Word segmentation of Vietnamese texts: a comparison of approaches

ĐINH Quang Thắng[*], **LÊ Hồng Phương**[*†], **NGUYỄN Thị Minh Huyền**[*], **NGUYỄN Cẩm Tú**[*], **Mathias ROSSIGNOL**[‡], **VŨ Xuân Lương**[⋆]

[*] Vietnam National University of Hanoi, Vietnam
[†] LORIA, France
[‡] MICA, Hanoi, Vietnam
[⋆] Vietlex, Hanoi, Vietnam
dqthang@vnu.edu.vn, lehong@loria.fr, huyenntm@vnu.edu.vn, ncamtu@vnu.edu.vn,
mathias.rossignol@mica.edu.vn, vuluong@vietlex.com

Abstract

We present in this paper a comparison between three segmentation systems for the Vietnamese language. Indeed, the majority of Vietnamese words is built by semantic composition from about 7,000 syllables, that also have a meaning as isolated words. So the identification of word boundaries in a text is not a simple task, and ambiguities often appear. Beyond the presentation of the tested systems, we also propose a standard definition for word segmentation in Vietnamese, and introduce a reference corpus developed for the purpose of evaluating such a task. The results observed confirm that it can be relatively well treated by automatic means, although a solution needs to be found to take into account out-of-vocabulary words.

1. Introduction

Despite the fact that, for historical and practical reasons, a variant of the Latin alphabet is now used to represent Vietnamese, its linguistic mechanisms remain close to that of languages using syllabic alphabets, like Chinese. In particular, the Vietnamese language creates words of complex meaning by combining syllables that most of the time also possess a meaning when considered individually. That creates problems for all NLP tasks, due to the difficulty in identifying what constitutes a word in an input text.

We present in this article three systems developed by separate research teams to address that issue, and compare their performance on a corpus of about 1,500,000 words manually segmented for the purpose of this experiment.

The two first systems are based on the principle of maximum matching, that is, the search for the combination of words that produces the segmentation having the smallest number of words. The first one, *vnTokenizer*, completes this principle by relying on statistical textual data (word and bigram frequencies) to deal with possible ambiguities (Lê et al., 2008). The second, *PVnSeg* does not modify the maximum matching algorithm, but performs heavy pre- and post-processing of segmented files using pattern matching techniques.

The third system, *JVnSegmenter*, adopts for its part a radically different approach, employing statistical machine learning techniques to identify word boundaries from local contextual characteristics of the text.

We first present in Section 2. an overview of word segmentation in various languages. Section 3. is then dedicated to the description of our corpus and the specification of the type of segmentation we wish to achieve. Sections 4. to 6. presents the three systems in greater detail, before proceeding to Section 7., containing the description of the experimental setup and the result of the tests. We conclude in Section 8. with a few teachings for future research in that field.

2. Existing works on word segmentation

Inflected languages (typically, western languages) also have the problem of compound words, but it lies in the identification of stabilized syntactic constructs that refer to a very precise meaning. Those words are often not present in dictionaries, and their relevance may be limited to a specific domain, which is why such research is mostly met in the field of terminology extraction (Kageura et al., 2004). By contrast, in isolating languages compound words belong to the core of the language; they are present in dictionaries and extremely frequent (in Vietnamese, 28,000 compound words in a 35,000-word dictionary). Therefore, we believe the problems to be quite distinct, and shall focus in this section on Asian languages.

The task of segmentation can be made more or less difficult by the writing system: in Thai, for example, each syllable is transcribed using several characters, and there is no space in the text between syllables (Kawtrakul et al., 2002). The problem of word segmentation is thus double: first, syllable segmentation, then word segmentation itself. For Chinese or Vietnamese, the situation is easier, since basic lexical units are easily identifiable: Chinese *hanzi* (Sproat et al., 1996) are each represented by one character, and Vietnamese *tiếng* are separated by spaces.

In (Ha, 2003), L. A. Ha separates the task of text segmentation into two sub-tasks:

- Disambiguation between possible word sequences using a lexicon and statistical methods (Wong and Chan, 1996).
- Identification of unknown words using collocation detection measures such as mutual information and t-score: that is the approach of (Sun et al., 1998) for Chinese and (Sornlertlamvanich et al., 2000) for Thai.

It can also happen that morphosyntactic analysis tools integrate their own segmentation rules based on syntactic evidence (Feng et al., 2004).

The tools presented in this paper are mostly concerned with the task of disambiguating between possible word sequences. Although some attempts are made to extend those results to unknown sequences presenting salient features (proper nouns, numbers, *etc.*), no work yet presents the ability to discover fully unknown compound words from corpus. Before delving further into the characteristics of those tools, we detail in the next section the exploited experimental data.

3. Experimental data

In order to perform a thorough evaluation and provide a reference corpus usable for further research, great care has been taken to properly specify the segmentation task. We therefore present in this section, first the specification of the segmentation task, then the contents and characteristics of our corpus.

3.1. Segmentation specification

We have developed a set of segmentation rules based on the principles discussed in the document of the ISO/TC 37/SC 4 workgroup on word segmentation (2006).
Notably, the segmentation of the test corpus follows the following rules:

Compounds: word compounds are considered as words if their meaning is not compound from their subparts (e.g. *xe*/vehicle, *đạp*/pedal - *xe đạp*/bicycle), or if their usage frequency justifies it.

Derivation: when a bound morpheme is attached to a word, the result is considered as a word (*học*/study - *tâm lí học*/psychology). The reduplication of a word (common phenomenon in Vietnamese) also gives a lexical unit (e.g. *tháng*/month – *tháng tháng*/ month after month.)

Multi-word expressions: expressions such as "*bởi vì*/because of" are considered as lexical units.

Proper names: names of people and locations are considered as lexical units.

Fixed structured locutions: numbers, times, and dates, which can be written in letters or numbers or using a mix of both, are recognized as lexical units (e.g. 30 – ba mươi/ thirty).

Foreign language words: foreign language words are ignored in the process of segmentation

3.2. Corpus constitution

Our test corpus gathers a selection of 1,264 articles from the "Politics – Society" section of the newspaper *Tuổi Trẻ*, for a total of 507,358 words that have been manually spell-checked and segmented by linguists from the Vietnam Lexicography Center (Vietlex).
The following sections provide detailed descriptions of the compared tools.

4. vnTokenizer

vnTokenizer implements a hybrid approach to automatically tokenize Vietnamese text. The approach combines both finite-state automata technique, regular expression parsing and the maximal-matching strategy which is augmented by statistical methods to resolve ambiguities of segmentation. The Vietnamese lexicon in use is compactly represented by a minimal finite-state automaton. A text to be tokenized is first parsed into lexical phrases and other patterns using pre-defined regular expressions. The automaton is then deployed to build linear graphs corresponding to the phrases to be segmented. The application of a maximal matching strategy on a graph results in all candidate segmentations of a phrase. It is the responsibility of an ambiguity resolver, which uses a smoothed bigram language model, to choose the most probable segmentation for the phrase.
vnTokenizer is written in Java and bundled as an Eclipse plugin. It is distributed under the GPL and freely downloadable from `http://www.loria.fr/~lehong/projects.php`.

5. PVnSeg

PVnSeg is a command-line tool for the segmentation of Vietnamese texts combining several simple programs written in Perl. Its basic operating principle is, once again, maximum matching, using a backtracking algorithm for increased efficiency. The specificity of PVnSeg is that it exploits the power of Perl for text analysis and pattern matching to implement a series of heuristics for the detection of compound formulas such as proper nouns, common abbreviations, dates, numbers, URLs, e-mail addresses, *etc.*
Work is underway to include the detection of other categories of standardized formulations, such as street addresses, and the automatic extraction from corpora of lists of common abbreviations. Emphasis is also put on intelligent punctuation segmentation using evidence such as capitalization, presence of numbers, of special characters...

6. JVnSegmenter

JVnSegmenter departs from the traditional maximum matching approach and uses statistical machine learning techniques to identify word boundaries in Vietnamese text. JvnSegmenter casts the word segmentation task as the problem of tagging sentences with three predefined labels: BW (beginning of a word), IW (inside a word) and O (others). Each sequence of tagged syllables in which the first one is tagged as BW and the others are tagged as IW forms a word. Two methods are presented: (1) Linear Conditional Random Fields with first order Markov Dependency and (2) Support Vector Machines with second degree polynominal kernel.
Two kinds of feature functions are used in linear CRFs: edge features which obey to the first Markov property, and per-state features which are generated by combining information concerning the context of the current position in the observation sequence (context predicate) with the current label. Based on the same idea, JVnSegmenter integrates two kinds of features into the SVM model, static features

and dynamic features. While SVM models decide upon dynamic features in the tagging process by considering the two previous labels, static features are very similar to vertex features in the CRF model, in that they also takes into account context predicates at the current observation.
Experiments presented in detail in (Nguyen et al., 2006) suggest that the best results are to be obtained by using the full set of defined features, both techniques (CRF and SVM) exhibiting comparable performance. In the tests presented in this paper, we have therefore exploited the same features and present results for the CRF approach only.

Now that we have described all considered systems, we present in the next section the devised experimental setup and obtained results.

7. Experiment

We present in this section the experimental setup used to compare the presented tools, as well as the segmentation comparison algorithm, in order to permit result comparison with other similar studies, and finally the obtained figures in Section 7.3..

7.1. Experimental setup

Some of the tools we wish to compare require a training phase. We have chosen to provide all systems with the opportunity to use training data if they need it, by performing a 10-fold cross validation.
In the case of JVnSegmenter, since it is distributed with pre-trained parameter files, we have computed performance both with those parameters and with parameters acquired from the training corpus.

7.2. Evaluation method

For each test run, the resulting segmented file is aligned with the hand-segmented reference by counting all non-blank characters; we then count all identical parallel tokens towards the global score.
Precision is computed as the count of common tokens over tokens of the automatically segmented files, recall as the count of common tokens over tokens of the manually segmented files, and F-measure is computed as usual from these two values.

7.3. Results

Table 7. presents the values of precision, recall and f-measure computed for all the considered systems.
The first interpretable result is that JVnSegmenter really needs to be trained for the considered task, which is not surprising since we cannot know whether the original model files were trained with the same segmentation rules.
From the relatively good results of PVnSeg, we can conclude that efforts at integrating lexical and linguistic knowledge in the tool, in the form of pattern-matching rules, are more fruitful than efforts to solve segmentation ambiguities. Indeed, that phenomenon seems, after closer sudy of the data, relatively rare. The majority of errors, for all systems, are due to the presence in the texts of compounds absent from the dictionary.
Finally, it should be noted that vnTokenizer is, of the three systems, the one with the most consistent results, *i.e.* the lowest standard deviation of performance between articles.

8. Conclusion

We have presented three systems for the segmentation of Vietnamese texts into words, and evaluated them on a reference corpus segmented by Vietnamese linguists. All three offer performance within a 2 % range around 95 %, with varying strengths and weaknesses. An important teaching of this experiment is that unknown compounds are a much greater source of segmenting errors than segmentation ambiguities, which are, after all, relatively rare. Future efforts should therefore be geared in priority towards the automatic detection of new compounds, which can be performed by means statistical (given a large enough corpus) or rule-based (using linguistic knowledge about word composition) or hybrid .

9. Acknowledgements

This work has been carried on in the framework, and with the support of the National Vietnamese project KC.01.01/06-10 on the development of essential tools and resources for Vietnamese language and speech processing.

10. References

J. Feng, L. Hui, C. Yuquan, and L. Ruzhan. 2004. An enhanced model for Chinese word segmentation and part-of-speech tagging. In *SIGHAN Workshop, Meeting of the Association for Computational Linguistics (ACL 2004)*, Barcelona, SP.

L. A. Ha. 2003. A method for word segmentation in Vietnamese. In *Proceedings of the International Conference on Corpus Linguistics*, Lancaster, UK.

K. Kageura, B. Daille, H. Nakagawa, and L.F. Chien. 2004. Recent trends in computational terminology. *Terminology*, 10(2):1–21.

A. Kawtrakul, M. Suktarachan, P. Varasai, and H. Chanlekha. 2002. A state of the art of Thai language resources and Thai language behavior analysis and modeling. In *Proceedings of the ACL-02 - Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, University of Pennsylvania, USA.

H. P. Lê, T. M. H. Nguyen, A. Roussanaly and T. V. Ho. 2008. A hybrid approach to word segmentation of Vietnamese texts. In *2nd International Conference on Language and Automata Theory and Applications*, Tarragona, Spain.

ISO/TC 37/SC 4 AWI N309. 2006. Language resource management - word segmentation of written texts for mono-lingual and multi-lingual information processing - part 1: General principles and methods. Technical report, ISO.

C. T. Nguyen, T. K. Nguyen, X. H. Phan, L. M. Nguyen, and Q. T. Ha. 2006. Vietnamese word segmentation with CRFs and SVMs: An investigation. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation (PACLIC 2006)*, Wuhan, CH.

System	Precision	Recall	F-measure
vnTokenizer	93.68 %	94.42 %	94.05 %
PVnSeg	96.89 %	96.21 %	96.55 %
JVnSegmenter (original)	85.22 %	81.40 %	83.27 %
JVnSegmenter (re-trained)	95.03 %	93.82 %	94.42 %

Table 1: Precision, recall and f-measure of the three systems for word segmentation.

V. Sornlertlamvanich, T. Potipiti, and T. Charoenporn. 2000. Automatic corpus-based Thai word extraction with the C4.5 learning algorithm. In *Proceedings of the International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, DE.

R. Sproat, C. Shi, W. Gale, and N. Chang. 1996. A stochastic finite-state word-segmentation algorithm for chinese. *Computational Linguistics*, 22(3):377–404.

M. Sun, D. Shen, and B. K. Tsou. 1998. Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of COLING-ACL 98*, Montreal, Quebec, CA.

P. Wong and C. Chan. 1996. Chinese word segmentation based on maximum matching and word binding force. In *Proceedings of the 16th conference on Computational linguistics*, Copenhagen, DK.