VNLP: An Open Source Framework for Vietnamese Natural Language Processing

Ngoc Minh Le
University of Trento
minh.lengoc@unitn.it

Bich Ngoc Do
University of Groningen
d.ngoc@student.rug.nl

Vi Duong Nguyen
ePi Technologies
duong@epi.com.vn

Thi Dam Nguyen
ePi Technologies
dam@epi.com.vn

ABSTRACT

Natural Language Processing (NLP) for Vietnamese has been researched for more than a decade but still lacks of an open-source NLP pipeline. As the result, researchers have to spend a lot of time on various fundamental tasks before working on the task of interest. Besides, the circumstance holds back text processing technology in Vietnam because an application costs much more money and time to reach a deliverable state. This work is an attempt to solve this issue. By incorporating available open-source software packages and implementing new ones, we have created an open-source, production-ready solution for Vietnamese text processing. Via three experiments, we demonstrated its effectiveness and efficiency. The software has helped us to develop our solution for Vietnamese sentiment analysis and online reputation management and we hope that it will also facilitate research in Vietnamese NLP.

Categories and Subject Descriptors

I.2.7 [**Natural language processing**]: Language parsing and understanding; D.2.13 [**Reusable software**]: Reusable libraries

General Terms

Algorithms

Keywords

word segmentation, pos tagging, named-entity recognition, dependency parsing, coreference resolution, NLP pipeline

1. INTRODUCTION

NLP is an inherently vast and intricate problem and sometimes considered "AI-complete". To cope with its complexity, researchers have divided it into various sequential subproblems such as word segmentation, part-of-speech tagging (POS tagging), syntactic parsing, named-entity recognition (NER) and coreference resolution, among others. By adopting this approach, the NLP enterprise is carried out by developing solutions for each subproblem independently, usually by independent groups, and then integrating them into a system.

For various reasons, sometimes non-technical, most researches in NLP have been concerned with English while much less attention is paid to some languages such as Vietnamese. As a consequence, there are few software packages developed for fundamental tasks such as word segmentation, POS tagging and syntactic parsing; no available solution for some other tasks such as NER and coreference resolution and no attempt has been made to incorporate available pieces of software into a framework. [1] [2]

This circumstance has restricted Vietnamese NLP researches. At universities, students work on dead simple and repeated projects because more interesting ideas are not realizable. Researchers waste their time on reimplementing solutions for fundamental subtasks and integrating them into a pipeline to be able to attack the subtask of interest. Many beneficial and profitable applications becomes out-of-reach for independent software developers while companies have to invest more money and time to deliver their products.

We have developed VNLP as an answer to these problems. VNLP is a framework for Vietnamese NLP consisting of command line tools, plugins for GATE (Cunningham et al., 2002) and GATE applications. GATE serves as a feature-rich integrated development environment and an efficient and comprehensive Java library. Building upon GATE, we can make use of many high-quality components and ensure that our framework is extensible.

VNLP can solve subtask individually or jointly, work in GUI or command line (with some limitations), stand-alone or integrated into another system, solve ad-hoc problems or massive text processing. The framework is used to improve our sentiment analysis algorithms in a commercial online reputation management system. Furthermore, we publicly

SoICT 2013 December 05-06, 2013 Da Nang, Vietnam
Copyright 2013 ACM 978-1-4503-2454-0 ...$10.00.

[1] However, have a look at JVnTextPro for a bundle of command line, open-source tools consisting of sentence segmentation, "sentence tokenization", word segmentation and POS tagging

[2] In the current writing, we concern software packages that are readily available on the Internet or purchasable from a vendor. We are aware of researches such as Nguyen and Cao (2008) and the VLSP national project but could not get their products.

distribute[3] the framework under GPL license in the hope of providing the Vietnamese NLP community with:

- An out-of-the-box, production-ready solution.
- A fundamental and common framework for research.
- A set of basic algorithms that may serve as a baseline for future evaluations or contests.

Although some algorithms employed in the framework were state-of-the-art when we started to work on VNLP, i.e. around 2011-2012, we have made no attempt to update it with latest advancements in the field. One obvious reason is that there are many new algorithms come out each year and few of them is open-sourced. Moreover, some algorithms may be quite complicated or require a huge amount of training data to work properly, which is not suitable for a generic framework for both novices and experts. In spite of accuracy, we focus on the simplicity, speed and stability.

The rest of the paper is organized as follows. We first briefly describe the various subtasks implemented in VNLP in Section 2 and then show how to incorporate them into a complete pipeline in Section 3. Section 4 assesses the performance of the framework by three experiments. Finally, in Section 5, we conclude by discussing the contribution of this work and future developments.

2. IMPLEMENTING SUBTASKS IN VIETNAMESE NLP

2.1 Word segmentation

Word segmentation is the first and fundamental step in Vietnamese text processing. The results of this step impose a hard constraint on all the rest of the framework in the sense that if it mistakes a word, there is no chance for a subsequent step to get right. As a result, word segmentation has received quite much attention in Vietnamese NLP community.

Among other works, we chose vnTokenizer (Le et al., 2008) to include in the framework. vnTokenizer is a hybrid maximal matching and graph-based word segmenter with word accuracy up to 97%. It has been used to tokenize 2 million syllables in Vietnamese Treebank project and recently to tokenize 94 million words in VietnameseWAC corpus (Kilgarriff and Le-Hong, 2012).

Recently, vnTokenizer was criticized for its large memory footprint and low speed.[4] However, less than 20MB of RAM is negligible for modern computers and we have significantly improved its speed by three modifications. Firstly, instead of reading XML-encoded data via an intermediate document object model, we used SAX, an event-driven API, to read it directly. Secondly, to tokenize a document into basic elements (syllables, numbers and punctuations, for example) we do not use regular expressions but a LL(*) parser generated by ANTLR. Thirdly, we use an automaton with default transitions (see Figure 1) to reduce word identification time of an ambiguous phrase from $O(w_{\max}s^2)$ to $O(w_{\max}s)$ where s is the number of syllables in the phrase and $w_{\max}$ is the maximal length in character of a word. The impact of those improvements will be investigated in Section 4.1.

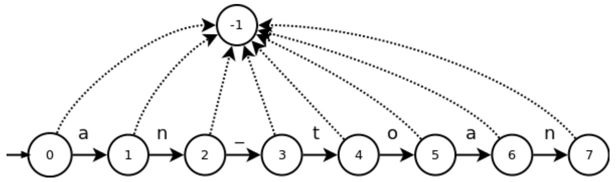


Figure 1: An automaton for "an toàn" (safe) showing default transitions (dotted lines). `-1` is a dead-end state, i.e. it has no out-going transition, therefore when the automaton reaches state `-1`, we can safely stop looking for words.

While the first and second improvements are quite trivial, the last one deserves more investigation. The algorithm adopts a minimal deterministic finite automaton (MDFA) to represent the vocabulary. For every ambiguous phrase, the MDFA is used to find out all possible words. Originally, for each of s syllables, the algorithm concatenates it with every succeeding syllables and search for the resulting sequence whose length is $O(w_{\max}s)$. Therefore it costs $O(w_{\max}s^2)$ for each phrase. In our implementation, we don't concatenate syllables but validate characters one by one against the MDFA. As soon as we reach the state `-1`, which means the current sequence can prefix no word in our vocabulary, the search stops. Because no prefix can be longer than $w_{\max}$, the time to analyze a phrase is $O(w_{\max}s)$.

2.2 Part-of-speech tagging

Part-of-speech (POS) tagging is a shallow parsing subtask useful for many problems including, but not limited to, full parsing, named-entity recognition and information extraction.

To the best of our knowledge, there are currently two open-source solutions for Vietnamese POS tagging: JVnTagger and vnTagger (Le-Hong et al., 2010b). Additionally, vnQTag[5] (Nguyen et al., 2003) is a freeware that appeared around 10 years ago. vnTagger is the latest and, at the time of writing, still among state-of-the-art taggers for Vietnamese. Therefore we chose vnTagger to include in our framework.

2.3 Syntactic parsing

Syntactic parsing has been a central challenge in NLP for decades. For Vietnamese, this problem is also the topic of many researches. Some attempted directions are tree-adjoining grammar (Le-Hong et al., 2006, 2010a, 2012), head-driven phrase structure grammar (HSPG; Do and Le, 2008), probabilistic context-free grammar (PCFG; Hoang et al.), lexicalized PCFG (Le et al., 2009), lexical functional grammar (Le and Phan, 2009) and link grammar (Le and Nguyen, 2012). Apart from an immature link grammar parser, none of the researches mentioned above has resulted in a deliverable (either free or commercial) software.[6]

[3] `https://bitbucket.org/epilab/vnlp`

[4] Tuan Anh Luu, Yamamoto Kazuhide, Pointwise for Vietnamese word segmentation, vietlex.com, accessed on July 22, 2013

[5] `http://raweb.inria.fr/rapportsactivite/RA2003/led/id2642150.html`

[6] However, VLSP project has a HPSG demonstration at `http://vlsp.vietlp.org/` that allows us to parse a sentence via its web interface and vnLTAG provides a

For our syntactic parsing component, we employed Maltparser (Nivre and Hall, 2005). Maltparser is an open source parser generator for dependency parsing. It has been around for more than six years and used to develop state-of-the-art parsers for a number of languages. To the best of our knowledge, we are the first to explore the potential of Maltparser and transition-based dependency parsing for Vietnamese.

In (Nguyen and Nguyen, 2012), we have established a procedure to generate dependencies from VietTreeBank. Training on this corpus with the default feature model of Maltparser, we have successfully generated a dependency parser for Vietnamese with an accuracy of around 70%.

2.4 Named-Entity Recognition

NER for Vietnamese has been tackled by several methods. Supervised (Nguyen et al., 2005) and semi-supervised (Pham et al., 2012; Sam et al., 2011) approaches have been attempted. Used algorithms include rule-based (Vo and Ock, 2012), support vector machine (SVM; Tran et al., 2007) and conditional random field (Nguyen et al., 2005; Pham et al., 2012; Sam et al., 2011). Some other researches focus on corpus building (Thao et al., 2007; Nguyen et al., 2010). Unfortunately, none of those research has resulted in a free/open-source software.

For simplicity, we have developed a rule-based NER system. Analogous to ANNIE (Cunningham et al., 2002), an information extraction system shipped with GATE, our system is broken into two parts: a word searching component called gazetteer in GATE's terminology and a pattern matching component called transducer. The gazetteer matches a document against various lists of organizations (schools, firms, state agencies, etc.), locations (roads, wards, districts, cities, countries, regions, etc.), persons (celebrities, family names, popular middle and first names), products (extracted from famous online shops) and some auxiliary lexical units (for example: units, job names, titles, directions). Because GATE gazetteers disregard Vietnamese word boundary, we need to perform an additional step to remove annotations crossing word boundary. The transducer then match sequences of annotations against a set of patterns to decide the appropriate annotations for entities in the document. A transducer rule expressed in JAPE (Java Annotation Patterns Engine) looks like:

```
Rule: ProperName1
Priority: 15
(
   ({Lookup.majorType == per_family-name,
     Token.orth == upperInitial})
   (
      ({Lookup.majorType == per_family-name,
        Token.orth == upperInitial})?
      ({Lookup.majorType == per_middle-name,
        Token.orth == upperInitial})[1, 3]
   )
   ({Lookup.majorType == per_first-name,
     Token.orth == upperInitial})
):name
-->
:name.Person = {rule = "ProperName1"}
```

The snippet matches Vietnamese names of the form: family name + an optional family name + middle name + first name. Should a sequence of annotations matches this pattern, the transducer will put an annotation of type `Person` with a feature `rule=ProperName1` to colocate with it.

2.5 Coreference resolution

Coreference resolution (CR) is an essential subtask after NER that explores and links entities in a document together for better understanding. A good writer never repeats a name from time to time in a passage but makes use of pronouns, abbreviations and alternative names to refer to a mentioned entity. Without the help of CR, we will lose the majority of information involving entities.

CR has received little attention of Vietnamese NLP community. The only two researches that we are aware of are Le et al. (2011) which used SVM and Sam et al. (2011) which used a rule-based CR system as a means to bootstrap NER.

We approach this problem by heuristic rules. The system consists of two components: an orthographical matcher (orthomatcher) and a coreferencer. Based on ANNIE's English orthomatcher, we built our orthomatcher with 17 rules such as exact name matching, the matching of a short name and the first part of a long name, acronyms. The coreferencer performs pronominal coreferencing and integrates everything into coreference lists. The best antecedent of a pronoun is chosen to best match the pronoun's features (e.g. gender, number, title) and appear closest to it.

3. VIETNAMESE NLP PIPELINE

The major contribution of this work is a complete solution for Vietnamese NLP and it will not be possible if we do not put all the above subtasks together to form a single pipeline. Hence, VNLP is distributed with two GATE applications for parsing (`parsing.gapp`) and NER (`ner.gapp`). The applications can be run and adjusted manually or embedded in a bigger system and called programmatically.

Figure 2 shows the parsing application. "Document Reset PR" is a processing resource that reverses any modification made by previous runs. The application first tokenizes a document, marks sentences using ANNIE's sentence splitter, annotates each token with a POS tag and then adds dependencies to them.

The NER application is shown in Figure 3. A POS tagging step after sentence detection is optional and sometimes reduces the performance of the pipeline. The application does not require a sentence to be parsed therefore Maltparser is omitted. NER gazetteer, lookup correction and NER transducer are used to mark names. Then, it finishes with the orthomatcher and the coreferencer.

If one needs to parse as well as annotate named-entities in a corpus, she can easily do so by appending Maltparser processing resource to the NER application.

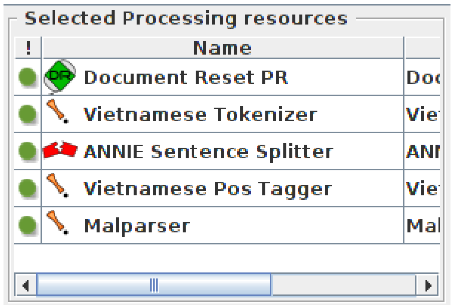
4. EXPERIMENTS

This section describes the assessment of components in VNLP. Because vnTokenizer and vnTagger have been evaluated and used for years, we will not repeat their experiments here.

4.1 Word segmenter speed experiment

To assess our modified version of vnTokenizer, we measure

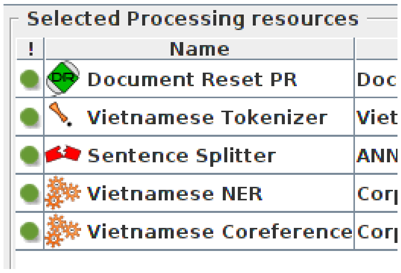
small grammar for testing at `http://wiki.loria.fr/wiki/VnLTAG`.



Selected Processing resources

!	Name	
●	DR Document Reset PR	Doc
●	Vietnamese Tokenizer	Vie
●	ANNIE Sentence Splitter	ANI
●	Vietnamese Pos Tagger	Vie
●	Malparser	Mal

Figure 2: VNLP for syntactic parsing (`parsing.gapp`) as it loaded in GATE



Selected Processing resources

!	Name	
●	DR Document Reset PR	Doc
●	Vietnamese Tokenizer	Viet
●	Sentence Splitter	ANN
●	Vietnamese NER	Cor
●	Vietnamese Coreference	Cor

Figure 3: VNLP for named-entity recognition (`ner.gapp`) as it loaded in GATE

the running time of four tokenizers against the 2-million-syllable corpus from VLSP project. vnTokenizer is a graph-based word segmenter that chooses the most likely way to segment a sequence of syllables according to a language model. vnTokenizer 4.1.1 only computes probabilities according to an unigram model. Our modified version lets the user choose between unigram and bigram models while the later is slightly more accurate given appropriate parameters. Besides, we has tuned vnTokenizer for speed as described in Section 2.1. Dong Du is a pointwise segmenter by Tuan Anh Luu appeared in late 2012.

Table 1 shows that our implementation is significantly faster than both the original vnTokenizer and Dong Du with a speed of about 0.3 million syllables per second or about 440 documents per second.

4.2 Dependency parsing experiment

Our parser is evaluated against the converted version of VietTreeBank (into dependency grammar). We used 2-planar algorithm (Gómez-Rodríguez and Nivre, 2010) which is among state-of-the-art transition-based parser and runs in linear time. The algorithm produces 2-planar dependency trees that have been proved to cover 99% of current treebanks in 8 languages. Its SVM model is trained by LIBLINEAR library for speed and accuracy. We divided the converted treebank into 70% for training and 30% for testing. The experiment is repeated 5 times.

Segmenter	Time
vnTokenizer (unigram)	234
Dong Du	44
Modified vnTokenizer (unigram)	6
Modified vnTokenizer (bigram)	11

Table 1: Running time (in seconds) of some segmenters against 2 million syllables

We computed two scores: labeled attachment score (LAS) and unlabeled attachment score (UAS). LAS is the accuracy of dependency arcs, taking into account each dependency's two nodes, direction and label while UAS concerns only the first three aspects but leave out the arc label.

As the result, our parser achieved **73% in LAS** and **69% in UAS**. This performance is lower than reported results of other parsers for Vietnamese but still helpful for many applications. Besides, as this is among the first researches in dependency parsing for Vietnamese and we did not try Vietnamese-specific algorithms, many things can still be done along this direction.

4.3 Named-entity recognition and coreference resolution experiment

Finally, we investigate the performance of NER and CR components. Because there is no NER annotated corpus for Vietnamese yet, we first set out to build a corpus.[7]

We collected hundreds of reviewing articles about mobile phones from `sohoa.vnexpress.net` and `vnreview.vn`. Two teams of students from Hanoi University of Science and Technology are hired to annotate the articles. Each team works on the same set of articles independently. Each annotator knows only about the articles he or she is given but not what articles others are working on. Any discussion about an article, either its title, author or content, is banned. Even the fact that they are divided into two teams is known only by researchers.

Since the beginning, we established a guideline for annotators. During the annotating process, annotators kept close contact with researchers to communicate difficulties and clarify the requirements. We also compared annotations between the two teams every week and sent corrected annotations to them. Vagueness and inconsistency in the guideline which was not apparent at first was pointed out and repaired. To ensure that every annotator has time to annotate carefully and learn from the errors she makes, we restrict the number of documents an annotator can work on per week. The salary for an annotator is computed based on how much and how accurate she does and be visible to her at any time via a web interface. We also rank annotators by accuracy and show it to every one in the team. The purpose of this procedure is to ensure that the quality of the resulting corpus is as high as possible.

Due to the shortage of time and money budget, only about 140 documents were annotated and 104 of them were annotated by two annotators. We measure the inter-annotator agreement of those documents by the traditional Precision, Recall and F1 scores as in Table 2.

For practical purposes, we pay attention to product names in the text while disregarding date/time expression as in traditional NER settings. Table 3 shows the evaluation result of the NER pipeline. It should be noted that annotations

[7] The corpus is downloadable from `https://bitbucket.org/epilab/vnlp/downloads/sentiment-analysis.zip`, it is also annotated with opinions for opinion mining/sentiment analysis.

Type	F1
Location	0.97
Organization	0.89
Person	0.88
Product	0.89
All	0.89

Table 2: Inter-annotator agreement of 104 documents that are annotated by two annotators.

Type	Precision	Recall	F1
Location	0.17	0.79	0.28
Organization	0.68	0.82	0.74
Person	0.71	0.91	0.80
Product	0.63	0.52	0.57
All	0.57	0.59	0.58

Table 3: The performance of NER pipeline on our 140-document corpus.

evaluated in this experiment is the result of a full pipeline execution including tokenization. The precision of location annotations is low because of highly ambiguous rules such as prepositions (for example, "on" + proper noun) and postal addresses (for example, a number + proper noun). In general, the performance is modest compared to the alleged results of other researches. However, we hope that it will help people to attack more problems in Vietnamese NLP.

5. CONCLUSION AND FUTURE WORK

In this paper, we have reported the first open-source framework for Vietnamese NLP. To build its components, we studied the current state of NLP for Vietnamese in various subtasks: word segmentation, part-of-speech tagging, syntactic parsing, named-entity recognition and coreference resolution. Based on this knowledge, we chose to reuse two open-source software, vnTokenizer and vnTagger, adopt Maltparser for Vietnamese parsing and implement our solution for other tasks. The resulting framework was evaluated in accuracy and speed against VietTreeBank and our newly built corpus.

Our main contributions are to provide two ready-to-use pipelines for Vietnamese text processing and an open-source dependency parser for Vietnamese. Besides, an annotated corpus for NER and opinion mining/sentiment analysis was created and publicly released.

Although the framework has proved useful in our applications, the performance of some components is still modest and needs improvement. We also want to extend the framework to cover more tasks such as diacritics restoration, chunking and information extraction.

References

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

Ba Lam Do and Thanh Huong Le. Implementing A Vietnamese Syntactic Parser Using HPSG. In *The International Conference on Asian Language Processing (IALP)*, 2008.

Carlos Gómez-Rodríguez and Joakim Nivre. A transition-based parser for 2-planar dependency structures. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1492–1501. Association for Computational Linguistics, 2010.

Anh Viet Hoang, Thi Phuong Thu Dinh, and Quyet Thang Huynh. Vietnamese parsing applying the PCFG model.

Adam Kilgarriff and Phuong Le-Hong. Vietnamese Word Sketches. In *Proceedings of the First International Workshop on Vietnamese Language and Speech Processing*, pages 1–4, 2012.

Anh-Cuong Le, Phuong-Thai Nguyen, Hoai-Thu Vuong, Minh-Thu Pham, and Tu-Bao Ho. An Experimental Study on Lexicalized Statistical Parsing for Vietnamese. In *2009 International Conference on Knowledge and Systems Engineering*, pages 162–167. IEEE, October 2009. ISBN 978-1-4244-5086-2. doi: 10.1109/KSE.2009.41.

Duc-Trong Le, Mai-Vu Tran, Tri-Thanh Nguyen, and Quang-Thuy Ha. Co-reference Resolution in Vietnamese Documents Based on Support Vector Machines. In *2011 International Conference on Asian Language Processing*, pages 89–92. IEEE, November 2011. ISBN 978-1-4577-1733-8. doi: 10.1109/IALP.2011.63.

Hong-Phuong Le, Minh-Huyen Thi Nguyen, Azim Roussanaly, and Tuong-Vinh Ho. A Hybrid Approach to Word Segmentation of Vietnamese Texts. *Language and Automata Theory and Applications*, page 240, 2008.

Manh Hai Le and Thi Tuoi Phan. Vietnamese Lexical Functional Grammar. In *2009 International Conference on Knowledge and Systems Engineering*, pages 168–171. IEEE, October 2009. ISBN 978-1-4244-5086-2. doi: 10.1109/KSE.2009.45.

Ngoc Minh Le and Thanh Huong Nguyen. Application of Link Grammar Formalism in Vietnamese-English Translation. *Journal of Information and Communication Technology*, 8(28):44–56, 2012. URL `https://bitbucket.org/ngocminh/lienkate/downloads/ADJ_Translation_2.doc`.

Phuong Le-Hong, Minh-Huyen Thi Nguyen, Laurent Romary, and Azim Roussanaly. A Lexicalized Tree-Adjoining Grammar for Vietnamese. In *International Conference on Language Resources and Evaluation - LREC 2006*, 2006.

Phuong Le-Hong, Thi Minh Huyen Nguyen, Phuong Thai Nguyen, and Azim Roussanaly. Automated extraction of tree adjoining grammars from a treebank for Vietnamese. *Journal of Computer Science and Cybernetics, Vietnamese Academy of Science and Technology, Vietnam*, 26(2):153–171, 2010a.

Phuong Le-Hong, Azim Roussanaly, Thi Minh Huyen Nguyen, and Mathias Rossignol. An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts. In *Traitement Automatique des Langues Naturelles-TALN 2010*, 2010b.

Phuong Le-Hong, Thi Minh Huyen Nguyen, and Azim Roussanaly. Vietnamese Parsing with an Automatically Extracted Tree-Adjoining Grammar. In *2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future*, pages 1–6. IEEE, February 2012.

Cam Tu Nguyen, Thi Oanh Tran, Xuan Hieu Phan, and Quang Thuy Ha. Named entity recognition in vietnamese free-text and web documents using conditional random fields. In *The 8th Conference on Some selection problems of Information Technology and Telecommunication*, 2005.

DatBa Nguyen, SonHuu Hoang, SonBao Pham, and Thai-Phuong Nguyen. Named entity recognition for Vietnamese. In *Intelligent Information and Database Systems*, pages 205–214. 2010.

Thi Minh Huyen Nguyen, Xuan Luong Vu, and Phuong Le-Hong. A case study of the probabilistic tagger QTAG for Tagging Vietnamese Texts. In *Proceedings of the 1st National Conference ICT RDA*, 2003.

Truc-Vien T. Nguyen and Tru H. Cao. VN-KIM IE: Automatic Extraction of Vietnamese Named-Entities on the Web. *New Generation Computing*, 25(3):277–292, March 2008. ISSN 0288-3635.

Vi Duong Nguyen and Thi Dam Nguyen. Depedency grammar for Vietnamese. 2012. URL `https://bitbucket.org/epilab/vnlp/downloads/DependencyGrammarForVNese.doc`.

Joakim Nivre and Johan Hall. Maltparser: A language-independent system for data-driven dependency parsing. In *Proc. of the Fourth Workshop on Treebanks and Linguistic Theories*, pages 13–95, 2005.

Thi-Ngan Pham, Le Minh Nguyen, and Quang-Thuy Ha. Named Entity Recognition for Vietnamese documents using semi-supervised learning method of CRFs with Generalized Expectation Criteria. In *Asian Language Processing (IALP), 2012 International Conference on*, pages 85 – 88, 2012.

Rathany Chan Sam, Huong Thanh Le, Thuy Thanh Nguyen, and Thien Huu Nguyen. Combining proper name-coreference with conditional random fields for semi-supervised named entity recognition in Vietnamese text. In *Advances in Knowledge Discovery and Data Mining*, pages 512–524. Springer, 2011.

Pham T. X. Thao, T. Q. Tri, Ai Kawazoe, Dien Dinh, and Nigel Collier. Construction of Vietnamese corpora for named entity recognition. pages 719–724, May 2007.

Tri Q. Tran, Thao T. X. Pham, Hung Q. Ngo, Dien Dinh, and Nigel Collier. Named entity recognition in Vietnamese documents. *Progress in Informatics*, 4:5–13, 2007.

Duc-Thuan Vo and Cheol-Young Ock. A hybrid approach of pattern extraction and semi-supervised learning for Vietnamese named entity recognition. *Computational Collective Intelligence. Technologies and Applications*, 2012.