Using Search Engine to Construct a Scalable Corpus for Vietnamese Lexical Development for Word Segmentation

Doan Nguyen
Hewlett-Packard Company
doan.nguyen@hp.com

Abstract

As the web content becomes more accessible to the Vietnamese community across the globe, there is a need to process Vietnamese query texts properly to find relevant information. The recent deployment of a Vietnamese translation tool on a well-known search engine justifies its importance in gaining popularity with the World Wide Web. There are still problems in the translation and retrieval of Vietnamese language as its word recognition is not fully addressed. In this paper we introduce a semi-supervised approach in building a general scalable web corpus for Vietnamese using search engine to facilitate the word segmentation process. Moreover, we also propose a segmentation algorithm which recognizes effectively Out-Of-Vocabulary (OOV) words. The result indicates that our solution is scalable and can be applied for real time translation program and other linguistic applications. This work is here is a continuation of the work of Nguyen D. (2008).

1 Introduction

The Vietnamese language as a minority language is gaining popularity including content and audience. It is important to emphasize a need for natural language such as search engines or translation tools to process the data correctly. With this emphasis, we need to have a way to improve and automate the training process as well as expanding its training data. Previous works in constructing segmentation systems for the Vietnamese language relied on single source of information such as newspapers or electronic dictionaries (Le H. Phuong et al. 2008, Dinh Dien and Vu Thuy, 2006, Le T. Ha et al., 2005). Mono-source corpora would work best within their domain, and might not work well externally per O'Neil (2007). Le A. Ha, (2003) described the dictionary based approach as problematic due to the lack of consistency and completeness. This speaks to the need of standardizations between dictionaries, concrete grammar theories, and being up-to-date with the arrival of new words. In the work of Nguyen C. T. et al. (2007), corpus training was done manually by linguists. This was very time-consuming and costly. Because the task is performed only once, a corpus will go stale and will get out-of-date. Dinh et al. (2008), in a comparison with major Vietnamese segmentation approaches, concluded that the handling of unknown compound words is a much greater source of segmenting errors and underscored that future effort should be geared at prioritizing towards the automatic detection of new compounds.

In this paper, we first present the main issues with the Vietnamese word segmentation problem. We describe the two approaches in obtaining raw text from the Web. Then, we present our approach in building a large web corpus for a word segmentation function and compare our result against a sophisticated algorithm built on a human trained corpus. Finally, we provide our conclusion and offer suggestions for future research directions.

2 Vietnamese Word Segmentation Problems

Vietnamese (Tiếng Việt) is the official language of Vietnam. The current writing system originates from the Latin alphabet, with diacritics for tones and certain letters. Vietnamese is often mistakenly judged as a "monosyllabic" language. However, the majority of the words are disyllabic (Le A. Ha, 2003) covering reduplication and adjectives. Its grammar depends on word ordering and sentence structure rather than morphology. Even though there is a space separating

Proceedings of the 7th Workshop on Asian Language Resources, ACL-IJCNLP 2009, pages 171–178,
Suntec, Singapore, 6-7 August 2009.

sound units, there is nothing used to identify word boundary.

Examples in Figure 1. are used to illustrate the difficulty of Vietnamese word segmentation when compared it to English. There are 256 possible sequences (2^{n-1}) of segmentation in this example.

English: A woman sells tea along the road .
A | woman | sells | tea | along |the | road . (1)

Vietnamese: Một người đàn bà bán nước trà ven đường .
Một | người đàn bà | bán | nước trà |ven | đường . (1)
Một | người đàn bà | bán nước | trà |ven | đường . (2)
Một | người |đàn bà | bán nước| trà | ven | đường . (3)
Một | người |đàn | bà | bán nước| trà | ven | đường . (4)
And many others combinations ….

Figure 1. Ambiguity of word segmentation

The major segmentation problems with the Vietnamese word segmentation include: the handling of word ambiguities, detection of unknown words, and recognition of named entities.

2.1 Addressing Words Ambiguities

In a sequence of Vietnamese syllables, S, composing of two syllables A and B occurring next to one another, if S, A, and B are each words, then there is a conjunctive ambiguity in S. In contrast, in a sequence of Vietnamese syllables, S, composing of three syllables A, B, and C appearing contiguously, if A B and B C are each words, then there is a disjunctive ambiguity in S. In order to attain a higher precision rating, word ambiguity must be addressed.

2.2 Detection of Unknown Words

In a dictionary word segmentation based approach, only the words that are in the dictionary can be identified. The unknown words might belong to one of the following categories: (1) Morphologically Derived Word (MDW). There are some lexical elements that never stand alone, which express negation such as: "bất" in "bất quy tắc" (irregular) or transformation such as "hoá" in "công nghiệp hoá" (industrialize). (2) Interchanging usage of vowels i and y and changing in position of tone. For example: "dược sĩ" and "dược sỹ". Both mean "pharmacist". (3) Phonetically transcribed words. This can be seen in naturalized words like: "phô mai" (fromage), "híp hóp" (hip hop music), or "iPhône" (Apple iPhone).

2.3 Recognition of Named Entities

Unlike other Asian languages, Vietnamese personal, location, and organizational names all have the initial letter capitalized. For example: "Nguyễn Du" (a famous Vietnamese poet). Due to the language syntax standardization, a proper name could be written in many different forms. The following organizational name has three acceptable forms: Bộ Nông Nghiệp, Bộ Nông nghiệp, or Bộ nông nghiệp (Department of Agriculture). We use the following shape features (pattern) to assist with the recognition process:

Word Shape	Examples
Capitalized	Sài Gòn (Location)
All Caps	WTO (World Trade Organization)
Containing digit	H5N1 (Bird flu)
Containing hyphen	Vn-Index (Securities market of Việt Nam)
Mixed case	VnExpress (Vietnam News Daily)

Table 1. Word Shape features for identifying Vietnamese Name Entities

3 Using World Wide Web as a Resource to Build Corpora

There are two approaches to obtain linguistics data from the Web. The first approach is to crawl the web (Baroni et al., 2006 and O'Neil, 2007). This option gives flexibility in choosing or restricting sites to crawl upon. To have good coverage, it requires extensive hardware resource to support storage of content documented in the work of Baroni et al. (2006). Other complexities include a filtering capability to recognize content of a target language from crawling data, removing html code, and handling page duplication. The work of Le V. B (2003) indicated that it is very difficult to crawl on Web pages located in Vietnam due to a low network communication bandwidth.

A second approach is to use search engines via a web service API to find linguistic data. In the work of Ghani et al. (2001), a term selection method is used to select words from documents to use for a query. Documents from a search result list are downloaded locally to process and build corpus data. The technical challenges of this approach are: (1) Corpus being biased and being dictated by a ranking of a search engine. (2) Li-

mited number of search queries is allowed by a search engine per day.

4 Our Approach to build corpus

We are structuring our system with two main components. The first component works as a word training and recognition system. The second component utilizes the training information provided from the first component to perform just a word segmentation task by leveraging the computed lexical statistics. This is a clear distinction between our work and Nguyen D. (2008). Because there is a limited number of search request imposed by commercial search engines each day, this approach is not practical for a condition where there is constant usage of search requests, for word segmentation purpose. Aside from this limitation, lexical statistics have to be recomputed for each new word segmentation request.

Figure 2. depicts the overall system consists of two components: The training Processing includes a new word discovery function and Normal Segmentation process. The training process would execute continuously and feed the lexical statistics to the second process for segmentation task purely.

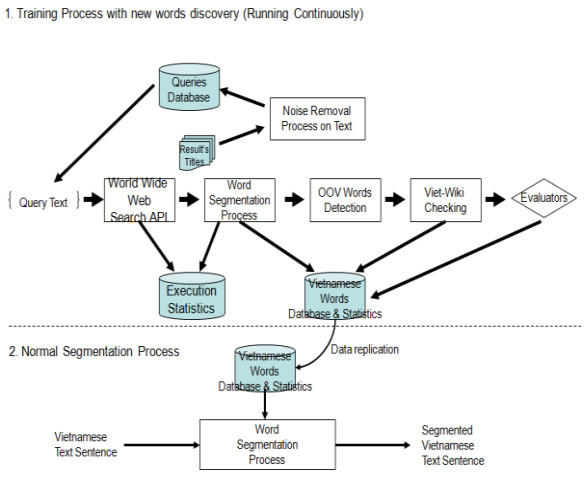

Figure 2. Vietnamese Words Corpus Construction Process

4.1 Word Training and Recognition System

This component trains identified words inside a Vietnamese Word Database with its frequency of occurrences. Newly encountered OOV words are recognized by the system then verified by a check against the Vietnamese Wikipedia programmatically. We do not wish to include all words from the Vietnamese Wikipedia as there are many foreign words. For examples: *St. Helens*, *Oregon*. The remained frequently found OOV words are evaluated by linguists for validity and will be included into the word database as confirmed. Unlike the work of Ghani (2001), in our work, a query to submit to an engine is a sentence derived from an unknown document title. The reason here is to enable the system to discover the unknown words and their frequencies naturally. This system performs:

- Seed the queries database with an initial set of queries, Q_n.
- Randomly select a query from Q_n and send to a search engine.
- From a search result list, process on document titles and snippet texts directly.
- Perform Vietnamese word segmentation on recognized sentences using question mark, exclamation mark, periods as separators. Update the word database with recognized segmented words and their computed frequencies and weights.
- Recognize and validate OOV words, using the Vietnamese Wikipedia or through morphological rules programmatically.
- Bootstrap Q_n with retrieved document titles.
- Return to step 2 above.

5 Word Segmentation System

In the Vietnamese language, as the white space cannot be used to denote word boundary, the function of a word segmentation system is to segment a sentence into a sequence of segmented words such that a context (or meaning) of a sentence is preserved.

5.1 Data Gathering and Words Extraction

In the first step, a search query is submitted to a search engine API and requests for **N** returned documents. The engine returns a search result list, which consists of document titles and their summary text. We parse the data and extract the required text. Syllables in the search query are then matched against the parsed text to extract potential words covering both monosyllabic and polysyllabic words. This function keeps track and counts their occurrences. At this stage, we also determine if a word is a proper name. We use the various word shape features in capitalization forms to assist with the recognition process. We compute the likelihood of extracted words to be proper names by taking the account of the number of identified capitalized words over the

total of the same words in appearing the documents set, N documents. Once the extraction process is complete, we perform additional validation steps to discard incorrect generated words. To be accepted as a potential word, a word must satisfy one of the following rules: (1) It appears in the word database. (2) It is recognized as a proper name word. (3) It is a MDR word. (4) It is an OOV word with strong world collocation as defined below.

An OOV word is identified when there is a strong collocation (cohesion) attached between its syllables. That is the following condition(s) is/are met: (1) For two syllable words to collocate: $P(s_1\ s_2) > P(s_1)P(s_2)$, (2) For three syllable words to collocate: $P(s_1\ s_2\ s_3) > MAX\{ P(s_1)P(s_2)P(s_3), P(s_1)P(s_2s_3), P(s_1s_2)P(s_3) \}$ where $w = s_1\ s_2\ s_3$, $P(s_1 \ldots s_n) = Freq(s_1 \ldots s_n)/N$, and N is the number of documents returning from a search engine.

Collocation concept has been utilized in the merging syllables to determine the best possible segment in the work of Wirote (2002).

Suffix	**Translation**	**Result Lexical Category**	**Morphological Rules**	**Examples**
học	"-logy, -ics"	Noun	**IF** Syllable_Suffix("học") **AND** Prefix_With_Word((Noun(W)) **THEN** WORD(W+ " "+ "học")	ngôn ngữ (language) + học → ngôn ngữ học (linguistics)
hóa	"-ize, -ify"	Verb	**IF** Syllable_Suffix("hóa") **AND (**Prefix_With_Word((NOUN(W)) **OR** Prefix_With_Word((ADJECTIVE(W))) **THEN** WORD(W+ " " +"hóa")	công nghiệp (industry) + hóa → công nghiệp hóa (industrialize)
Prefix	**Translation**	**Result Lexical Category**	**Morphological Rules**	**Examples**
sự	"Action-"	Noun	**IF** Syllable_Prexix("sự") **AND** (Suffix_With_Word((Verb(W)) **OR** Suffix_With_Word((Adjective(W))) **THEN** WORD(sự+ " "+ W)	sự + thảo luận (discuss, debate) → sự thảo luận (discussion)
bất	"Un-"	Noun	**IF** Syllable_Prexix("bất") **AND** (Suffix_With_Word((Verb(W)) **OR** Suffix_With_Word((Adjective(W))) **THEN** WORD(bất+ " "+ W)	bất + hợp pháp (legal, lawful) →bất hợp pháp (Not legal)

Table 2. Examples of derivational morphology and morphological rules to construct compound words

To recognize for morphological derived words (MDW), we have identified a range of prefixes and suffixes (Goddard, 2005). When a morpheme modifies another morpheme, it produces a subordinate compound word (Ngo, 2001). For example: nhà (as a prefix) + báo (newspaper) → nhà báo (journalist). The table 2. provides a few examples of Vietnamese suffixes, prefixes, and Morphological Rules to derive subordinate compound words.

5.2 Sentences Construction

Given a set of potential segmented words obtained from step 5.1, applied only for training process or for a normal segmentation process (Figure 2.), the task of sentences constructor is to assemble the identified words in such a way that they appear in the same order as the original query. We use Greedy algorithm to construct sentences using the following heuristic strategies: (1) Selection of polysyllabic words over monosyllabic words whenever possible. (2) Eliminating segments which have already examined. (3) Declaring a solution when a constructed sentence has all of segmented words appearing in the same order as in the original query text.

5.3 Sentences Refinement and Reduction through Ambiguity Resolution

Since there is only a single solution to present to a user, we need to have an algorithm to improve upon proposed sentences and reduce them to a manageable size. The algorithm **Sentences_Refine_Reduce** below describes the

steps in refining the sentences to a finer solution(s).

Definition: Let the pipe symbol, |, be designated as a boundary of a segment. Two segments, in two sentences, are overlapped if their first and last syllables are: (1) located next to a segmented boundary. (2) Identical and positioned at the same location. For example, in the following two sentences:

Sentence #1: tốc độ | truyền | thông tin | sẽ | tăng | cao
Sentence #2: tốc độ | truyền thông | tin | sẽ | tăng | cao

The overlapped segments are: "tốc độ", "sẽ", "tăng", and "cao". We are now describing an algorithm to perform sentences refinement and reduction as follows:

Algorithm : Sentences_Refine_Reduce()	
Input: SBuffer - for input sentences **Output:** SBuffer - for output sentences	
1:	**Until** Converged(SBuffer) **Do**:
2:	Itr_Sentences_Buf = {}
3:	**For** si in SBuffer **Do**:
4:	Find sj such that Max {|Overlapped_Segment (si,sj)|} for sj ∈ SBuffer and si != sj
5:	Res_Segments=Overlapped_Segments(si,sj) U Conjunctive_Segments_Resolutions(si,sj) U Disjunctive_Segments_Resolutions(si,sj)
6:	Itr_Sentences_Buf = Itr_Sentences_Buf U Sentence(Res_Segments)
7:	SBuffer=(SBuffer!=Itr_Sentences_Buf) ? Itr_Sentences_Buf : SBuffer

For conjunctive ambiguity resolution, to determine if all syllables should be classified as a single word or appeared as individual words, we utilize word collocation strength. We define collocating strength as follows.

$$P(s_1...s_n) = \frac{Freq(s_1...s_n)}{N} \qquad (2)$$

We compare it against a probability of finding the syllables occur independently in N documents as shown in equation (3). The outcome determines if the syllables should be collocated or separately appeared:

$$P(s_1)...P(s_n) = \frac{Freq(s_1)}{N} \times ... \times \frac{Freq(s_n)}{N} \qquad (3)$$

For disjunctive ambiguity resolution, because a determination involves multiple words with overlapping text, we determine the best possible segments by computing their probability distribution of word segments to find out which one has the highest probability of success. This is discussed further in the section "Sentences Scoring and Ordering" below. Figure 3 illustrates a process where sentences are refined through disambiguating words.

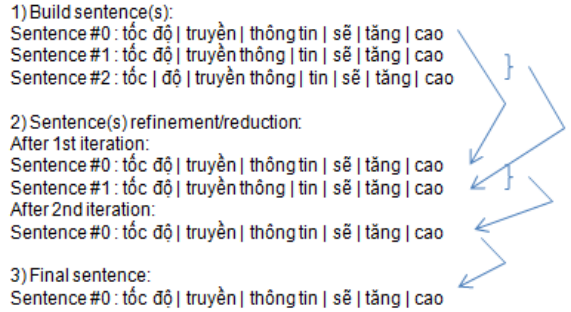

Figure 3. An Example of Sentences Refinement

After the 1st iteration, the sentences 1 and 2 are combined through a resolution of conjunctive ambiguity between "tốc độ" vs. "tốc | độ" . After the 2nd iteration, sentences 1 and 2 are combined through a resolution of disjunctive ambiguity between "truyền | thông tin" vs. "truyền thông | tin". The process exits when a converged condition is reached. The final segmented sentence is translated in English as "The speed of information transmission will increase".

5.4 Sentences Scoring and Ordering

The task in this phase is to score and order the candidates. A language model is usually formulated as a probability distribution p(s) over strings s that attempts to reflect how frequently a string s occurs as a sentence in a corpus, Chen et al. (1998). For a segmented sentence $S = w_1 w_2 ... w_n$, where w is an identified segmented word, using a bigram model, we compute the probability distribution of a sentence s as follows:

$$p(s) = \prod_{i=1}^{n} P(w_i \mid w_1 ... w_{i-1}) \approx \prod_{i=1}^{n} P(w_i \mid w_{i-1}) \qquad (4)$$

However, there is an event such that $P(w_i \mid w_{i-1}) = 0$. To handle this condition, we applied Additive Smoothing to estimate its probability. The formula was experimented and slightly modified to fit our needs and defined as follows:

$$P_{Add\_Theta}(w_i \mid w_{i-1}) = \frac{\delta + Freq(w_{i-1}w_i)}{\delta|W| + \sum_{w_i} Freq(w_{i-1}w_i)} \qquad (5)$$

We define δ parameter as Freq(w_i)/|W| where |W| is an estimate number of the total words appears in N returned documents and $0 < \delta < 1$.

6 Experimental Results

With no restriction, there were 167,735 searches performed using the Yahoo! Boss Web Service API. We bootstrapped the initial core lexicons from Ho's Word List (2004) and built up to gather lexical statistics and discovered new OOV words. The corpus syllables classifications and their occurrences are shown in Figure 4.

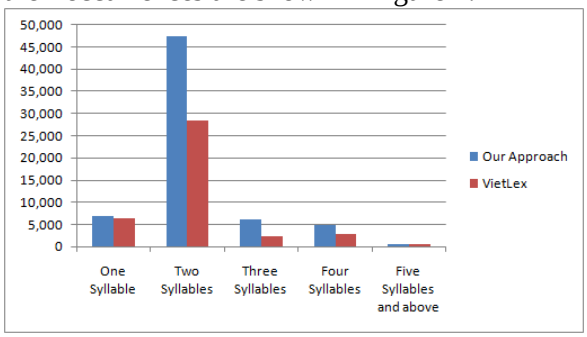

Figure 4. Syllables Types by Frequency

We compared our collected lexical data, using our approach, against VietLex (Dinh et al., 2008) and found a resembling to one, three, four, and five syllables. For the two syllables, there is a big difference: roughly about 19,000 words. This contributes to the fact that the original Ho's word list had already covered 49,583 two-syllable words to begin with. On top of it, we have included 3,000 additional new OOV words including MDW and proper names words. According to the Wiki's - Vietnamese_morphology, it estimates about 80% of the lexicon being disyllabic. In our corpus, we have 72% of disyllabic words.

1-Syllable	English Translation	Frequency
người *	person; people	571236
không *	not;negative	515704
những *	the;certain;some	446096
trong *	in;clear	439849
của *	of	356214
một *	one	330102
được *	able;possible	294234
cho *	give	220853
chúng	they, them, you	197476
cũng*	too;also	187149
tôi	I	183386
các *	every; all	176094
là *	to be; then	171868
có *	to be; to have	169001
nhưng	but	168465
và *	and	166743
với *	with	165368
thành *	to achieve; to make	158612
như	as;like	157848
phải *	must	155478

Table 3. The top 20th one-syllable words comparing with corpus of Le A. H[1] (2003)

[1] The star marker indicates the same word is co-occurred in Le's of top unigram listing.

Table 3 provides a top 20 one-syllable words obtained from our word database. The star marker indicates the same word is also co-occurred in Le's of top unigram listing.

The following disyllabic words, in Table 4, are a few of the new OOV words identified by our approach and absent from Ho's Word List (2004) .

Common Disyllabic Words	Frequency	Uncommon Disyllabic Words	Frequency
Việt Nam (Viet Nam)	206704	lan rộng (spread)	263
Người Việt (Vietnamese)	41260	ga lông (gallon)	14
Trung Quốc (China)	35345	Côn Phụng (Island)	9
Tiếng Việt (Vietnamese)	28460	nghị sỹ (congress gress-man)	22
Hoa Kỳ (America)	21262	công xôn (console)	2

Table 4. Some OOV disyllabic words

We evaluated our segmentation system against a popular Vietnamese word segmentation tool - the JVnSegmenter (Nguyen C. T, 2007): A Java-based Vietnamese Word Segmentation Tool (SVM). This tool was also a part of Dinh et al. (2008) evaluation aforementioned. With a source data provided by a neutral evaluator, and about 9600 sentences with an estimate of 100K words, we ran an experiment. The texts were input into both methods. To keep the fairness of the evaluation, the segmented output texts were sent out to a neutral assessor to analyze for results. The performance results are presented in Table 5. below.

Evaluation Areas	JVnSegmenter	Our Approach
Recall	0.814	0.821
Precision	0.883	0.897
F-Measure	0.847	0.857
OOV Rate	0.06	0.06
OOV Recall	0.921	0.951
IV Recall	0.807	0.813

Table 5. Performance Results Comparison

From the data above, the low OOV rate and high OOV recall in both systems could be explained by the nature of the testing corpus: Vietnamese novels/stories chosen by a neutral evaluator. With this type of content, the numbers of OOV words are much lesser when compared to other areas such as news, technology. Even though the results don't seem much higher than those obtained by JVnSegmenter, given the fact that JVnSegmenter used a manual trained corpus, our result is worth encouragements. Table 6 provides a few examples of the segmentation results.

Q1: tốc độ **truyền thông tin** sẽ tăng cao (Ambiguity) JVnSegmenter: [tốc độ] [**truyền thông tin**] [sẽ] [tăng] [cao] Our Approach: tốc độ | **truyền** | **thông tin** | sẽ | tăng | cao	Q2: **hàn mặc tử** là một nhà thơ nổi tiếng (Proper Name) JVnSegmenter: **[hàn mặc] [tử]** [là] [một] [nhà thơ] [nổi tiếng] Our Approach: **hàn mặc tử** | là | một | nhà thơ | nổi tiếng
Q3: một người đàn bà làm nghề **bán nước trà** ven đường (Ambiguity) JVnSegmenter: [một] [người đàn bà] [làm nghề] [**bán nước] [trà**] [ven đường] Our Approach: một | người đàn bà | làm nghề | **bán** | **nước trà** | ven đường	Q4: thủ tướng trung quốc **ôn gia bảo** (Proper name) JVnSegmenter: [thủ tướng] [trung] [quốc] **[ôn] [gia bảo]** Our Approach: thủ tướng | trung quốc | **ôn gia bảo**

Table 6. Sample outputs of the two approaches: Our approach vs. JVnSegmenter

7 Conclusion

We presented our approach to segment Vietnamese text and to build a web corpus for the function. We made use of the web document titles and their snippet text to build a scalable corpus for segmenting query text. The results so far have shown that this approach has the following benefits:

- From a practical and performance perspective, this approach does not require extended manual effort in building a corpus. The learning from the training engine, running continuously, discovers new OOV words and feeds them into a normal word segmentation process where it supplies solutions to requesters efficiently.
- The approach discovers new OOV words and disambiguates words. Additionally, we discovered new proper nouns which are not a part of any dictionaries continuously. We integrated the finding knowledge from the Vietnamese Wikipedia into our OOV words confirmation process automatically. This makes the validation of new words much easier as suppose to rely on word adjudicators manually as per O'Neil (2007). And last, the evaluation result is a better edge when comparing to a popular Vietnamese segmentation tool in all the metrics considered. This tool has a corpus trained manually.
- Frequently found OOV words identified by our process which are not available in the Vietnamese Wikipedia can be suggested to Wiki authors' communities to create content and make them available for the worldwide audiences for their benefit.

For future works, we would like to look into the possibility of applying grammatical rules in conjunction with our current statistical based system to obtain a higher identification rate. Spelling suggestion and cross-lingual search are other interesting aspects, as now words can be identified along with their lexical statistics.

Acknowledgement

Our work is credited from the works of Nguyen Bon et al. (2006), Ho Ngoc Duc (The Free Vietnamese Dictionary Project), Cam T Nguyen et al. (JVnSegmenter - 2007), O'Neil (2007), and Yahoo! Boss Web Service, which made the API available limitlessly during the course of the work, and many anonymous contributors and reviewers. A Special thank to Mr. Thuy Vu who contributed to an assessment of our approach and the JVnSegmenter.

Reference

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.

Cliff Goddard. 2005. The Languages of East and Southeast Asia (pages 70-71)

Dinh Dien, Vu Thuy. 2006. A Maximum Entropy Approach for Vietnamese Word Segmentation. In Proceedings of the 4th IEEE International Confe-

rence on Computer Science- Research, Innovation and Vision of the Future 2006, HCM City, Vietnam, pp.247–252.

Dinh Quan Thang, et al, 2008. Word Segmentation of Vietnamese Texts: a comparison of approaches. LREC : 2008

Ghani, R., Jones, R., Mladenic, D. 2001. Using the Web to create minority language corpora'. Proceedings of the 10th International Conference on Information and Knowledge Management

Ho Ngoc Duc, 2004: Vietnamese word list: Ho Ngoc Duc's word list – http://www.informatik.uni-leipzig.de/~duc/software/misc/wordlist.html

John O'Neil. 2007. Large Corpus Construction for Chinese Lexical Development, Government Users Conference: http://www.basistech.com/knowledge-center/unicode/emerson-iuc29.pdf

Le Thanh Ha, Huynh Quyet Thang, Luong Chi Mai. 2005. A Primary Study on Summarization of Documents in Vietnamese. The First International Congress of the International Federation for Systems Research, Japan.

L. H. Phuong and H. T. Vinh, 2008, Maximum Entropy Approach to Sentence Boundary Detection of Vietnamese Texts, IEEE International Conference on Research, Innovation and Vision for the Future RIVF 2008, Vietnam.

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

Marco Baroni, Motoko Ueyama. 2006. Building general and special-purpose corpora by Web Crawling. Proceedings of the 13th NIJL International Symposium, Language Corpora: Their Compilation and Application. 31-40.

Ngo. N. Binh, B. H. Tran. 2001. Vietnamese Language Learning Framework – Part One: s Linguistic.

Nguyen D. 2008. Query preprocessing: improving web search through a Vietnamese word tokenization approach. SIGIR 2008: 765-766.

Stanley F. Chen, J. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Center Research in Computing Technology, Harvard University, TR-10-98

Thanh Bon Nguyen, Thi Minh Huyen Nguyen, Laurent Romary, Xuan Luong Vu. 2006. A lexicon for Vietnamese language processing. Language Resources and Evaluation. Springer Netherlands

V-B. Le, B. Bigi, L. Besacier, E. Castelli, 2003. Using the Web for fast language model construction in minority languages", Eurospeech'03, Geneva, Switzerland, September 2003

Wirote Aroonmanakun. 2002. Collocation and Thai Word Segmentation, Proceedings of SNLP-Oriental COCOSDA 2002

Vietnamese morphology: From Wikipedia: http://en.wikipedia.org/wiki/Vietnamese_morphology

Yahoo! Boss Web Service API http://developer.yahoo.com/search/boss