Vietnamese Word Segmentation

Dinh Dien, Hoang Kiem, Nguyen Van Toan
Faculty of Information Technology
National University of HCM City
227 Nguyen Van Cu, Dist. 5, HCM City, VIETNAM
ddien@saigonnet.vn

Abstract

Word segmentation is the first and obligatory task for every NLP. For inflectional languages like English, French, Dutch,.. their word boundaries are simply assumed to be whitespaces or punctuations. Whilst in various Asian languages, including Chinese and Vietnamese, whitespaces are never used to determine the word boundaries, so one must resort to such higher levels of information as: information of morphology, syntax and even semantics and pragmatics. In this paper, we present a model combining WFST (Weighted Finite State Transducer) approach and Neural Network. This word segmentation system is applied to Text-to-speech of Vietnamese and POS-tagger of Vietnamese. We evaluate the performance by comparing its word segmentation results with the manually annotated corpus and its performance proves to be very good. Our algorithm achieves 97% of accuracy on a corpus of Vietnamese Electronic Textbooks.

1 Introduction

Vietnamese is an isolated language, however, unlike such other insolated languages as Chinese, Thai, Vietnamese is written in extended Latin characters. So, the treatment of other languages cannot be mechanically applied to Vietnamese and one of the pending works in the NLP of Vietnamese at present is identifying the word boundaries.

In linguistics, word is a basic unit. Therefore, in order to computationally process Vietnamese, the first and foremost is determining the word boundaries, which is still pending for Vietnamese now.

Unlike Euro-Indian languages, where *"Word is a group of letters having meaning separated by spaces in the sentence." (Definition in Webster Dictionary),* in Vietnamese and other Asian languages, whitespaces are not used to identify the word boundaries. However, the word segmentation is indispensable due to such various applications as: Search engines, Word processors, Spelling Checkers, Voice Processing, etc.

Therefore, this has become an interesting matter for the circles of linguistics and computer science. The following points must be resolved to proceed with the word segmentation:

a. Local ambiguity in compound words.
b. No comprehensive dictionaries.
c. Recognition of proper nouns and names.
d. Morphemes and reduplicatives.

2 General Instructions of Vietnamese linguistics

2.1 "Tiếng" (Vietnamese syllable / morpheme)

Vietnamese has a special linguistic unit called "tiếng" (equivalent to hanzi of Chinese) which is similar to traditional morphemes in respect of content and similar to traditional syllables in respect of form. Unlike the hanzi of Chinese, each "tiếng" of Vietnamese has one and only one way of pronunciation. "Tiếng" is the basic unit in Vietnamese and it is constructed from phonemes under the following structure (Đinh Lê Thư, 1999). For example: "toán" (math / group)

Initial letter = t	tone mark = ‘ (acute)		
	Pre-sound = o	main sound = a	final sound = n

"Tiếng" may be:

- A word, e.g. "chị" (sister), "tôi" (I)

- A morpheme, e.g. “hoa” (flower) and “hồng” (pink) in the word “hoa hồng” (rose).
- A sub-morpheme, e.g. “bù” and “nhìn” in the word “bù nhìn” (puppet).

For simplicity, we can consider “tiếng” as “Vietnamese morpheme”, or “Vietnamese syllable” or “syllable” in short.

2.2 Words

There exist various definitions of Vietnamese words but all the linguists reach the unanimous agreement on the following points (Đinh Điền, 2000):

a. They must be integral in respects of form, meaning and be independent in respect of syntax.

b. They are structured from “tiếng” (Vietnamese morpheme/syllable).

c. They consist of simple words (1-tiếng, monosyllable) and complex words (n-tiếng, n < 5, polysyllable) , e.g. reduplicatives and compounds.

For example: “chị” (sister) : simple word; “hoa hồng” (rose): compound word; “chúm chím” (smile slightly) : reduplicative word (repeated initial consonants and/or tones); …

3 Previous Works

At present, there have been some models for Vietnamese which prove not to be so practical whereas for such other isolated languages as Chinese, Thai, this matter has been resolved in a rather acceptable manner with the following approaches:

3.1 Rule-based approach

This approach is well shown in the following models.

a. Longest Matching, Greedy Matching Models (Yuen Poowarawan, 1986 ; Sampan Rarunrom, 1991).

b. Maximal Matching Models.

This model is divided into “forward maximum match” and “backward maximum match”, for which the fully comprehensive dictionary is indispensable. However, it is obvious that there is no comprehensive dictionary and depending on different contexts that this model requires suitable dictionaries.

- Thai, Sornlertlamvanich (1993).
- Chinese, Chih-Hao Tsai (1996), MMSeg 2000; accurate 98% in a corpus with 1300 simple sentences without solution for proper nouns and unknown words.

3.2 Statistics-based approach

This approach is based on the word context in considering the information of the neighboring words to issue relevant decisions. There are two points to be resolved for this approach, which are the context width and the applied statistical method. As far as the context width is concerned the wider the more complex.

As far as the statistical method is concerned, the hidden first-order Markov has always been applied. However, this method greatly depends on the corpus for its training. In case one method is applicable for the political corpus, it cannot be applied to literal ones. In addition, there are some words of high probability but of syntactical function only, which lessens the role of probability.

a. HMM, based on Viterbi algorithm (Asanee Kawtraku, 1995 ; Surapant, 1995).

b. Expectation-Maximization (EM). This method is based on the resolvement of the “chicken and egg” question through its repetition (Xianping, 1996).

3.3 Other approaches

Most of these approaches are hybrid in combination with such other linguistic models as WFST, TBL (Transformation-Based Error driven Learning) (Julia, 1996). However, due to the requirement of various manipulations, the processing becomes quite time-consuming but its accuracy proves to be high.

However, the linguistic knowledge, which is much applied in the rule-based models, is rarely found in all the above models.

4 Our Model

Based on the definition of the Vietnamese words in section 2, we suggest a model (Fig.1), which can meet all the requirements of the above-mentioned definition as follows:

At first, we input a sentence into the pre-processing stage where we eliminate all the errors of sentence presentation. In addition, what is more important is the normalization of

accenting, the way of wording y, i… in Vietnamese. Due to no unanimous standardization, there exist some Vietnamese syllables with different writing but with identical meaning and pronunciation, e.g.*: thời kỳ = thời kì* (stage) *, hoà = hòa* (equal), etc.

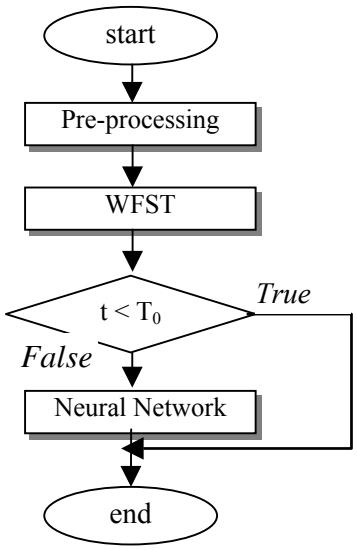

Figure 1: Flowchart of our model.

Then, that sentence is introduced to the WFST model where reduplicatives, proper nouns, date-time, numbers,... are further identified (One of the characteristics of Vietnamese is that all initial letters of proper names must be capitalized). In case of any further ambiguity, the Neural network model, which is our new approach, will be applied.

4.1 WFST Model

Vietnamese word segmentation can be considered as a stochastic transduction problem. We apply WFST model for Chinese Word segmentation into our task as follows (Richard Sproat, 1996):

We represent the dictionary D as a Weighted Finite State Transducer. Supposed:

H: set of "tiếng" (syllables).

p: no use, due to characteristic of "tiếng" (see 2.2.a).

P: set of grammatical Part-of-speech (POS) labels.

Each arc of D maps is either from an element of H to an element of H, or from ε (the empty string) to an element of P.

Each word in dictionary D is represented as a sequence of arcs, starting from the initial state of D (labeled with an element S of H) and terminated with a weighted arc labeled with an element of $\varepsilon \times P$. The weight represents the estimated cost of the word.

Next, we represent the input sentence as an unweighted Finite-State Acceptor (FSA) *I* over H. Let us suppose the existence of a function Id, which takes as input an FSA A and produces as output a transducer that maps all and only the strings of symbols accepted by A to themselves. Finally, we define the best segmentation to be the sentence with the smallest weight or best path in $Id(I) \times D^*$.We have improved this WFST model to make the Vietnamese word segmentation more convenient with the following peculiarities:

4.1.1 Dictionaries

The dictionary is arranged in the multiway tree (Fig.3). In case of multiway trees, more memory will be occupied, the binary tree is recommended instead to represent the dictionary of multiway trees. In the dictionary, each node represents a Vietnamese letter and for every word, we attribute to it such additional details as POS, word frequency, and syntactic features.

As mentioned above, the dictionary consists of a sequence of nodes and arcs. Each word is terminated with an arc describing a transformation between ε and their POS. In addition, an estimated weight is also attributed. In case of considering the probability of a word as the weight, it is difficult for the calculations to be executed due to their too small figures. Therefore, the weight is assigned through the logarithm of the probability of a concrete word:

$$C = -\log\left(\frac{f}{N}\right)$$

where f : word frequency; N: size of corpus.

Some words might have more than one POS. Therefore, we represent the POS of a word with an integer, in which each bit corresponds with a certain POS (Fig.2). As such, in case a word possesses more than one POS, more than one bit of its representing integer will be on.

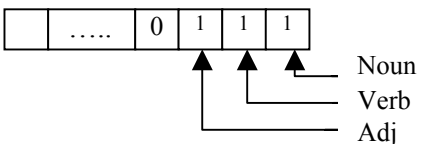

Figure 2: *Array of integers to represent POS*

At present, the classification of Vietnamese POS is still under argument, however, we only choose the following POS (Hoàng Phê, 2000) which are:

Noun *verb*
Adjective *Pronoun*
Preposition *Conjunction*
Adverb *Interjection*
Particle *Abbreviation*
Idiom and other symbols

For example, an entry in our dictionary:

```
struct entry
{
    string word;       // e.g. "bàn"
    int  POS;          // e.g. 3  (N+V)
    float frequency;   // e.g. 4.21
    string  syn_fea;  // e.g.  N_concrete,...
}
```

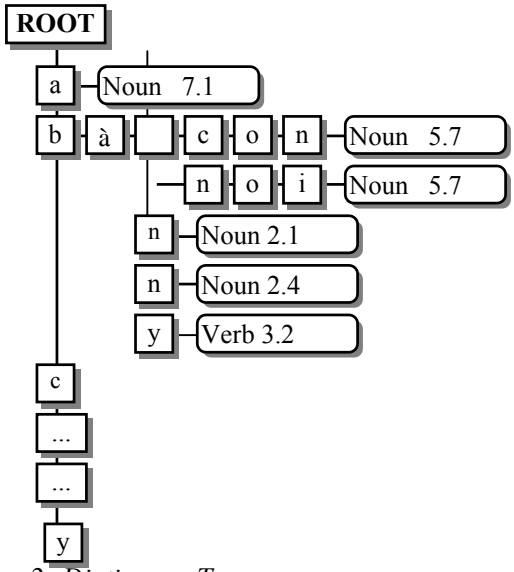

Figure 3: *Dictionary Tree*

The probability of words is calculated based on a corpus of 2,000,000 words. In fact, we have constructed a dictionary of 34,000 words based on the one of the Center of Lexicography (under the National Center of Social Sciences and Humanities). In addition, we have calculated the probability and searched for new words based on the corpus of 2,000,000 words consisting of:

- 1.6 MB from Complete works of Ho Chi Minh.
- 0.6 MB from Vietnam PC-WORLD magazines.
- 0.9 MB from newspapers in Science and Technology.
- 0.5 MB from famous works of Vietnamese poets.
- 3.7 MB from Vietnamese literary works.

4.1.2 Identification of proper names

One of the advantages of Vietnamese compared with such morphosyllabic languages as Chinese, Thai is the capitalization of all the initial characters of proper names, which makes the identification of proper names easier.

However, the ambiguity here is that the initial letter of a sentence is also capitalized and besides, as mentioned above, since the syntax of Vietnamese is rather complicated and not standardized yet, we are facing also with a lot of difficulties for the processing in this stage. For example: the following word is a proper name with 3 different acceptable forms: Ex: *Bộ Chính trị, Bộ chính trị, or Bộ Chính Trị (politburo).*

We make use of heuristic to attribute appropriate weights to these words and then consider them as conventional words to be processed at WFST with a very satisfactory result (further refer to the conclusion) and this is also a new approach based on the peculiarity of Vietnamese writings.

4.1.3 Proper names

We have also studied the names of the Vietnamese in the large-scale corpus and found out some peculiar rules which have also been applied by Richard Sproat for Chinese .

1. *word → name*
2. *name → Family Given1*
3. *name → Family Given2*
4. *Family → Family1*
5. *Family → Family2*
6. *Family1 → VN- syllable*
7. *Family2 → VN-syllable, VN-syllable*
8. *Given1 → VN- syllable*
9. *Given2 → VN-syllable, VN-syllable*

E :- Hồ Chí Minh (Family1 + Given2)
- Nguyễn Du (Family1 + Given1)
- Lê Nguyễn Trang Đài (Family2 + Given2)

4.1.4 Identification of Reduplicatives

One major feature of Vietnamese is there is a large number of reduplicatives which are also frequently multiplied in the course of communication. No dictionary can be comprehensive enough with all these reduplicatives due to no exhaustive statistics. Here, we make use of the rule of morpheme transformation in reduplicatives to identify them. Ex :

- *lèo tèo* (scattered), *lẩm bẩm* (murmur): only initial consonant is changed.
- *hổn hển (pant), chúm chím* (smile slightly): only main sound is changed.
- Other cases: one/many component(s) of "tiếng" will be changed but at least one component is kept unchanged.

4.1.5 Morphological analyzer

However, one inevitable thing is that no dictionary is comprehensive. There is also in Vietnamese a class of words which are not available in the dictionaries due to their morphological aspect. Those are the words morphologically derived (R.Sproat,1996), e.g.:

• cố gắng *(attempt, v)*→ *sự* cố gắng *(attempt, n)*

• hiện đại *(modern ,a)*→ *hiện đại hóa (modernize)*

• chủ tịch *(president,n)* → *phó* chủ tịch *(vice-president), ...*

Similar to English, there appear also prefixes and suffixes, which are however much simpler in the morphology of Vietnamese. Therefore, we apply further morphological analysis to easily identify this class of words. The critical point here is to determine the weight of these derived words (due to their unavailability in the dictionary). The weight of these new words will be calculated through the application of the conditional probability of Good-Tuning (Baayen). Supposedly, we need to calculate cost(ABC), in which AB is the radical and C is the suffix. Given *p(C): probability of C; p(unseen(C)): probability of C in which C is next to AB)* (Fig.4).

$$\Rightarrow p(unseen(C)) = p\left(unseen(C)/C\right) * p(C)$$

$$\Rightarrow cost(ABC) = cost(AB) + cost(unseen(C)) \text{ with}$$

$$cost(X) = -\log(p(X))$$

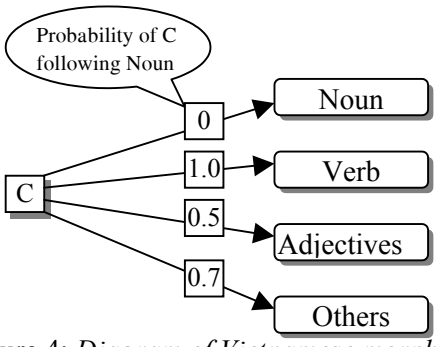

Figure 4: *Diagram of Vietnamese morphology*

Therefore, as for the words in the dictionary with prefixes and suffixes (temporarily referred to as C), we further store the probability of C when it is located right after a certain POS.

4.1.6 Method of segmenting sentences into sequences of words

The point here is to decrease the combinatorial explosion in the generating of the sequences of possible words from a string of syllables in a sentence. Supposed in an n-syllable sentence (in fact, the longest word in Vietnamese is fewer than 5 syllables), we will have at most 2^{n-1} different word segmentations. And in case every Vietnamese sentence has 24 syllables on the average, we must tackle 8,000,000 possibilities of word segmentation.

The new method we suggest here is making combined use of the dictionary to restrict the generating of these combinatorial explosions. When finding out that a certain word segmentation is not appropriate (not available in the dictionary, not reduplicatives, not proper names,…), we will eliminate the branches originated from that word segmentation by calculating in advance its weight in a sentence. With this method, we will restrict the number of possible segmentations to hundreds of cases (in comparison with millions of cases).

And in order to avoid too much repeated dictionary consultation, we make use of the above check to create the possibilities of a word in a sentence and separately store all their concerned details of POS, probability… for the convenience of future evaluation. Naturally, this separated storage will be of small size (approximately some hundreds of elements). And the dictionary consultation up to this moment has been terminated.

4.1.7 Method of selecting the best sentence

After achieving a set of possible word segmentations of a sentence, we usually can select the best word segmentation through the algorithms of Like-Viterbi. We made use of a simple method by selecting the word segmentation with the smallest total weight. For example : Input = "*Tốc độ truyền thông tin sẽ tăng cao.*" (The speed of information transmission will increase). In the dictionary, we have:

= "*tốc độ*" (speed) 8.68
= "*truyền*" (transmit) 12.31
= "*truyền thông*" (communicate) 12.31
= "*thông tin*" (information) 7.24
= "*tin*" (news / information) 7.33
= "*sẽ*" (will) 6.09
= "*tăng*" (increase) 6.95
= "*cao*" (high) 6.95

$Id(I) \circ D^*$ =" *Tốc độ # truyền thông # tin # sẽ # tăng # cao*." 48.79 (1)

=" *Tốc độ # truyền # thông tin # sẽ # tăng # cao*." 48.70 (2).

BestPath =" *Tốc độ # truyền # thông tin # sẽ # tăng # cao*." 48.70

(1):8.68+12.31+7.33+6.09+7.43+6.95= 48.79.

(2):8.68+12.31+7.24+6.09+7.43+6.95= 48.70.

In case of applying this model only, we are able to segment most of the sentences without any ambiguity as well as the scientific and technical material. However, there remain large classes of sentences with the structural ambiguities, which cannot be completely resolved by this model.

4.2 The Neural network model

4.2.1 Role of Neural network

After the word segmentation through the WFST model, we define a threshold value t_0 as follows to determine the accuracy of the above segmentation : In case the weight difference between the segmented sentences and the one with the smallest weight is more than t_0, the above segmentation result proves to be quite accurate. Otherwise, the WFST model is still not sufficient to determine the word boundary and then we have to further process these sentences through the neural network model.

For example, we consider this sentence "*Học sinh học sinh học* " (*Pupils learn biology*). After the WFST processing, only the three following sentences are left (due to too small weight difference, others are not mentioned here:

1. Học sinh(N) học(V) sinh học(N)
(Pupil | learn | biology)
2. Học sinh(N)học sinh(N)học(V)
(Pupil | pupil | learn)
3. Học(V) sinh học(N) sinh học(N)
(Learn | biology | biology)

In fact, there is sometimes a sequence of POS in Vietnamese which cannot serially stand next to each other as in case of adjectives and nouns and the application of the rule-based model to resolve any ambiguities here might be not appropriate. Even if the rule-based model is used, we face with a very tough question of how many rules to be applied. In case of too few rules, some correct sentences might be ignored and in case of too many rules, all the sentences are understood to be correct.

In order to resolve this problem, we have proceeded with the machine learning for the ambiguous sentences through the neural network model. We make use of this model to evaluate the suitability of the sequence of POS in a sentence and let's examine the above example again where our suggested neural network model is used to evaluate 3 sequences of POS : NVN, NNV, VNN. However, when calculating the weight of a word of more than one POS we examine all these cases.

This model will be trained by the ambiguous sentences after being processed with the first model. These ambiguous sentences will be manually segmented to be trained by the computer. In order to check the appropriateness of a sequence of POS in a sentence, we make use of a "k context" for each word in the sentence with a window of k words and its description will slide on the concerned sentence from the first to the last word of that sentence.

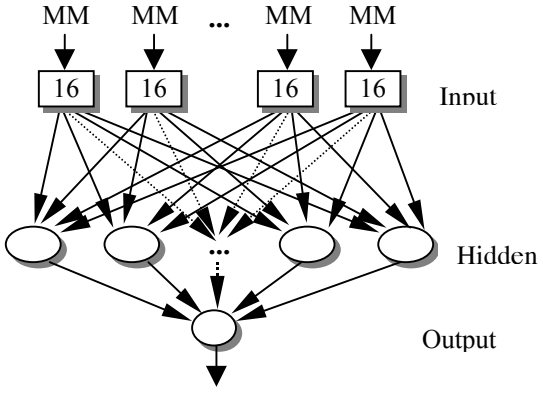

Figure 5: *MM: Descriptor Array for one word*

Actually, our network model consists of 6 input nodes, 10 hidden nodes and 1 output nodes (Fig.6). Every input node is a 16-dimension vector (This vector is described as an integer at the installation) representing the POS of a word as mentioned at 4.1.1. All the punctuations are considered as POS and a certain value will be attributed according to its form of punctuations.

The input layer of the neural network is fully linked to 10 hidden nodes with a propagation function. These hidden nodes are further fully linked to an output layer. The output node is a real value ranging from 0 to 1 representing the appropriateness of a sequence of POS standing next to each other in a sliding window. When this window slides from the beginning to the end of a sentence, we cumulatively add all the results to be attributed as the weight to the sentence. The function sigmoid $f(h_i) = \frac{1}{1 + e^{-h_i / T}}$, which is quite popular in all the neural networks, is chosen as the propagation function. The selected sentence is the one with the maximum weight.

4.2.2 The parameters in the Neural network

- The number of hidden nodes:

To determine the optimal number of nodes of the hidden layer for learning the sequence of POS in a sentence so that the word segmentation of good syntactical structure can be verified, we have tried checking various values of the size of the layer of hidden nodes to obtain the statistical result as in the following table:

Table 1: The result of sentences inside the training corpus:

#nodes Hidden	wrong #1	Correct	wrong #2
1	**2.13**	**1.64**	**0.78**
2	0.92	2.04	0.66
3	0.79	2.21	1.04
4	1.38	2.41	1.5
5	1.88	2.07	0.87
6	1.02	1.63	1.02
7	1.74	1.89	0.77
8	2.16	2.96	1.43
9	1.68	2.41	1.14
10	0.4	3.21	0.61
11	1.66	2.25	0.89
12	**1.63**	**1.62**	**0.74**
13	2.19	2.96	1.48
14	**1.49**	**1.45**	**0.48**
15	**2**	**1.88**	**0.84**

- As for the sentences inside the training corpus:

It is realized that in case the hidden nodes is 10, the difference between the correct and wrong sentences is maximum. Especially, in case the hidden node is 1 or more than 12, the result is opposite. In this case, we select the repetition to be 1000 (Table 1).

- As for the sentences outside the corpus:

It is also realized that in case the hidden nodes is 10, the difference between the wrong and the correct sentences is maximum. In this case, we also select the repetition to be 1000.

Table 2: The result of sentences outside the training corpus:

#nodes Hidden	#correct (1)	#wrong (2)	(1) – (2)
1	3.65	1.78	1.87
2	2.91	1.09	1.82
3	2.91	1.38	1.53
4	3.45	1.98	1.47
5	4.16	1.98	2.18
6	2.01	1.18	0.83
7	4.16	1.86	2.3
8	5.37	2.27	3.1
9	4.42	1.74	2.68
10	**3.98**	**0.64**	**3.34**
11	4.06	1.47	2.59
12	3.62	1.55	2.07
13	5.36	2.3	3.06
14	3.08	1.01	2.07
15	4.91	1.92	2.99

5 Results

5.1 Experiment

Applying the above model in processing unrestricted corpus, we have achieved the following results (Table 3)

Table 3: The result of evaluation:

	Number of training sentences	Number of correct sentences	Number of incorrect sentences	Ratio (%)
Techno - Science	550	541	9	98.36
Novels	150	142	8	94.67

(Note : the correct sentence must have all the correct segmentation).

Especially, this model can well segment the words in the sentence: "*Học sinh học sinh học." (Pupils learn biology) a*s well as the derived sentences. In our experimental results, the Neural Net has improved the original WFST model approx. 5-10% depending on the style of texts.

5.2 Reasons of mistakes committed

Most of the mistakes committed in applying this model are due to no exhaustive dictionaries and overall ambiguities and here are some overcoming solutions: Upon experimenting the program, we have found out that nearly 500 conventional words are still not available in the dictionary. Therefore, more entries have been introduced into the dictionary to enhance the reliability of this program. And one unavoidable thing is that we are not able to fully determine all the overall ambiguities in a sentence, for example: input = *Ông già đi nhanh quá*.
→ *Ông # già đi# nhanh quá. (The grandfather gets old so fast)*
→ *Ông già# đi# nhanh quá. (The old man goes so fast)*. Both of them are plausible. The correct one depends on context.Even human beings find it difficult to determine where the word boundary is and this matter cannot be resolved until the computer has read through and fully understood the full paragraph. Additionally, this model has sometimes made unreasonable segmentations due to its incorrect morphological analysis.

6 Conclusion

Even though there is still no complete corpus in Vietnamese (as an objective condition, in other countries, complete corpus have been created for the research), we have been trying our best to proceed with the collection for a corpus sufficient for our thesis as well as in other works. And one more difficulty is that since there is still no unanimous and standard norms on words, the results of the word segmentation are not able yet to satisfy everybody's requirements.

However, in considering the basic norms of words as mentioned in part 2.2, the result is quite satisfactory. As for the first requirement, we have made use of a reliable dictionary with the datamining of corpus to further recognize the concerned words. That is why our model has met the first requirement. As for the third requirement, we are also able to further determine some reduplicatives through the model of reduplicatives. And the model of compound words has been verified through the WFST model of probability. Incidentally, this model is helpful in determining the concerned words through the sentence structure and the sequence of words. Therefore, through the application of the linguistic knowledge as well as the probability (Neural network...), we have improved the WFST method to become more easily understood, more simple and better applied in the concrete background of Vietnamese to obtain a satisfactory result.

Additionally, we have tried combining WFST model and trigram model but its result turns out to be not so good as that of the combination of WFST and Neural Net model.

Finally, this model can be further improved (through the proper adjustment of thresholds) to be more extensively applied in various fields. We expect that this model as well as our program can be served as the first stage of sound foundation to effectively assist such future Vietnamese processing programs as POS-tagger, Machine Translation, etc.

References

Asanee Kawtraku. 1995. *Alexibase Model for Writing Production Assistant System.*

Chih-Hao Tsai. 1996. *MMSEG: A Word Identification System for Mandarin Chinese Text Based on Two Variants of the Maximum Matching Algorithm.* www.casper.beckman.uiuc.edu/~c-tsai4/chinese/wordseg/mmseg.html.

Đinh Điền.2000. *Từ tiếng Việt. (Vietnamese words)* VNU-HCMC.

Đinh Lê Thư. 1999. *Cơ cấu ngữ âm tiếng Việt*. (Structure of Vietnamese phonetics). VNU-HCMC.

Hoàng Phê. 2000. *Từ điển tiếng Việt*. (Vietnamese Dictionary). Center of Lexicography, Institute of Linguistics. Đà Nẵng.

Julia Hockenmaire, Chris Brew. 1996. *Error-Driven Learning for Chinese Word Segmentation.*

Richard Sproat et al. 1996. *A Stochastic Finite-State Word Segmentation Algorithm for Chinese*. ACL Vol 22 N3.

Sampan Rarunrom.1991. *Dictionary-base Thai Syllable Separation.*

Surapant Meknavin. 1995. *Towards 99.99% Accuracy of Thai Word Segmentation* . 1995.

Xianping Ge, Wanda Pratt, Padhraic Smyth. 1996. *Discovering Chinese Words from Unsegmented Text.*

Yuen Poowarawan. 1986. *Dictionary-base Thai Syllable Separation.*