A Hybrid Approach to Vietnamese Word Segmentation using Part of Speech tags

Dang Duc Pham, Giang Binh Tran, Son Bao Pham
Faculty of Information Technology
College of Technology
Vietnam National University, Hanoi

Abstract

Word segmentation is one of the most important tasks in NLP. This task, within Vietnamese language and its own features, faces some challenges, especially in words boundary determination. To tackle the task of Vietnamese word segmentation, in this paper, we propose the WS4VN system that uses a new approach based on Maximum matching algorithm combining with stochastic models using part-of-speech information. The approach can resolve word ambiguity and choose the best segmentation for each input sentence. Our system gives a promising result with an F-measure of 97%, higher than the results of existing publicly available Vietnamese word segmentation systems.

1. Introduction

Word segmentation is one of the most basic and important tasks in Natural Language Processing (NLP). In some studies, it is the foremost task that must be completed before further analysis [13]. For example, most of text classification systems use word segmentation approaches combined with machine learning algorithms [9]. Word segmentation also plays a significant role in parsing and machine translation.

Word segmentation faces different challenges depending on the type of language. This task for Eastern Asian languages, such as Chiness, Japaness, Thai and Vietnamese, is very difficult since they are isolating languages in which spaces are not always the boundaries among words and one word can consist of more than one token [1, 2, 7, 10]. Thus, to determine words boundaries we need higher-level analysis like word form, morphology, syntax or semantics analysis. Part-of-speech (POS) is also a useful piece of information for word segmentation. In the view of linguistics, the difference in word segmentation gives a difference result in POS tagging, and a POS tagging sequence brings about a correlative word segmentation result. Therefore, POS can be helpful for word segmentation and resolving word ambiguity [3].

By analyzing existing approaches to Vietnamese word segmentation and considering the relationship between POS tagging and word segmentation, we propose a hybrid approach that uses Maximum matching algorithm combining with statistical models and part-of-speech information to address the word ambiguity problem.

In section 2, we will give an overview the Vietnamese word segmentation task and its challenges. We cover related works in section 3 and then propose our approach in section 4. Section 5 describes our experimental setup and results. Section 6 concludes with pointers to our work in the future.

2. Vietnamese word segmentation: difficulties and challenges

In Vietnamese, syllable is the smallest linguistic unit, and one word consists of one or more syllables [5, 7, 13]. It leads to an ambiguity problem when determining word boundaries. There are two type of ambiguity namely cross ambiguity and overlap ambiguity. In cross ambiguity, some syllables themselves have meaning (can be a word), and their combination also has meaning. For example, in the sentence "*"Bàn là một công cụ học tập",* "Bàn" means "Desk", "là" means "is", and "Bàn là" means "iron". It is very difficult to tackle this problem. However, in Vietnamese, it occurs less frequently than the overlap ambiguity [10]. Overlap ambiguity occurs in a situation that a syllable when combined with previous syllable or following syllable in a sentence generates words. For example, in the sentence "*Tốc độ truyền thông tin ngày càng cao", "truyền thông"* and "*thông tin"* both are words. Since overlap ambiguity occurs more often, solving this kind of ambiguity can improve the Vietnamese word segmentation system.

Besides word boundary ambiguity, Vietnamese word segmentation faces with a problem in which there are lots of new words appearing in a document. These new words are normally names that refer to people, location, abbreviation of foreign words, currency units, etc [5, 10].

3. Related works

There are many approaches tackling the word segmentation task. They are divided into 3 main categories: dictionary-based, statistics-based and hybrid-based [7, 16]. Figure 1 shows a diagram of these categories.

Traditionally, most studies use dictionary-based approaches because of their simplicity. This type of approach is the base for further studies. Two most effective techniques are Maximum matching (MM) and Longest matching (LM). Some word segmentation systems like MMSEG for Chinese language also use these techniques [12]. In this system, the authors use Maximum matching algorithm together with rules built from observation. This approach is simple yet gives a promising result. However, most of the dictionary-based approaches often do not give high accuracy because they fail to solve many ambiguous cases.

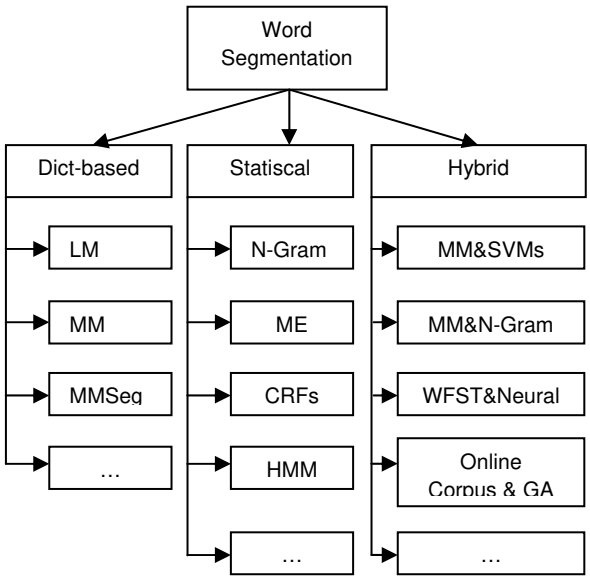

Figure 1. Categories of word segmentation approaches

Statistical approaches utilize information from a very large annotated corpus. Examples include N-gram Language Model [2], Hidden Markov Model (HMM) [4], Conditional Random Fields (CRFs) [8] and Maximum Entropy (ME) [1]. This type of approach proves to be useful when deploying in different languages.

Hybrid approaches combine different approaches to make use of individual advantages and overcome disadvantages. Many hybrid models are published and applied in many different languages. They consist of dictionary-based techniques (Maximum matching, Longest matching), statistics-based techniques (N-gram, CRFs, ME) and machine learning algorithms (Support Vector Machines - SVMs, Genetic Algorithm - GA) [11, 14, 18].

These approaches are well applied for the task of Vietnamese word segmentation. Some studies using techniques like N-gram model with 10 million syllables corpus [13], Hidden Markov Model [15], Maximum Entropy [6] show high accuracy. Some models are combination of Maximum matching and SVMs [19], statistics of the Internet and Genetic algorithm [17], WFST and Neural network [7], Maximum matching and N-gram language model [10].

4. Our approach

In Vietnamese, a sentence itself has diversified structures where words may have many POS tags when taken out of context. However, when a sentence is put into a specific situation, the POS tagging sequence is unique. Therefore, there is a relationship between POS tagging and word segmentation in Vietnamese. In a particular situation, a wrong POS tagging can bring about wrong word segmentation correspondingly. On the contrary, a right POS tagging can lead to right correlative word segmentation. Based on this observation, we propose a Vietnamese word segmentation technique using POS tags. Our approach does not aim to tackle the task of POS tagging and word segmentation in parallel but aims to take advantage of the POS tag information to address the word ambiguity problem.

4.1. Overview of our approach

As mentioned, our approach is a hybrid technique combining Maximum matching algorithm, POS tags and statistical models to tackle the overlap ambiguity. Given a sentence S as input, the general formula for our approach is:

$$\begin{aligned}(\hat{W},\hat{T}) &= \underset{W,T}{argmax}\, P(W,T|S) \\ &= \underset{W,T}{argmax}\, \frac{P(W,T,S)}{P(S)} \\ &= \underset{W,T}{argmax}\, P(W,T,S) \\ &= \underset{W,T}{argmax}\, P(W,T)\end{aligned}$$

$$= \underset{W,T}{argmax}\, P(W|T) * P(T)$$

Where W = w_1, w_2, …, w_n is a word segmentation solution for S, and T = t_1, t_2, …, t_n is the corresponding POS sequence.

Our system consists of 5 main steps: preprocessing to standardize the input data, segmentation to give potential segmentation candidates, POS tagging based on Markov-1, Markov-2 and "front-back" hypothesis for each candidate, calculating the probability of corresponding segmentation candidates, and finally, choosing the best segmentation candidate. Figure 2 shows the architecture of our system called WS4VN.

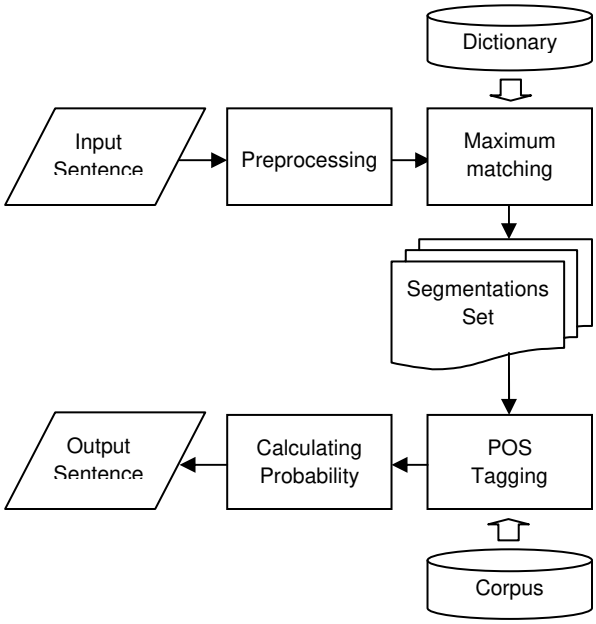

Figure 2. Architecture of our system WS4VN

4.2. Preprocessing

There is ambiguity of spelling rules in Vietnamese language. The most typical ones include using "i" or "y" and where to put marks on syllables. Thus, the preprocessing step aims to normalize data for further analysis using one standard. Moreover, this step aims identify new words in the form of named entities such as names of companies, people, factoids, etc. We use regular expression as the main technique to recognize named entities.

4.3. Maximum matching

We use Maximum matching algorithm to find the segmentation candidates by segmenting the input sentence into a sequence with the smallest number of words. With the overlap ambiguity, there is often more than one solution.

For example: "*Tốc độ truyền thông tin ngày càng cao"* can be segmented as:

- *"Tốc độ | truyền | thông tin | ngày càng | cao"*
- *"Tốc độ | truyền thông | tin | ngày càng | cao"*

The output of the Maximum matching algorithm is a set of segmented sequences with the smallest number of words.

4.4. POS tagging

We tag each segmented sequences received from the previous step. We build a POS tagger by using Hidden Markov approach with 3 hypotheses: Markov-1, Markov-2 and "front-back". We also use a published POS tagger VnQTAG [20] with the accuracy of 94%.

4.4.1. POS tagger using Hidden Markov model. Hidden Markov Model (HMM) is a statistical model that is used to determine hidden parameters based on observed parameters. It is widely used, especially in POS tagging for an input sequence. We will describe the POS tagging as a Hidden Markov model (Figure 3).

- Hidden part: tag sequence T
- Observed part: word sequence from previous step W
- Transition probability: $a_{i-1,i} = P(t_i|t_{i-1})$ with hypothesis Markov-1, or $a_{i-1,i} = P(t_i|t_{i-1}, t_{i-2})$ with hypothesis Markov-2
- Output probability: $b_i = P(w_i|t_i)$

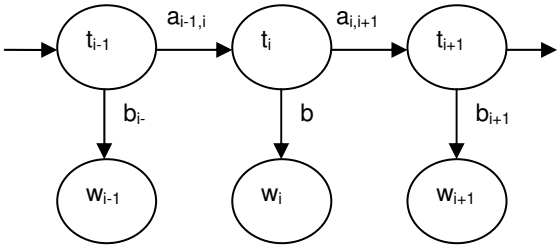

Figure 3. HMM for POS tagging task

A tagged sequence $\hat{T} = t_1, t_2, \dots, t_n$ satisfies:

$$\hat{T} = \underset{T}{argmax}\, P(T|W)$$
$$= \underset{T}{argmax} \frac{P(W|T) * P(T)}{P(W)}$$
$$= \underset{T}{argmax}\, P(W|T) * P(T)$$
$$= \underset{T}{argmax}\, P(w_1, w_2, \dots, w_n|t_1, t_2, \dots, t_n) * P(t_1, t_2, \dots, t_n)$$

Suppose that if we know the POS of a word, it is possible to determine this word. Thus, the probability P(W|T) only depends on basic probability like $P(w_i|t_i)$:

$$P(W|T) \approx \prod_{i=1}^{n} P(w_i|t_i)$$

Further calculation for P(T):

$$P(T) = P(t_1, t_2, \dots, t_n)$$
$$= P(t_1) * P(t_2|t_1) * \dots * P(t_n|t_1, t_2, \dots, t_{n-1})$$

Apply the Markov-1 hypothesis in which probability of a POS appearing in a sequence can be predicted when we know the probability of the previous POS. That means $P(t_i|t_1, t_2, \dots, t_{i-1}) \approx P(t_i|t_{i-1})$

Then:

$$P(T) = P(t_1, t_2, \dots, t_n)$$
$$= P(t_1) * P(t_2|t_1) * \dots * P(t_n|t_1, t_2, \dots, t_{n-1})$$
$$\approx P(t_1|t_0) * P(t_2|t_1) * \dots * P(t_n|t_{n-1})$$
$$= \prod_{i=1}^{n} P(t_i|t_{i-1})$$

(t_0 is a special tag representing for sentence's beginning)

Finally we get:

$$\hat{T} = \underset{T}{argmax} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

Where probabilities $P(w_i|t_i)$ and $P(t_i|t_{i-1})$ can be estimated by Maximum likelihood technique based on an annotated corpus.

In a similar manner, when apply the Markov-2 hypothesis we get:

$$\hat{T} = \underset{T}{argmax} \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-2}, t_{i-1})$$

Therefore, we then can use dynamic programming algorithm Viterbi to solve the POS tagging task.

4.4.2. POS tagger using "front-back" model. With the tagger using Hidden Markov model, the probability of a POS tag appearing depends only on the previous tags. That means there is no relationship between previous tags and the next tags. However, in reality, these relationships still exist. Thus, we propose a hypothesis called "front-back" in order to make use of this relationship. According to the "front-back" hypothesis, the probability of a POS tag t_i depends on the previous tag t_{i-1} and the next tag t_{i+1}, then we can calculate $\boldsymbol{P(T) = P(t_1, t_2, \dots, t_n)}$ as follows:

$$P^2(T) = P(t_1, t_2, \dots, t_n) * P(t_1, t_2, \dots, t_n)$$
$$= P(t_{n-1}|t_1, t_2, \dots, t_{n-2}, t_n) * P(t_1, t_2, \dots, t_{n-2}, t_n) * P(t_n|t_1, t_2, \dots, t_{n-1}) * P(t_1, t_2, \dots, t_{n-1})$$

Then:

$$P(t_{n-1}|t_1, t_2, \dots, t_{n-2}, t_n) \approx P(t_{n-1}|t_{n-2}, t_n)$$
$$P(t_1, t_2, \dots, t_{n-2}, t_n) \approx P(t_1, t_2, \dots, t_{n-2}) * P(t_n)$$
$$P(t_n|t_1, t_2, \dots, t_{n-1}) \approx P(t_n|t_{n-1}, t_{n+1})$$

Then:

$$P^2(T) = P(t_1, t_2, \dots, t_n) * P(t_1, t_2, \dots, t_n)$$
$$= P(t_{n-1}|t_1, t_2, \dots, t_{n-2}, t_n) * P(t_1, t_2, \dots, t_{n-2}, t_n) * P(t_n|t_1, t_2, \dots, t_{n-1}) * P(t_1, t_2, \dots, t_{n-1})$$
$$\approx P(t_{n-1}|t_{n-2}, t_n) * P(t_1, t_2, \dots, t_{n-2}) * P(t_n) * P(t_n|t_{n-1}) * P(t_1, t_2, \dots, t_{n-1})$$
$$\approx P(t_{n-1}|t_{n-2}, t_n) * P(t_{n-3}|t_{n-4}, t_{n-2}) * P(t_1, t_2, \dots, t_{n-4}) * P(t_n) * P(t_n|t_{n-1}, t_{n+1}) * P(t_{n-2}|t_{n-3}, t_{n-1}) * P(t_1, t_2, \dots, t_{n-3}) * P(t_{n-1})$$
$$\approx \dots$$
$$\approx \prod_{i=1}^{n} P(t_i|t_{i-1}, t_{i+1}) * P(t_i)$$

Where t_0 and t_{n+1} represents for sentence's first and last element respectively. Finally, we get:

$$\hat{T} = \underset{T}{argmax}\, P(T|W)$$
$$= \underset{T}{argmax} \frac{P(W|T) * P(T)}{P(W)}$$
$$= \underset{T}{argmax}\, P(W|T) * P(T)$$
$$= \underset{T}{argmax}\, P^2(W|T) * P^2(T)$$
$$= \underset{T}{argmax} \prod_{i=1}^{n} P^2(w_i|\, t_i) * P(t_i|t_{i-1}, t_{i+1}) * P(t_i)$$
$$= \underset{T}{argmax} \prod_{i=1}^{n} P(w_i|\, t_i) * P(w_i, t_i) * P(t_i|t_{i-1}, t_{i+1}) \qquad (3)$$

Thanks to the general calculation (3), we propose a dynamic programming algorithm to tackle the POS tagging task. The algorithm is described as below:

- Let $Q[i, t_{i+1}, t_i]$ is the maximum probability according to (3) when calculate to the i^{th} word and its POS is t_i and POS tag of the $i+1^{th}$ word is t_{i+1}
- Initialize $Q[0, t, t_0] = 1\ \forall t_i \in L_1$
- At i^{th}:

$$Q[i, t_{i+1}, t_i]$$
$$= \underset{t_{i-1}}{max}\{Q[i-1, t_i, t_{i-1}] * P(w_i|t_i) * P(w_i, t_i) * P(t_i|t_{i-1}, t_{i+1}), \forall t_{i-1} \in L_{i-1}\}$$

We use a Trace array to store progress of calculation then we can explore back and forth to determine the result POS tag sequence.

$$Trace[i, t_{i+1}, t_i]$$
$$= \underset{t_{i-1}}{argmax}\{Q[i-1, t_i, t_{i-1}] * P(w_i|t_i) * P(w_i, t_i) * P(t_i|t_{i-1}, t_{i+1}), with\ t_{i-1} \in L_{i-1}\}$$

4.4.3. ***Outer POS tagger VnQTAG.*** VnQTAG is a Vietnamese POS tagger published by H. Nguyen It is published with an accuracy of 94%. So we use VnQTAG as an optional component in our system [20].

4.5. Calculating true probabilities and choosing the best solution

After the 3th step of deciding the POS tags for each word segmentation candidate, we calculate the probability for each segmented sequence P(T|W). We consider it as the true probability representing for the possibility of T as the true tag sequence of the input sentence W:

According to the Markov-1:

$$P(T|W) \approx \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-1})$$

According to the Markov-2:

$$P(T|W) \approx \prod_{i=1}^{n} P(w_i|t_i) * P(t_i|t_{i-2}, t_{i-1})$$

According to the "front-back":

$$P(T|W) \approx \prod_{i=1}^{n} P(w_i|\, t_i) * P(w_i, t_i) * P(t_i|t_{i-1}, t_{i+1})$$

The best candidate is the one with the highest probability.

5. Experiment

5.1. Evaluation method

We use F-measure to evaluate performance of our word segmentation approach:

- *Precision:* $P = \frac{N_3}{N_1} * 100$
- *Recal:* $R = \frac{N_3}{N_2} * 100$
- *F-Measure:* $F = \frac{2*P*R}{P+R}$

Where

- N_1 is the number of words recognized by the system
- N_2 is the number of words in reality appearing in the corpus
- N_3 is the number of words that are correctly recognized by the system

5.2. Data preparation

Our corpus consists of:

POS dictionary: we use 2 sets of POS tags: POS level 1 and POS level 2 with all tags defined in the Corpus of VnQTAG [20]. POS level 2 contain the full set of 47 types of POS. POS level 1 contains 8 different tags where some tags in level 2 having similar semantics are grouped together.

Vocabulary dictionary: 37454 words and their possible tags from corpus of VnQTAG [20]. If a word is not defined, it will be tagged as X label.

N-gram training data: documents of various topics like literature, science, news journal, etc. They are manually segmented and POS tagged. The data contains 74756 words with 47 POS level 2 and some labels for special characters like sentence's marks. We also do a statistical calculation on the training data with 8 POS level 1 of VnQTAG [20].

Testing data: the testing data consists of 19417 sentences with approximate 490116 words, provided by H.P. Le [10]. The data has been manually word segmented.

5.3. Experimental parameters setup

Table 1 describes a list of parameters for the experiments. They are used to measure performance of the system in each case. The results will be discussed and analyzed. The experiments can be grouped as follows:

Table 1: Experimental parameters list

No	*Parameters*	*Description for correlative POS level and hypothesis using*
0	*P1&M1*	*POS level 1 and Markov 1 hypothesis*
1	*P1&M2*	*POS level 1 and Markov 2 hypothesis*
2	*P2&M1*	*POS level 2 and Markov 1 hypothesis*
3	*P2&M2*	*POS level 2 and Markov 2 hypothesis*
4	*P1&M0*	*POS level 1 and "front-back" hypothesis*
5	*P2&M0*	*POS level 2 and "front-back" hypothesis*
6	*OMM*	*Only Maximum matching, no POS*
7	*AS11*	*Without Maximum matching, choose the best solution from all of segmentation results.*
8	*EPT11*	*Using VnQTAG tagger instead of the tagger that we build*

Parameters for POS level and hypothesis (No.0-5): This set of parameters is to consider which level of POS tags and which hypothesis Markov-1, Markov-2 or "front-back" will be used.

Parameters for Maximum matching step (No.6-7): This parameter is to evaluate effectiveness of

Maximum matching algorithm when applying for the word segmentation task. We calculate accuracy of our system in both following cases and compare them with each other:

- Using Maximum matching for word segmentation
- Not using Maximum matching for word segmentation

Parameters for choosing POS taggers (No.8): They are to evaluate performance of system in following cases:

- Using our POS tagger with different hypothesis that we build
- Using outer POS tagger VnQTAG

Compare performance with other systems: We compare the performance of our system with JVnSegmenter using CRF approach [5] and VnTokenizer with Maximum matching and N-gram [10].

5.4. Result and discussion

Table 2 shows the results of our system with parameters for POS level and hypothesis.

There is no great difference among results with respect to parameters' values. Thus, all of POS levels and hypotheses appear to have similar influence in the system performance. However, with the promising result and simplicity, we propose using the POS level 1 and Markov-1 model. Moreover, it proves that the "front-back" model can bring about good performance and can be applied for many further applications.

Table 2: Results of system with parameters for POS level and hypothesis

Score / *Parameter*	***Precision***	***Recall***	***F-Measure***
P2&M0	*97.156*	*96.711*	*96.933*
P1&M0	***97.161***	***96.716***	***96.938***
P2&M2	*97.147*	*96.703*	*96.924*
P2&M1	*97.169*	*96.724*	*96.946*
P1&M2	*97.177*	*96.732*	*96.954*
P1&M1	***97.185***	***96.740***	***96.962***

Table 3 shows the results of the system when considered choosing Maximum matching algorithm or not. The system run with Maximum matching algorithm and no information of POS gives the lowest result while the system run with Maximum matching algorithm and use information of POS gives highest result. It proved that within our experiment, using Maximum matching to choose set of segmentations that have the smallest number of words together with POS information is better than using all of word segmentation sequences. Moreover, System with Maximum matching algorithm has better complexity.

Table 3: Results of the system when running with and without the Maximum matching algorithm.

Score / *Parameter*	***Precision***	***Recall***	***F-Measure***
OMM	*91.242*	*94.852*	*93.012*
AS11	*95.338*	*95.784*	*95.560*
P1&M1	***97.185***	***96.740***	***96.962***

Table 4 shows the performance of our system when choosing different POS taggers. There is no great difference between using VnQTAG and using our POS tagger. This proves that the accuracy of the system does not depend on which tagger is used. However, our POS tagger results in better running time for the whole system as we don't have to call outer POS tagger to get the POS sequence.

Table 4. Performance of the system when choosing different POS taggers

Score / *Parameter*	***Precision***	***Recall***	***F-Measure***
EPT11	*96.978*	*96.533*	*96.755*
P1&M1	***97.185***	***96.740***	***96.962***

Figure 4 shows the chart of the results with different experimental parameters.

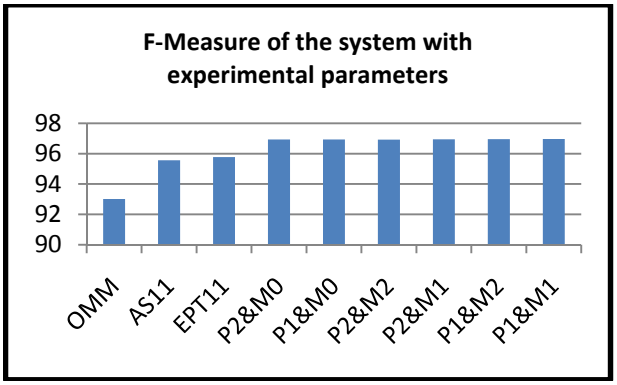

Figure 4. Performance of system with different parameters

The Figure 5 and Table 5 show the accuracy of our system against two existing Vietnamese word segmentation systems. These two systems have been retrained and tested on the same training and test corpus respectively as ours. It can be seen that our system gives the best result.

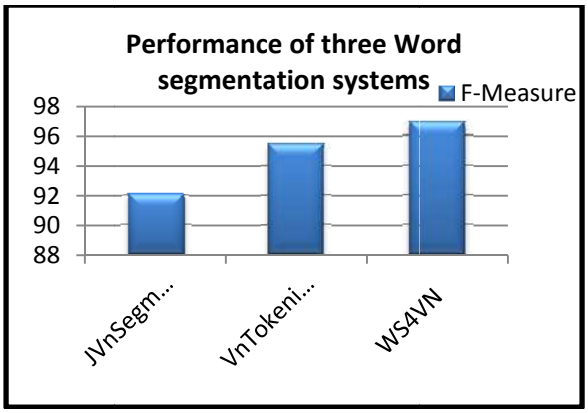

Figure 5. Performance of the system when comparing with other word segmentation systems

Table 5: Accuracy of word segmentation systems

Score / System	***Precision***	***Recall***	***F-Measure***
JvnSegmenter	*94.768*	*92.473*	***93.606***
VnTokenizer	*96.098*	*94.869*	***95.480***
WS4VN	*97.185*	*96.470*	***96.962***

The training data with 74756 words is probably not big enough for JvnSegmenter and VnTokenizer to perform at their best. But this highlights an advantage of our approach that by using POS information, we can afford to achieve good result with a smaller corpus.

6. *Conclusion and future works*

In this paper, we propose a new approach for Vietnamese word segmentation task. It is a hybrid approach combining Maximum matching algorithm and statistical models on POS tags. The experiment shows a promising result with an F-measure of approximately 97%. It aims to address the overlap ambiguity.

One of the limitations is that we have not a big corpus for better training. In the future, we would like to extend our corpus and try some other approaches to improve performance of our system.

Reference

[1] A.J. Jacobs, & W.Y. Wong. 2006. Maximum Entropy Word Segmentation of Chinese Text. *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing, p.108-117.*

[2] B. Capenter. 2006. Character Language Models for Chinese Word Segmentation and Named Entity Recognition. *ACL, p.169-172.*

[3] C.H. Chang, & C.D. Chen. 1993. A Study on Integrating Chinese Word Segmentation and Part-Of-Speech Tagging. *Communications of COLIPS, Vol.3, No.2, p.69-77.*

[4] C.P. Papageorgiou. 1994. Japanese Word Segmentation by Hidden Markov Model. *Proceedings of the HLT Workshop, p. 283 – 288*

[5] C.T. Nguyen, T.K. Nguyen, X.H. Phan, L.M. Nguyen, & Q.T. Ha. 2006. Vietnamese Word Segmentation with CRFs and SVMs: An Investigation. *Proceedings of the 20th PACLIC, Wuhan, China, p.215-222.*

[6] D. Dinh, & T. Vu. 2006. A Maximum Entropy Approach for Vietnamese Word Segmentation. *Proceedings of 4th RIVF, VietNam, p.12-16.*

[7] D. Dinh, K. Hoang, & V.T. Nguyen. 2001. Vietnamese Word Segmentation. *The 6th NLPRS, Tokyo, Japan, p. 749-756.*

[8] F. Peng, F. Feng, & A. McCallum. 2004. Chinese Segmentation and New Word Detection using Conditional Randome Field. *Proceedings of COLING, p.562-568.*

[9] F. Peng, X. Huang, D. Schuurmans, & S. Wang. 2003. Text classification in Asian Language without Word Segmentation. *Proceedings of 6th IRAL, p.41- 48.*

[10] H.P. Le, T.M. H. Nguyen, A. Roussanaly, & T.V. Ho. 2008. A Hybrid Approach to Word Segmentation of Vietnamese Text. *Proceedings of 2nd LATA.*

[11] J.S. Zhou, X.Y. Dai, R.Y. Ni, & J.J. Chen. A Hybrid Approach to Chinese Word Segmentation around CRFs. *Proceedings of 4th SIGHAN Workshop on Chinese Language Processing.*

[12] K.J. Chen, & S.H. Liu. 1992. Word identification for Mandarin Chinese sentences. *Proceedings of the 15th COLING.*

[13] L.A. Ha. 2003. A method for Word segmentation in Vietnamese. *Proceedings of the Corpus Linguistics 2003, Lancaster, UK.*

[14] N. Xue, S.P. Converse. 2002. Combining Classifiers for Chinese Word Segmentation. *First SIGHAN Workshop attached with the 19th COLING, p.57-63.*

[15] P.T. Nguyen, V.V. Nguyen, & A.C. Le. 2003. Vietnamese Word Segmentation Using Hidden Markov Model. *International Workshop for Computer, Information, and Communication Technologies on State of the Art and Future Trends* of *Information Technologies in Korea and Vietnam.*

[16] S. Foo, H. Li. 2004. Chinese Word Segmentation and Its Effect on Information Retrieval. *Information Processing & Management: An International Journal. Vol.40, No.1, p.161-190.*

[17] T.H. Nguyen. 2006. Word Segmentation for Vietnamese Text Categorization: An online corpus approach. *Proceedings of 4th RIVF, Ho Chi Minh, VietNam.*

[18] X. Lu. 2005. Towards a Hybrid Model for Chinese Word Segmentation. *Proceedings of 4th SIGHAN Workshop on Chinese Language Processing.*

[19] D. Vu, N.L. Nguyen, D. Dinh. 2006. Application of Maximum matching and SVMs for Vietnamese word segmentation *ICT.rda'06, Đà Lạt.*

[20] T.M.H. Nguyen, X.L. Vu, & H.P. Le. 2003. Using QTAG POS tagging for Vietnamese documents. *ICT.rda'03, Ha Noi.*