if we choose a high r, it means we rely on the dictionary more than the result of the classifier, and otherwise. Figure 3 depicts our analysis on the affection of threshold r to our system.

Due to the analysis, we can conclude that our system's results on Vietnamese Treebank corpora is not too sensitive with the variability of r in a wide range from 0.25 to 0.40. However, the high similarity between domains of the training data and test set is one reason for the high performance of our system. To adapt for other domains, it may face more

Table I
FEATURE TEMPLATES USED FOR LOGISTIC REGRESSION.

No.	Template
1	$(f_i), i = -2, -1, 0, 1, 2$
2	$(f_i, f_{i+1}), i = -2, -1, 0, 1$
3	$(t_i), i = -2, -1, 0, 1, 2$
4	$(t_i, t_{i+1}), i = -2, -1, 0, 1$ and $t_i \neq LOWER$
5	$(t_i, t_{i+1}, t_{i+2}), i = -2, -1, 0$ and $t_i \neq LOWER$
6	$(t_0 = t_1 = LOWER$ and $f_0 = f_1)?$
7	$(t_0 = t_1 = UPPER$ and $isVNFamilyName(s_0))?$
8	$(t_0 = t_1 = UPPER$ and $isVNSyllable(s_0)$ and $!isVNSyllable(s_1))?$

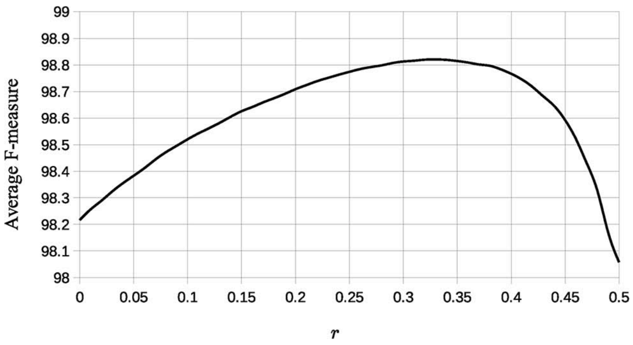

Figure 3. Affection of threshold r on word segmentation result.

problems with new words which even the dictionary cannot cover. In this situation, a validation set is needed in order to choose a proper threshold r for new domain.

D. Comparison to other toolkits

Our approach is compared to other approaches that have been presented in other studies. The accuracy figures are depicted in Table III.

Our system provides better result compared to other toolkits on Vietnamese Treebank corpus. It should be repeated that our classifier does not take information from the dictionary. We suspect that this is the reason why it performs better than other stochastic-based toolkits, DongDu and JVnSegmenter. vnTokenizer [1], which uses regexes to cover proper names before handling normal words, fails in many cases where upper-case syllables appear consecutively. It is obvious that statistical systems can perform better than vnTokenizer because the training data is not too different from the test set in term of content domain.

To make another comparison, we retrained each toolkit using the full corpus of Vietnamese Treebank and then evaluated them on an independent test set that consists of 10 files from *800001.seg* to *800010.seg* provided by VLSP project. From Table III, we can see that performances of statistical segmentation systems are decreased considerablely, because the new test set has a totally different domain, with many new words that have not appeared in neither the training data nor the dictionary of these systems. Our system with the main component using logistic regression is not an exception but it still has a good performance because of the simple feature set which does not make use of information from dictionary. vnTokenizer performs quite stablely and its result is slightly increased. Notably, vnTokenizer's dictionary has more than 40k words [1], this number of ours is 32k. Although having a poorer dictionary, our system is still able to outperform vnTokenizer.

Moreover, we also collected a corpus of 1k articles from Vietnamese online newspapers to measure segmentation speed of toolkits. Except DongDu which is developed in C++, the other toolkits are developed in Java. The evaluation is processed on a personal computer with 4 Intel Core i5-3337U CPUs @ 1.80GHz and 6GB of memory. The results is reported in Table IV. Our system can run faster than other toolkits. DongDu toolkit also utilizes LIBLINEAR for machine learning, however, its feature set is much more complicated and its LIBLINEAR version is older than ours. We suspect these are the reasons why DongDu's speed is not as high as our system's. vnTokenizer and JVnSegmenter were written in old versions of Java. Their code to process on *String* seems to be inefficient so that their speeds are quite low.

E. UETsegmenter

Our toolkit used for the above experiments is written in Java and called UETsegmenter. It provides APIs for Vietnamese word segmentation using a pretrained model and also some methods for training and testing new models. The toolkit and related resources are freely available for download[3].

VI. CONCLUSIONS

In this paper, we propose a hybrid approach to Vietnamese word segmentation using longest matching and logistic regression. We cast this task as a binary classification problem for white spaces and the results show that longest matching algorithm, logistic regression combining with our simple post-processing techniques helps to gain high accuracy. Our system can reach state-of-the-art result at 98.82% for F-measure while evaluating on Vietnamese Treebank corpus. Moreover, the system can perform at a high speed of 34k tokens per second. For future works, we will make a deeper study on the affection of dictionary and the classifier on choosing proper threshold and extend post-processing to deal with other cases. We will also find an efficient way to enrich the dictionary to produce a better segmentation system.

[3] https://github.com/phongnt570/UETsegmenter

Table III
ACCURACY COMPARISON (%).

Toolkit	10-fold CV			Independent test set		
	P	R	F	P	R	F
vnTokenizer	97.61	96.86	97.23	96.98	97.69	97.33
JVnSegmenter - Maxent	97.18	97.28	97.23	96.60	97.40	97.00
JVnSegmenter - CRFs	97.58	97.68	97.63	96.63	97.49	97.06
DongDu	97.44	98.01	97.72	96.35	97.46	96.90
Ours	98.77	98.87	98.82	97.51	98.23	97.87

Table IV
SPEED COMPARISON.

Toolkit	JVnSeg (CRFs)	JVnSeg (MaxEnt)	vnTokenizer	DongDu	Ours
Speed (tokens/s)	764	1082	5322	16709	33705

ACKNOWLEDGMENT

This paper is supported by The Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2014.22.

REFERENCES

[1] H. P. Le, T. M. H. Nguyen, A. Roussanaly, and T. V. Ho, "A hybrid approach to word segmentation of vietnamese texts," in *2nd International Conference on Language and Automata Theory and Applications-LATA 2008*, vol. 5196. Springer Berlin/Heidelberg, 2008, pp. 240–249.

[2] D. D. Pham, G. B. Tran, and S. B. Pham, "A hybrid approach to vietnamese word segmentation using part-of-speech tags," in *International Conference on Knowledge and Systems Engineering 2009*. IEEE, 2009, pp. 154–161.

[3] D. Dinh, K. Hoang, and V. T. Nguyen, "Vietnamese word segmentation," in *NLPRS*, vol. 1, 2001, pp. 749–756.

[4] P.-T. Nguyen, X.-L. Vu, T.-M.-H. Nguyen, V.-H. Nguyen, and H.-P. Le, "Building a large syntactically-annotated corpus of vietnamese," in *Proceedings of the Third Linguistic Annotation Workshop*, ser. ACL-IJCNLP '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 182–185. [Online]. Available: http://dl.acm.org/citation.cfm?id=1698381.1698416

[5] Q. T. Dinh, H. P. Le, T. M. H. Nguyen, C. T. Nguyen, M. Rossignol, and X. L. Vu, "Word segmentation of Vietnamese texts: a comparison of approaches," in *6th international conference on Language Resources and Evaluation - LREC 2008*. Marrakech, Morocco: ELRA - European Language Resources Association, May 2008. [Online]. Available: https://hal.inria.fr/inria-00334760

[6] C. Huang and H. Zhao, "Chinese word segmentation: A decade review," *Journal of Chinese Information Processing*, vol. 21, no. 3, pp. 8–20, 2007.

[7] K.-J. Chen and S.-H. Liu, "Word identification for mandarin chinese sentences," in *Proceedings of the 14th Conference on Computational Linguistics - Volume 1*, ser. COLING '92. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 101–107. [Online]. Available: http://dx.doi.org/10.3115/992066.992085

[8] P.-k. Wong and C. Chan, "Chinese word segmentation based on maximum matching and word binding force," in *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, ser. COLING '96. Stroudsburg, PA, USA: Association for Computational Linguistics, 1996, pp. 200–203. [Online]. Available: http://dx.doi.org/10.3115/992628.992665

[9] C.-T. Nguyen, T.-K. Nguyen, X.-H. Phan, L.-M. Nguyen, and Q.-T. Ha, "Vietnamese word segmentation with crfs and svms: An investigation," in *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation (PACLIC 2006)*, 2006.

[10] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to japanese morphological analysis," in *EMNLP*, vol. 4, 2004, pp. 230–237.

[11] M. Sassano, "An empirical study of active learning with support vector machines for japanese word segmentation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 505–512. [Online]. Available: http://dx.doi.org/10.3115/1073083.1073168

[12] T. Nguyen, V. Nguyen, and A. Le, "Vietnamese word segmentation using hidden markov model," in *International Workshop for Computer, Information, and Communication Technologies in Korea and Vietnam*, 2003.

[13] C. P. Papageorgiou, "Japanese word segmentation by hidden markov model," in *Proceedings of the Workshop on Human Language Technology*, ser. HLT '94. Stroudsburg, PA, USA: Association for Computational Linguistics, 1994, pp. 283–288. [Online]. Available: http://dx.doi.org/10.3115/1075812.1075875

[14] L. A. Ha, "A method for word segmentation in vietnamese," in *Proceedings of Corpus Linguistics*, 2003.

[15] O. T. Tran, C. A. Le, and T. Q. Ha, "Improving vietnamese word segmentation and pos tagging using mem with various kinds of resources," *Information and Media Technologies*, vol. 5, no. 2, pp. 890–909, 2010.

[16] D. Dinh and T. Vu, "A maximum entropy approach for vietnamese word segmentation," in *International Conference on Research, Innovation and Vision for the Future, 2006*. IEEE, 2006, pp. 248–253.

[17] W. Liu and L. Lin, "Probabilistic ensemble learning for vietnamese word segmentation," in *Proceedings of the 37th International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, ser. SIGIR '14. New York, NY, USA: ACM, 2014, pp. 931–934. [Online]. Available: http://doi.acm.org/10.1145/2600428.2609477

[18] C. Haruechaiyasak, S. Kongyoung, and M. Dailey, "A comparative study on thai word segmentation approaches," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on*, vol. 1, May 2008, pp. 125–128.

[19] Z. Wang, C. Zong, and N. Xue, "A lattice-based framework for joint chinese word segmentation, pos tagging and parsing," 2013.

[20] N. Kaji and M. Kitsuregawa, "Accurate word segmentation and pos tagging for japanese microblogs: Corpus annotation and joint modeling with lexical normalization," in *EMNLP*, 2014, pp. 99–109.

[21] C.-T. Nguyen and X.-H. Phan, "Jvnsegmenter: A java-based vietnamese word segmentation tool," *Retrieved on*, vol. 30, 2011.

[22] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1390681.1442794