JAIST Repository
https://dspace.jaist.ac.jp/

Title	Exploring Effective Features for Learning Vietnamese Word Sense Disambiguation Classifiers
Author(s)	グエン，ハイミン
Citation	
Issue Date	2010-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/9149
Rights	
Description	Supervisor:Associate Professor Shirai Kiyoaki，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Exploring Effective Features for Learning Vietnamese Word Sense Disambiguation Classifiers

By Nguyen Hai Minh

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Kiyoaki Shirai

September, 2010

Exploring Effective Features for Learning Vietnamese Word Sense Disambiguation Classifiers

By Nguyen Hai Minh (0810204)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Kiyoaki Shirai

and approved by
Associate Professor Kiyoaki Shirai
Professor Akira Shimazu
Associate Professor Yoshimasa Tsuruoka

August, 2010 (Submitted)

Copyright © 2010 by Nguyen Hai Minh

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We first introduce Word Sense Disambiguation (WSD hereafter) problem and its important role in natural language processing. In many languages, such as English, numerous researches have been devoted in WSD. However, there has been no research on Vietnamese WSD. Therefore, our study aims to investigate a WSD method for Vietnamese, specifically to explore effective features for learning Vietnamese WSD classifiers.

1.1 Word Sense Disambiguation Overview

Making a computer which can understand human being language is a big dream of many researchers in computer science. However, the language that human can easily learn is the most difficult thing for a computer to master. A human can naturally understand the word *'bank'* in the sentence '*I enter the bank*' but a computer is confused whether that *'bank'* is a financial institution or an edge of a river. Lexical ambiguity is a fundamental characteristic of languages. 121 most frequent English nouns which occur about one in five words in real text, have on average 7.8 meanings each [**?**](page 1). This leads to the task of automatic disambiguation of senses, which has been noticed since the early days of applying computer to natural language processing in 1950s. Once this problem is solved, many systems that require text understanding, such as machine translation, information retrieval, question answering, semantic analysis, text mining, speech processing, etc. can be improved significantly [10].

The task of determining senses of words in a certain context is called Word Sense Disambiguation in the field of computational linguistics. Word senses can be understood as word meanings in an ordinary dictionary or word translations in a machine translation. Usually this problem can be seen as a task of classification where word senses are classes, the context provides the evidence, and each occurrence of a word is assigned to one or more of its possible classes based on the evidence.

In English, WSD has been researched for more than half century. The first experiment by Kaplan (1950) proved that only one or two words in both side of the ambiguous word can be evidence to disambiguate that word [11]. Later, more useful information from context are discovered by numerous work in WSD. Yarowsky introduced simple set of features (context around the ambiguous words) in accent restoration task [?]. This leads to many other improved set of features such as syntactic dependencies [?, 6, ?], or cross language evidence [8]. Besides the approaches utilizing the evidence provided by surrounding context of the ambiguous word, there are many other researches take advantages of knowledge bases without using any corpus evidence, such as approaches using dictionaries, thesauri, and lexical knowledge bases [14, 2, 1]. According to the knowledge sources used in sense distinguishing, methods in WSD are classified as knowledge-based, unsupervised corpus-based, supervised corpus-based, and combinations of them [?]. Among them, approach to supervised learning is the hot topic since it is one of the most successful approaches in the last fifteen years in WSD. However, the biggest problem of supervised learning methods is the knowledge acquisition bottleneck, which exposes challenges to the supervised learning approach for WSD.

1.2 Goal of Thesis

Vietnamese is one of languages including many highly ambiguous words. For example, the word 'biển' in Vietnamese can have different meanings: the sea, a sign-board, a large group of people. Hence, WSD is also an important task in Vietnamese language processing. However, among 970 papers in the ACL Anthology mention the term "*word sense disambiguation*", there is no research on Vietnamese WSD[1]. Therefore, our research is the first attempt to establish a WSD method on Vietnamese. Especially, this study aims to find effective features for training WSD classifiers in order to distinguish ambiguous senses of words in Vietnamese sentences. Another goal is to explore applicability of 'pseudoword' method for Vietnamese WSD. Pseudoword method, which will be introduced in Section 2.3, is s well-known technique widely used to develop WSD systems when no sense tagged corpus is available. This thesis will empirically evaluate validity of it for Vietnamese words.

The thesis is organized as follow. In Chapter 2, some previous approaches on WSD methods and the knowledge sources for WSD are introduced. Our method is described in Chapter 3, which was empirically deployed to investigate the effectiveness of features in Vietnamese WSD. Chapter 4 explains three WSD tasks we designed to explore effective features and evaluate pseudoword technique. Chapter 5 shows the experiment results and some discussion. Finally, a conclusion and future work is presented in Chapter 6.

[1] Statistics were collected in February 2010 on `http://aclweb.org/anthology/`, a digital archive of research papers in computational linguistics from 1979 to present.

Chapter 2

Background

In this chapter, we study some background of previous approaches on WSD, especially the supervised corpus-based methods and knowledge sources for WSD. Different sources of linguistic knowledge have been employed by WSD systems, such as part of speech, morphology, collocations, subcategorization, and frequency of senses, semantic word associations, selectional preferences, semantic roles, domain, topical word associations, and pragmatics. However, in order to be applied, they need to be coded as features. These features are required to be extracted from lexical resources, such as corpora, machine readable dictionaries. Sense tagged corpora used in WSD methods are far more useful for WSD than untagged corpora since it is easy to examine behavior of words in a particular sense. However, the main disadvantage of this kind of corpora is that it is extremely time-consuming to produce. A technique called pseudowords is introduced to deal with the bottleneck problem in supervised learning methods. The chapter is organized as follow,

2.1 Features for WSD Algorithms
2.2 Supervised Corpus based Method for WSD
2.3 Pseudowords Technique for WSD
2.4 Vietnamese WSD

2.1 Features for WSD Algorithms

Firstly, the features which have been applied in English WSD systems are presented. These features are extracted from corpora or machine readable dictionaries. We also analysis the potential of using these features in Vietnamese WSD system.

2.1.1 Target Word Specific Features

The most obvious way to identify sense of a target word[2] is to base on the information of the word itself, such as its morphology, part-of-speech and sense distribution.

The morphological form of a word can be used to clarify its senses. For example, the noun '*tin*' has two senses, '*small metal container*' and '*metal*'. The second sense

[2] 'Target word' refers to the ambiguous word or the word being disambiguated.

is uncountable noun, so when the noun '*tins*' appears in a sentence, it must be the plural noun of the first sense. This type of feature is effective in languages that have morphologies such as English and Basque. However, it cannot be applied in Vietnamese because Vietnamese is an isolate language in which words do not change forms.

Part-of-speech (POS) can be used to determine the grammatical category for each sense. For example, the word '*tide*' has two major senses, each one belongs to a category (noun and verb) [?]. This feature is easily identified using many available POS taggers. However, POS feature is only useful for distinguishing senses in different grammatical category, which is not the case of homorgraphs. Fox example, two senses of the word '*bank*' are both nouns, and knowledge of the part of speech in context will not provide any indication of which sense the word '*bank*' is used. We do not apply this kind of feature since we only consider the ambiguous words in the same category.

Sense distribution is also an effective feature for WSD, since most of the ambiguous words have a dominant sense and several other less frequent senses. Knowledge of the prior distribution of senses is useful information for WSD. For example, one in four senses of the word '*people*' appears 90% in the Semcor sense-tagged corpus [?](page 221). In our research, sense distribution is used as a baseline measurement of the proposed WSD system.

2.1.2 Local Features

Local Patterns around the target word

In reality, when an ambiguous word is given to a human, what he/she does is to extend the surrounding context of that ambiguous word until there are enough information to indicate the sense of that word. For the same purpose, local patterns around the target word aim to capture the important context for sense disambiguation. Patterns around the target word vary in terms of their extent and fillers, such as: n-grams around the target word, n-th word to the right or left of the target word, their POS tags, or a mixture of them. This kind of feature is the most easily extracted from a tagged corpus and is most commonly used with supervised approaches to WSD. In this research, we investigate the effectiveness of some local patterns around the target word, such as POS of the words around the target word, 2-grams, 3-grams and 4-grams around the target word.

Subcategorization

Subcategorization information can be a useful knowledge source in English, in which verbs can be disambiguated according to their behaviors. For example, the verb '*to grow*' is intransitive when it has the meaning '*become bigger*' *(She has grown up)* but transitive in all other meanings *(My mother grows this plant)*. Martinez et al. [?] used Minipar to derive subcategorization information for verbs from tagged corpora and gave a result of 86% precision. Although this information is useful, it requires a subcategorization dictionary which is not available for Vietnamese now.

Syntactic Dependencies

This type of feature encodes the associations between words in sentences with respect to various syntactic dependency relationships. For example, the direct object *'my mother'* in the sentence *'I miss my mother'* indicates that the verb *'miss'* is used with *'feel or suffer from the lack of'* [?]. The dependencies of a particular word sense can be extracted from a corpus which is parsed and tagged with word senses. In our research, we also investigate some important syntactic dependencies for Vietnamese WSD.

2.1.3 Global Features

This kind of feature uses wider context information around the target word.

Bag-of-Words

The context information around the target word is simple a list of single words and their frequencies in a certain window around the target word. In English, this kind of feature can be extracted easily from a raw text. However, since Vietnamese words are not separated by blanks, we can only extract this feature on a word segmented text.

Domain of texts

This feature encodes knowledge of the domain of the text, or the association between words in text. For example, if the word *'bat'* is found in a text about animal, then its sense should not be an equipment using in sports. If the domain information of the text is not explicit, the association between words can be used in the same manner, for example, when *'racket'* and *'court'* co-occur in the text, they can disambiguate each other without the need of a domain label. However, association between word senses and domains is typically extracted from dictionary definitions, which are not available for Vietnamese.

2.2 Supervised Corpus-Based Methods for WSD

Recently, machine learning techniques have been applied to a large variety natural language processing tasks under the name of "corpus-based", "statistical" or "empirical" methods [?]. They are applied for morphological and syntactic analysis [5], semantic interpretation [?], information extraction [3], machine translation [12]. These approaches usually decompose the complex problems into simple classifications. Regarding automatic WSD, since WSD is a task to determine the sense of a target word based on its surrounding context, it can be considered as a classification problem. Supervised learning methods for WSD have been utilized and achieved successful results. In this section, we briefly introduce Support Vector Machines (SVM) algorithm, one of the most commonly used learning algorithms in WSD. The most advantage of SVM is that it can handle high dimensional feature vectors well in running time as well as in accuracy.

The SVM algorithm is based on the statistical learning theory and the Vapnik–Chervonenkis dimension introduced by Vladimir Vapnik [?]. It learns a linear discriminant hyperplane that separates two classes of data with the maximum margin, as shown in Figure 2.1. The examples closest to the hyperplane are called support vectors. This learning algorithm has shown empirically good performance in many fields, such as bioinformatics, text, image recognition, etc.

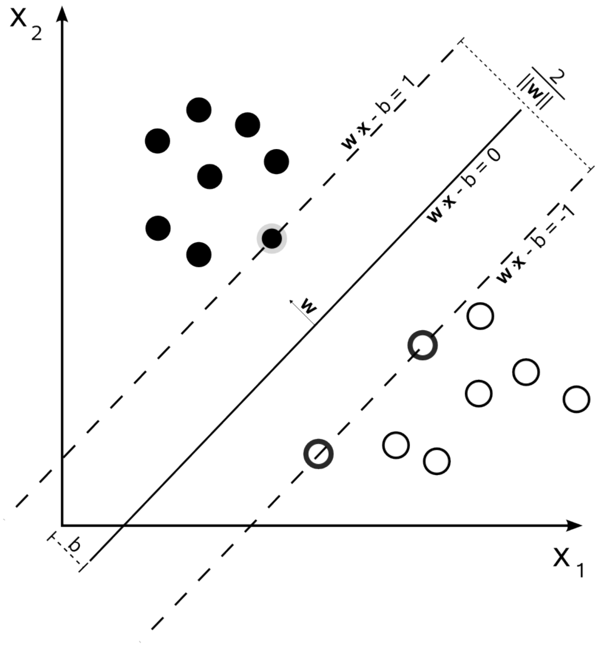


Figure 2.1: Separating Hyperplane of Support Vector Machine

2.3 Pseudoword Technique for WSD

Pseudoword technique was introduced by Gale et al. [9]. Gale constructed the pseudoword *'ability/mining'* by supposing that each use of either word is replaced by this pseudoword so that we can know the meanings of the pseudoword just by looking at the word. This method provided a pseudoword dataset of *'ability'* pseudo-sense and *'mining'* pseudo-sense. He chose two to three (nearly) unambiguous words to build up a pseudoword for an ambiguous word. The pseudoword corpus is applied for automatic testing and achieved an accuracy of 0.92. He concluded that this method is a promising one to deal with the bottleneck of acquisition of training and test data for supervised learning WSD system. However, the pseudowords in Gale's experiments are randomly chosen, which may not have a relation to real ambiguous word. Lu et al. presented equivalent pseudowords [15], in which they build up pseudowords based on real ambiguous words. However, they only performed evaluation on pseudowords (unsupervised WSD) and have no comparison between pseudowords and real ambiguous words. The task of classifying two different words are much more easier than distinguishing two senses of the same word.

In our research, we apply Lu's idea in contructing pseudowords for Vietnamese WSD and conduct experiments on both pseudowords and real words in other to have more precise evaluation on pseudoword technique.

2.4 Vietnamese WSD

Vietnamese is a language with high number of ambiguous words. Although there have been many researches on Vietnamese language processing, such as sentence segmentation, word segmentation, POS tagging, parsing, etc; in our knowledge there is no previous research on Vietnamese WSD. Dinh [7] attempted to construct a sense tagged corpus in Vietnamese by using English semantically tagged corpus and bilingual English-Vietnamese texts. However, he mainly annotated English texts in order to disambiguate English words which will be applied in English-Vietnamese machine translation system. And there is no evaluation on WSD based on his corpus, either.

Chapter 3

Method

This chapter describes the method to disambiguate word senses. SVM is used as machine learning algorithm. Features used in the SVM classifiers are also explained.

We only consider two senses for each ambiguous words since it is very difficult to cover all senses of an ambiguous word based on dictionary. Moreover, not all senses appear in the corpus. Therefore, the number of senses for an ambiguous word is supposed to be two in this paper.

The chapter is arranged as follows:

3.1 Support Vector Machines

3.2 Design of feature sets for Vietnamese WSD

3.3 Feature Selection

3.1 Support Vector Machines as Classifier for WSD

In this study, we use Support Vector Machines (SVM) for training WSD classifiers. SVM is a binary classifier. Our task is binary classification since the number of classes or senses are two. Thus SVM can be applied without any modifications. As we discussed in Section 2.2, SVM is powerful in high dimensional space. Our reported results are based on the linear kernel because in high dimensional space (the number of features is large), mapping data to a higher dimensional space does not improve the performance [**?**]. We found that other kernels gave poorer results than linear kernel in our preliminary experiment.

Figure 3.1 shows the diagram of steps in our system. Each target instance is represented as a feature vector. The last element of the vector, y, is its correct sense tag (1 or -1). Methods to construct these feature vectors will be describe in the next section.

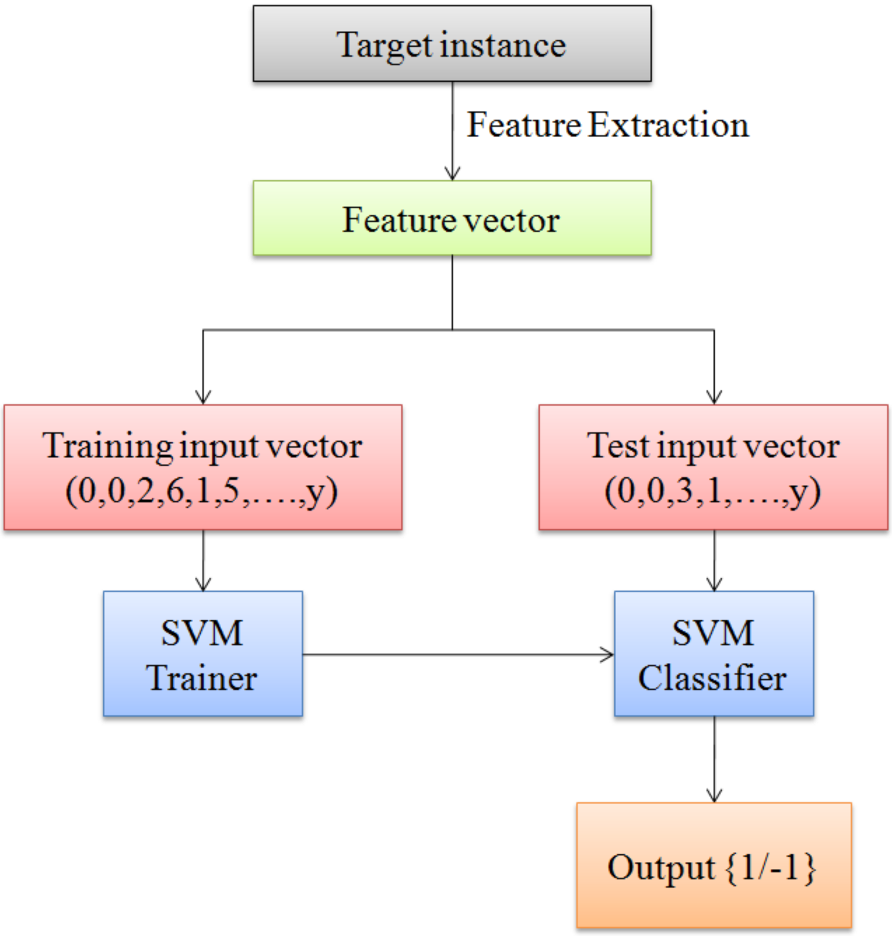


Figure 3.1: System flow chart

3.2 Design of Feature Set for Vietnamese WSD

For each target instance w, we encodes its surrounding context as feature vector. The feature set of w is denoted as in (3.1),where f_i is a feature.

$$F = \{f_1, f_2, ..., f_n\} \tag{3.1}$$

In our experiment, the feature vector is weighted according to the context of target instances in the training corpus (equation (3.2)), where ω_i is a weight of f_i. Methods for defining f_i and ω_i will be described in details for each type of feature.

$$\vec{f} = (\omega_1, \omega_2, ..., \omega_n) \tag{3.2}$$

3.2.1 Individual Features

Bag-Of-Words

Bag-Of-Words (BOW hereafter) feature encodes single words around the target word in a sentence. For example, in the following sentence, "*They **make** me happy*", the BOW of the target word "*make*" is *{they, me, happy}*. Therefore, f_i corresponds to a word appearing in the context of a target word. Numbers and punctuation marks are not used as the feature since they would not be effective clues for WSD. F contains all possible words appearing in the context of a target word in the training corpus. For each sentence l containing a target instance w in the corpus, f_i is weighted as in (3.3).

$$\omega_i = \begin{cases} t_i^1 & \text{if } f_i \text{ appears in } l \text{ and sense of } w \text{ is } s_1; \\ t_i^2 & \text{if } f_i \text{ appears in } l \text{ and sense of } w \text{ is } s_2; \\ 0 & \text{if } f_i \text{ does not appear in the context of } w \end{cases} \tag{3.3}$$

where t_i^j is the frequency of f_i that appears in the context of sense s_j of w in the training corpus.

For example, let us consider the case w = *'biển'*, s_1 = the sea, s_2 = a sign board. From the training corpus, we have the feature set as in (3.4), where two numbers in the parentheses denote frequencies of f_i in two sense set, (i.e.(t_i^1, t_i^2)).

$$\begin{aligned} F =& \{\text{nước(water)/(5,0),người(people)(0,4),xóa(clear)/(3,0),} \\ & \text{tất cả(everything)/(4,1),xe(vehicle)/(0,6),số(number)/(0,9)}\} \end{aligned} \tag{3.4}$$

Then, the BOW feature vector of the sentence "*Biển/xóa/tất cả (The sea clears everything)*" is $\vec{f} = (0, 0, 3, 4, 0, 0)$ when w has been tagged with s_1, while $\vec{f} = (0, 0, 0, 1, 0, 0)$ when w has been tagged with s_2.

Part-of-speech (POS)

This feature encodes part-of-speech of each word in a context window c around the target instance w as in (3.5), where p_i is the position of the word and P_i is its POS. p_i is an integer in the range $(-c, c)$ indicating the distance between a target word and a word in the context. If p_i is positive, the context word appears in the right context of the target word. Similarly, p_i is negative for words in the left context. If p_i exceeds sentence boundary, P_i is denoted by the null symbol ϵ. F contains all possible pairs of the position of the word in the context and its POS found in the training corpus. For each sentence in the corpus, f_i is weighted by ω_i as in (3.6).

$$f_i = (p_i, P_i) \tag{3.5}$$

$$\omega_i = \begin{cases} 1 & \text{if POS of the word at the position } p_i \text{ is } P_i; \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

For example, let us consider the case *w=‘biển’*, $c = 4$ and the set of POS features collected from the training corpus is (3.7)[1].

$$\begin{aligned} F =& \{(-4, V), (-4, P), (-4, N), (-3, N), (-2, \epsilon), \\ & (-1, E), (0, N), (1, V), (1, A), (2, A), (3, A), (4, .)\} \end{aligned} \tag{3.7}$$

Then, the feature vector of the sentence *‘Sá/V gì/P ,/, mặc/V cho/R giữa/E biển/N lạnh/A buốt/A ./.’ (No matter that the sea is very cold.)* is $\vec{f} = (0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1)$.

Collocations

This feature encodes a sequence of words (n-grams) that co-occurs with the target word. Let w_i denotes the *i-th* word to the right (or left if i is negative) of the target word, w_0 is the target word itself. If the *i-th* word exceeds sentence boundary, $w_i = \epsilon$. For each target word in the corpus, we extracted 9 collocation strings as follows:

- 2-grams: $C_{-1,0} = w_{-1}w_0; C_{0,1} = w_0w_1$
- 3-grams: $C_{-2,0} = w_{-2}w_{-1}w_0; C_{-1,1} = w_{-1}w_0w_1; C_{0,2} = w_0w_1w_2$
- 4-grams: $C_{-3,0} = w_{-3}w_{-2}w_{-1}w_0; C_{-2,1} = w_{-2}w_{-1}w_0w_1; C_{-1,2} = w_{-1}w_0w_1w_2; C_{0,3} = w_0w_1w_2w_3$

Each feature f_i is extracted as in (3.8), where l_i and r_i are the start and end position of a collocation string. Unlike the case of BOW, we don’t remove punctuation symbols or numbers in the collocations. F contains all possible collocation strings with w in the training data. For each sentence l containing the target word w in the corpus, f_i is weighted by ω_i as in Eq. (3.9).

$$f_i = (l_i, r_i, C_{l_i,r_i}) \tag{3.8}$$

$$1 < r_i - l_i < 4, l_i = -3, ..., 0, r_i = 0, ..., 3$$

$$\omega_i = \begin{cases} 1 & \text{if } C_{l_i,r_i} \text{ is found in } l; \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

For example, let us consider the sentence *l=“Sá/ gì/ ,/ mặc/ cho/ giữa/ biển/ lạnh/ buốt/./”*, $w_0 =$*“biển”*. The feature set is collected as in (3.10).

[1] ‘.’ represents POS of punctuation.

$$\begin{aligned}
F = \{&(-1, 0, \textit{trên-biển}), (-1, 0, \textit{giữa-biển}),\\
&(0, 1, \textit{biển-nóng}), (0, 1, \textit{biển-lạnh}),\\
&(-2, 0, \textit{đi-trên-biển}), (-2, 0, \textit{cho-giữa-biển}),\\
&(-1, 1, \textit{trên-biển-nóng}), (-1, 1, \textit{giữa-biển-lạnh}),\\
&(0, 2, \textit{biển-nóng-.}), (0, 2, \textit{biển-lạnh-cóng}),\\
&(-3, 0, \textit{tôi-đi-trên-biển}), (-3, 0, \textit{mặc-cho-giữa-biển}),\\
&(-2, 1, \textit{đi-trên-biển-nóng}), (-2, 1, \textit{cho-giữa-biển-lạnh}),\\
&(-1, 2, \textit{trên-biển-nóng-.}), (-1, 2,\textit{giữa-biển-lạnh-cóng}),\\
&(0, 3, \textit{biển-nóng-.-}\epsilon), (0, 3, \textit{biển-lạnh-cóng-.})\}
\end{aligned} \tag{3.10}$$

The extracted feature vector is $\vec{f} = (0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0)$ since $C_{-1,0} =$ *'giữa-biển'*, $C_{0,1} =$ *'biển-lạnh'*, $C_{-2,0} =$ *'cho-giữa-biển'*, $C_{-1,1} =$ *'giữa-biển-lạnh'*, $C_{-3,0} =$ *'mặc-cho-giữa-biển'*, and $C_{-2,1} =$ *'cho-giữa-biển-lạnh'* are found in l.

Syntactic Relation

Syntactic relations can be extracted from an annotated syntactic tree which is available in the corpus (more details of the corpus will be discussed in Chapter 4. Many relation types can be extracted from this tree, such as subject-verb, verb-object, etc. For each category of target word (that is, verb, noun or adjective), we can use different features according to Vietnamese grammar. Hereafter, each type of syntactic feature is presented as 'R-P' (e.g. Subj-N) where R stands for syntactic relation between the target word and the word used as a feature, and P stands for POS of feature word.

Syntactic features for verb

1. Subj-N: the word that is subject of the target verb w.

   For example, in Fig. 3.2(a)), *'ông'* (he) is the subject of the target word *'gửi'* (send).

2. DOB-N: the direct object of w.

   Fox example, in Fig. 3.2(a), *'đơn'* (application) is the direct object of the target word *'gửi'* (to send).

3. IOB-N: the indirect object of w.

   For example, in Fig. 3.2(b), *'kim'* (a person's name) is the indirect object of the target word *'gửi'* (send).

4. Head-V: the verb that is modified by w.

   For example, in Fig. 3.2(c), *'đi'* (go) is the head verb of the target word *'tháo gỡ'* (remove).

5. Mod-V: the verb that modifies *w*.

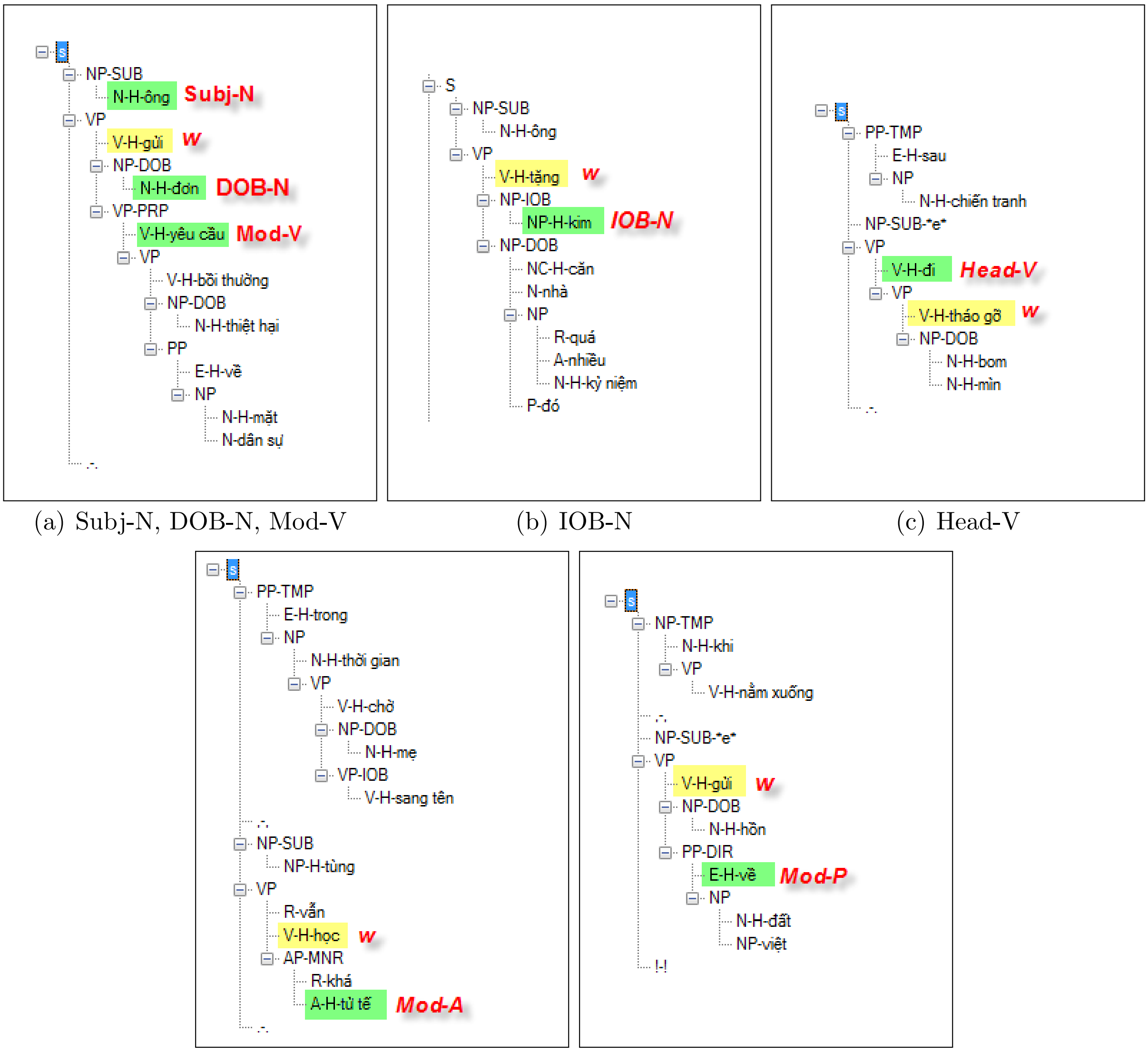
   For example, in Fig. 3.2(a), the verb *'yêu cầu'* (request) is the modifier of the target word *'gửi'* (send).

6. Mod-A: the adjective that modifies *w*.

   For example, in Fig. 3.2(d), the adjective *'tử tế'* (diligent) is the modifier of the target word *'học'* (study).

7. Mod-P: the preposition that modifies *w*.

   For example, in Fig. 3.2(e), the preposition *'về'* (to) is the modifier of the target word *'gửi'* (send).



(a) Subj-N, DOB-N, Mod-V

(b) IOB-N

(c) Head-V

(d) Mod-A

(e) Mod-P

Figure 3.2: Extracted syntactic relations for verb

Syntactic features for noun:

1. OB-V: the verb that is modified by the target noun w where w is its object.

   For example, in Fig. 3.3(a), the verb *'gặp'* (meet) has target word *'rắn'* (snake) as its object.

2. Head-N: the noun that is a head of w or modified by w.

   For example, in Fig. 3.3(b), the noun *'đầu'* (the beginning) is modified by the target word *'năm'* (year).

3. Head-P: the head preposition of the prepositional phrase including w.

   For example, in Fig. 3.3(c), the preposition *'trên'* (on) is the head preposition of the prepositional phrase including the target word *'biển'* (the sea).

4. Mod-A: the adjective that modifies w.

   For example, in Fig. 3.3(c), the adjective *'yên tĩnh'* (quiet) modifies the target word *'biển'* (the sea).

5. Mod-N: the noun that modifies w.

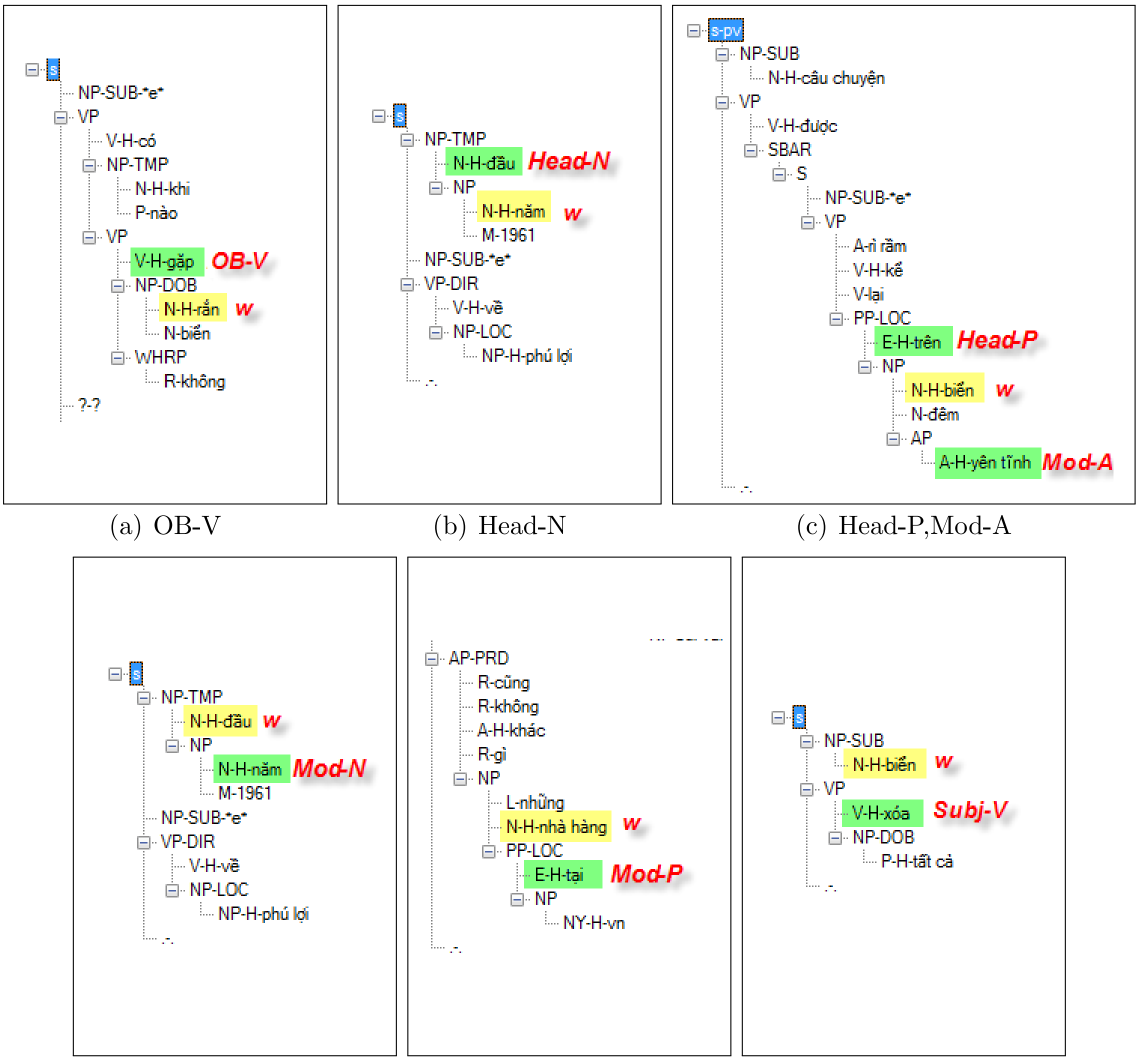
   For example, in Fig. 3.3(d), the noun *'năm'* (year) modifies the target word *'đầu'* (the beginning).

6. Mod-P: the head preposition of the prepositional phrase that modifies w.

   For example, in Fig. 3.3(e), the preposition *'tại'* (at) is the head preposition of the prepositional phrase that modifies the target word *'vn'* (Vietnam).

7. Subj-V: the predicative verb of w when w is a subject.

   For example, in Fig. 3.3(f), the verb *'xóa'* (clear) is the predicate of the subject *'biển'* (the sea).



(a) OB-V

(b) Head-N

(c) Head-P,Mod-A

(d) Mod-N

(e) Mod-P

(f) Subj-V

Figure 3.3: Extracted syntactic relations for noun

Syntactic feature for adjective:

1. Subj-N: the subject of the target adjective w where w is a predicate.

   For example, in Fig 3.4(a), the noun *'con cái'* (children) is the subject of the target word *'khôn lớn'* (grown).

2. S-V: the predicative verb of w where w is a subject.

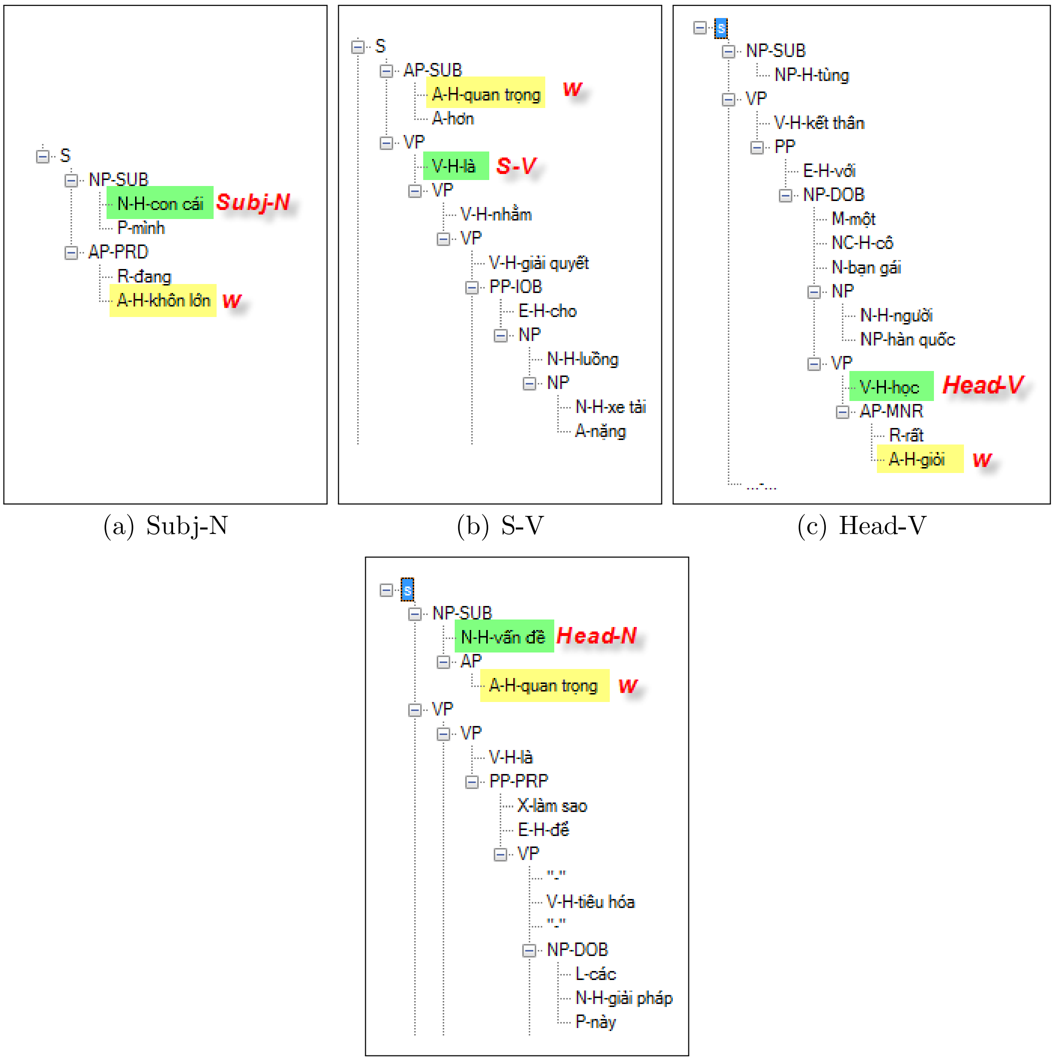
   For example, in Fig 3.4(b), the verb *'là'* (be) is the predicate of the target word *'quan trọng'* (important).

3. Head-V: the verb that is modified by w.

   For example, in Fig 3.4(d), the verb *'học'* (study) is modified by the target word *'giỏi'* (good).

4. Head-N: the noun that is modified by w.

   For example, in Fig 3.4(c), the noun *'vấn đề'* (problem) is modified by the target word *'quan trọng'* (important).



(a) Subj-N

(b) S-V

(c) Head-V

(d) Head-N

Figure 3.4: Extracted syntactic relations for adjective

The syntactic feature vector is constructed in the same manner as in POS and Collocation feature. Let sl_i denotes the syntactic relation (Subj-V,Mod-A,...), t_i is a word which has a syntactic relation sl_i with the target word. Each syntactic feature is represented as in (3.11). F is a set of all possible words that have some syntactic relations with the target word in the training corpus. For each sentence l containing a target word w in the corpus, f_i is weighted as in (3.12).

$$f_i = (sl_i, t_i) \tag{3.11}$$

$$\omega_i = \begin{cases} 1 & \text{if } w \text{ and } t_i \text{ are in the syntactic relation } sl_i \text{ in } l \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

For example, let us consider the sentence l which has syntactic tree as in Figure 3.5. In l, $w =$*'biển'* (the sea), and two syntactic relational words $Head - P$ and $Mod - A$ are *'trong'* (in) and *'sâu'* (deep). Assume that the feature set for w obtained in the training corpus is (3.13). Then, the extracted feature vector is $\vec{f} = (0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0)$ since (Head-P,trong) and (Mod-A,sâu) are found in l.

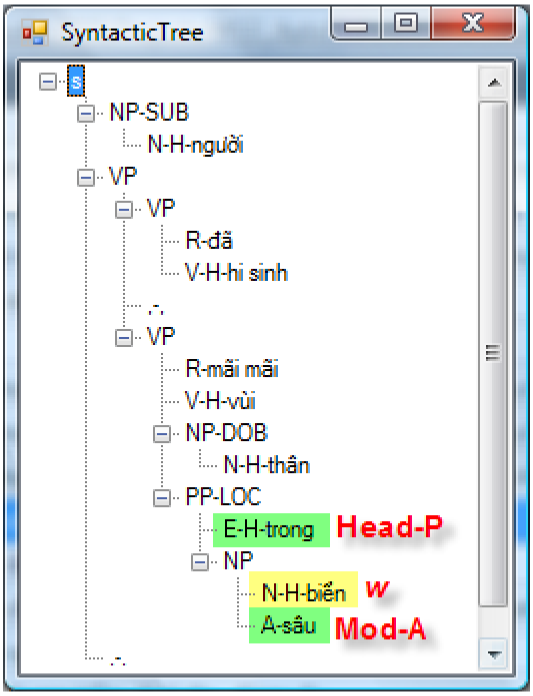


Figure 3.5: An example of extracted Syntactic feature for the target word *'biển'*

$$
\begin{aligned}
F = \{&(\text{OB-V}, \textit{đi}), (\text{OB-V}, \textit{tắm}),\\
&(\text{Head-N}, \textit{đáy}), (\text{Head-N}, \textit{mặt}),\\
&(\text{Head-P}, \textit{trên}), (\text{Head-P}, \textit{trong}),\\
&(\text{Mod-A}, \textit{đẹp}), (\text{Mod-A}, \textit{sâu}),\\
&(\text{Mod-N}, \textit{nhà}),\\
&(\text{Mod-P}, \textit{ngoài}),\\
&(\text{Subj-Verb}, \textit{thét gào})\}
\end{aligned} \tag{3.13}
$$

Syntactic features are extracted from syntactic trees annotated in the corpus. Detail procedures to extract syntactic features are described in Appendix A.

3.2.2 Feature Combinations

Feature combination is an effective way to enhance the performance of WSD systems. While each individual feature has some certain advantages and disadvantages, it would be expected that combining them together can take full advantage of each one. Since each individual feature is extracted as a numeric vector, we can easily concatenate those vectors together to build up a combined feature vector in our experiment.

In this research, the following feature combinations are considered:

o 2-feature-combination:

- BOW+Collocation: $F_{combine} = \{F_{BOW}, F_{Collocation}\}$
- BOW+Syntactic: $F_{combine} = \{F_{BOW}, F_{Syntactic}\}$
- Collocation+Syntactic: $F_{combine} = \{F_{Collocation}, F_{Syntactic}\}$

o 3-feature-combination:

- BOW+Collocation+Syntactic: $F_{combine} = \{F_{BOW}, F_{Collocation}, F_{Syntactic}\}$

We don't include POS feature in any combination since its performance is not so effective. The details will be discussed in Chapter 5.

3.3 Feature Selection

Feature selection is a technique to select a subset of relevant features for building a robust learning model. In BOW, we apply feature selection to eliminate words which are not really effective in disambiguation. In POS feature, we apply feature selection to find the best window size c for training WSD classifier of a target word. There are many methods of feature selection in classification. In this research, we apply the one introduced by Lee and Ng. [13] for BOW as follows:

A word k is considered a keyword (feature) of target word w iff:

1. The conditional probability of sense i of w given keyword k must not less than a predefined threshold M_1:

   $$cp(i|k) = \frac{N_{i,k}}{N_k} \geq M_1 \tag{3.14}$$

   where N_k is the number of occurrence of k, $N_{i,k}$ is the number of occurrence of k in the context of the sense i.

2. k occurs at least M_2 times in the contextx of one of sense i of w:

   $$N_{i,k} \geq M_2 \tag{3.15}$$

However, according to the results of our experiments, we see that M_2 does not affect the results as much as M_1. Moreover, M_1 can imply the meaning of M_2, so we only use M_1 in BOW feature selection. M_1 is determined by the feature selection procedure described below.

For POS feature, we vary many values of window size c to find the best one for training the classification model.

The feature selection procedure to determine the threshold M and window size c is summarized as follows:

1. For some T (T can be the threshold M or the window size c)
   (a) Split the training data into 80% training set and 20% development set.
   (b) Build the training feature vectors from the training set based on T.
   (c) Build the test feature vectors from the development set
   (d) Calculate the accuracy of the trained WSD classifiers on the development set.
   (e) Repeat the steps above 5 times by changing training and development set (5-fold cross validation).

2. Repeat the procedure 1 for various values of T.

3. Choose T with the highest accuracy.

We don't apply feature selection for Collocation and Syntactic feature since we hope that each collocation string or syntactic relation is meaningful for training the classification model. Furthermore, only a small number of features are available for one sentence. It might be better not to remove features by a selection algorithm but use all features.

The details of feature selection setup will be described in Chapter 5.

Chapter 4

Task

One of the goal of this thesis is to evaluate pseudoword technique. Pseudoword technique is well known for WSD, especially applied when no sense tagged corpus is available. This paper will explore how well pseudoword technique can simulate real WSD. This chapter describes three tasks to evaluate pseudoword technique as well as to explore effective features for supervised learning of Vietnamese WSD, and the corpora which were built for these tasks. The chapter is organized as follows,

4.1 Corpus
4.2 Pseudoword Task
4.3 Real Word Task
4.4 Pseudoword and Real Word Task

4.1 Corpus

Since there has been no sense tagged corpus for Vietnamese WSD, two kind of sense tagged corpora were built based on Vietnamese Treebank [?]. Vietnamese Treebank is a corpus contains around 10.000 sentences manually annotated with syntactic trees. POS of each word is also annotated in the corpus. POS and Syntactic features (described in Subsection 3.2.1) are derived from annotations in Vietnamese Treebank. In order to train and test WSD classifiers, correct senses for target instances in Vietnamese Treebank are tagged in two different manners, thus two sense tagged corpora, PW corpus and RW corpus, are constructed. The details of these two corpora are explained in the succeeding sections.

4.2 Pseudoword Task

Since no sense is annotated in Vietnamese Treebank, we first applied the pseudoword technique to automatically develop a sense tagged corpus. Let us suppose V_1 and V_2 are two different words. Pseudoword V_1-V_2 is imaginary word implying it is V_1 or V_2. Then V_1 or V_2 in the corpus are replaced with the pseudoword V_1-V_2. Now we can regard the original word V_1 or V_2 as a sense (we call it 'pseudo-sense' hereafter) of V_1-V_2. Note that the corpus after V_1 or V_2 are replaced with V_1-V_2 can be regarded as a sense tagged corpus. Pseudoword task (PW task hereafter) is a task to determine the pseudo-sense (V_1 or V_2) of the pseudoword V_1-V_2 in the sentence. Although it is not a real WSD, a pseudo-sense tagged corpus can be easily available without any human intervention.

In many previous researches applying pseudoword technique to evaluate WSD methods, two word V_1 and V_2 are selected randomly. However, in this research, V_1 and V_2 are chosen considering the meanings of a certain word, similar to 'equivalent pseudoword' proposed by Lu et al. [15].

Let us suppose w is a target word. We use VDict Vietnamese dictionary [**?**] to look up meanings of w. Let s_1, s_2 be two meanings (or senses) of w. Then, we find two Vietnamese words V_1, V_2 that reflect the meanings of s_1, s_2 respectively. V_1, V_2 are supposed to be monosemous. Next, V_1 and V_2 are joined together to make a pseudoword V_1-V_2. Finally, all appearances of V_1, V_2 in the corpus are replaced by V_1-V_2. Each sentence contains V_1, V_2 now has pseudoword V_1-V_2 as the target word, where V_1 and V_2 are two correct pseudo-senses of V_1-V_2. We call the obtained corpus as 'PW corpus'. Disambiguation of the pseudoword V_1-V_2 would simulate the disambiguation of the original target word w.

For example, the word '*biển*' in Vietnamese is an ambiguous word. It has two senses: '*the sea*' and '*a board*' in VDict dictionary. Two Vietnamese words: V_1=*'sông'*(river) and V_2=*'bảng'*(board) which reflect these two senses of the word '*biển*' are combined together to make a pseudoword *'sông-bảng'*. Then, all sentences contain '*sông*' and '*bảng*' are replaced by '*sông-bảng*'. The word '*sông*' or '*bảng*' in each sentence are now regarded as the correct sense of '*sông-bảng*' in that sentence. Disambiguation of *'sông-bảng'* would be similar to that of *'biển'*. Furthermore, in order to increase the number of training and test instances as well as maximize the ability of pseudoword to simulate real word, V_1 and V_2 can be more than one word.

We chose 9 verbs, 9 nouns and 5 adjectives as target words. Table 4.1, 4.2 and 4.3 reveals the target word and their two pseudo-senses of verbs, nounds and adjectives, respectively[1].

[1] IDs of target words in these tables are not continuous. This is because the set of target words are a subset of ones of RW task (as described in Section 4.3), and IDs are continuously assigned to target words of RW task.

Table 4.1: List of pseudo-verbs and their senses

ID	Target word	Pseudo-sense	Occurrences
V1	mang	đem	47
		chứa	18
V2	đưa	trao;trao tặng;chuyển giao	26
		hướng dẫn;điều khiển	22
V3	lấy	sử dụng	68
		cưới; kết hôn	15
V4	chuyển	gửi	129
		thay đổi; đánh đổi; đổi	87
V5	tiếp	đón	48
		tiếp tục	79
V6	nhận	chấp nhận;công nhận;chứng nhận;nhận lời	49
		xác nhận;phân biệt	29
V7	mất	mất mát;mất mùa;mất ngủ;mất tích	19
		chết	146
V8	xem	nhìn	190
		nghĩ	106
V9	bắt	giữ	72
		ép	12
Average number of sentences per target word			**129.11**

Table 4.2: List of pseudo-nouns and their senses

ID	Target word	Pseudo-sense	Occurrences
N1	nhà	nhà cửa; nhà đất; nhà .ang; nhà máy; nhà trọ; nhà xưởng	74
		gia đình	288
N2	nước	con nước;mặt nước;nước mắm;nước mắt;nước mặn;nước ngọt;nước ngầm;nước sạch;sông nước	95
		xã hội;đất nước;nhà nước;nước ngoài;nước nhà	216
N3	đường	đường phố;đường bộ;đường mòn	51
		hướng;cách	189
N5	biển	bảng	21
		sông	147
N6	thứ	loại	55
		hạng	17
N7	giờ	giờ phút;phút giây;phút	73
		hiện;hiện giờ	11
N9	chiều	hướng;chiều hướng	17
		chiều tối;đêm tối;tối;buổi sáng	59
N10	tên	tên tuổi	17
		kẻ	46
N11	hàng	gian hàng;mặt hàng; hàng hiệu;hàng quán;hàng hóa	40
		hàng ngũ;dòng	67
Average number of sentences per target word			**164.78**

Table 4.3: List of pseudo-adjectives and their senses

ID	Target word	Pseudo-sense	Occurrences
A1	lớn	lớn lao;rộng lớn;to lớn	16
		khôn lớn;lớn khôn;lớn tuổi;già	59
A2	nhỏ	nhỏ bé;nhỏ nhắn;nhỏ nhặt;nho nhỏ;nhỏ nhoi	20
		trẻ;trẻ trung;non trẻ	85
A4	khó	dễ	42
		nghèo	121
A5	dài	xa	71
		lâu;lâu dài	79
A9	nặng	nặng nề;nặng nhọc;trĩu nặng	28
		nghiêm trọng;quan trọng	47
Average number of sentences per target word			**113.6**

As described above, the pseudo-senses of some target words are represented by a set of words, such as V2.đưa, N1.nhà and A1.lớn. These target words and pseudo-senses are selected so that we can obtain considerable number of example sentences in PW corpus. Occurrences of pseudowords in PW corpus are also shown in the tables. The PW corpus comprises 1162 sentences for verbs, 1483 sentences for nouns and 568 sentences for adjectives. The average samples of pseudo-verbs, pseudo-nouns and pseudo-adjectives are 129.11, 164.78 and 113.6, respectively. The reason why number of adjective instances is less than verb and noun is frequency of ambiguous adjective in the corpus is not much. Besides, since the senses of adjectives are too fine-grained, it's very difficult to distinguish them.

In PW task, the experiments are conducted using only PW corpus. The whole procedure for each type of feature is summarized below.

1. Split the PW corpus into 90% training set and 10% test set.
2. Run feature selection in the training set (Section 3.3).
3. Build the training feature vectors from the training set based on feature selection's parameter.
4. Build the test feature vectors from the test set
5. Calculate the classification accuracy.
6. Change the training and test set and repeat step 2-5 (10-fold cross validation). Calculate the average accuracy on 10 times trial.

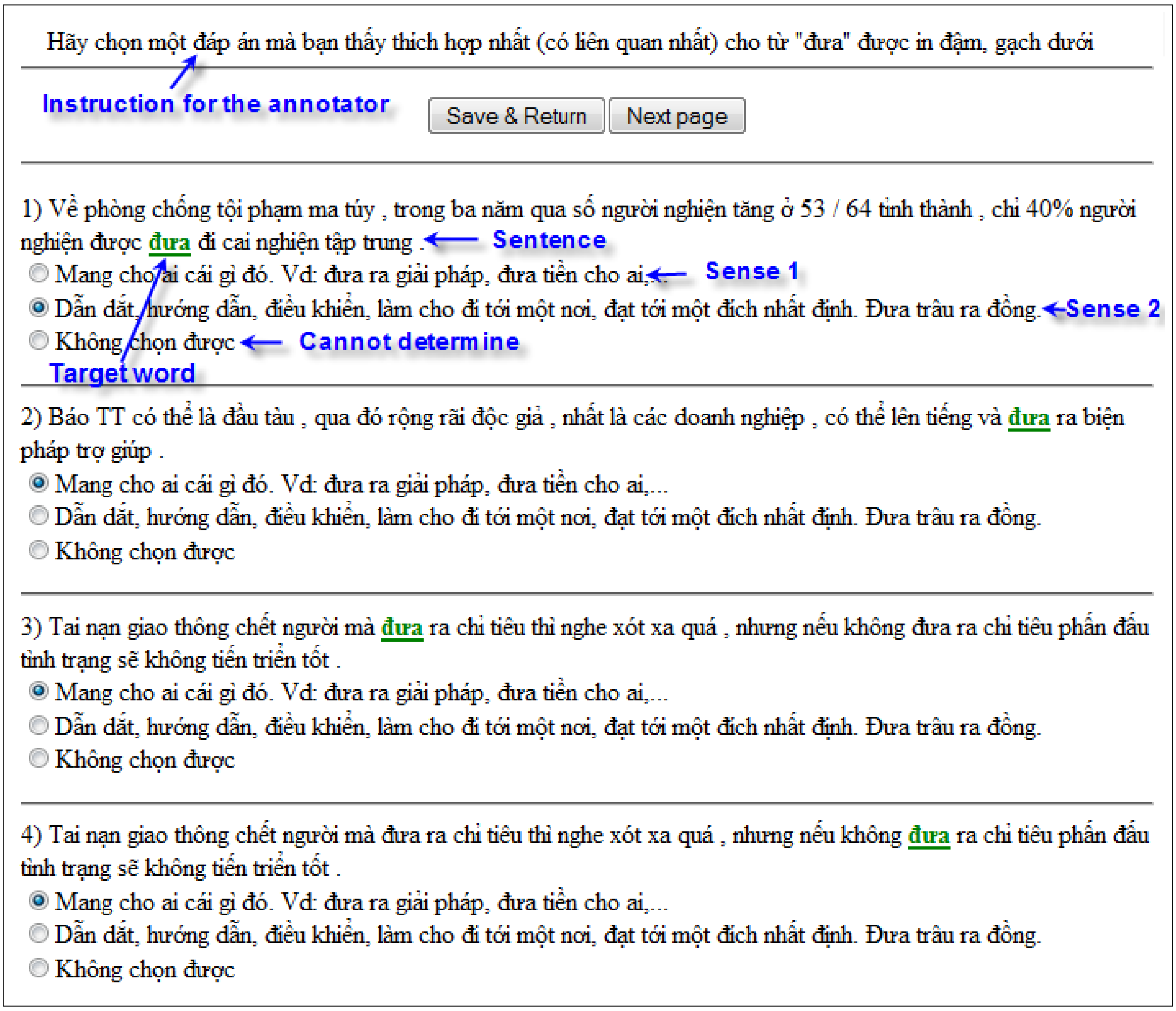
4.3 Real Word Task

Although the pseudoword task have several advantages for evaluation of WSD methods, it is obviously different with real WSD task. In order to investigate effective features more precisely, we conducted experiments of the ordinary WSD. In order to distinguish it with PW task, we call it Real Word task (RW task hereafter). Furthermore, we can evaluate applicability of pseudoword technique for WSD by comparing results between PW and RW tasks.

For target words of RW task, we use the same words selected as the target words in PW task. Furthermore, we added more tartget words in RW task. Number of target words of verbs, nounds and adjectives is 9, 11 and 9, respectively. Full lists of chosen target words and their senses are shown in Table 4.4 (verbs), 4.5 (nouns) and 4.6 (adjectives). Note that the ID for each target word corresponds to the ID in PW task (in Table 4.1, 4.2 and 4.3).

In order to train SVM classifiers in RW task, a sense tagged corpus is required. We manually tagged the senses of those target words based on VDict Vietnamese dictionary [?]. The tagging process was conducted as follows: for each target word, about 100 sentences were chosen for sense tagging, resulted in around 3000 sentences for all verbs, nouns and adjectives. Two Vietnamese native speakers were invited to judge which sense a target word has in those sentences. Two people did the task independently. The Inter-tagger aggreement (ITA) was 90.63%. Figure 4.1 shows an example of sense tagging page for the target word *'đưa'*. The first line in the figure is the instruction for the annotator to annotate senses of target word (*Please choose an answer that is the correct (or most relevant) meaning of the word 'đưa' which is bold in following sentences*). In each sentence, the annotator is given 3 answers for sense 1, sense 2 and 'cannot determine which sense is correct in this case.'

We call the above sense tagged corpus 'RW corpus'. Number of sentences for each sense of each target word is also shown in Table 4.4, 4.5 and 4.6. The average numbers of sentences for verbs, nouns and adjectives are 92.33, 116.73 and 92.11, respectively.



Hãy chọn một đáp án mà bạn thấy thích hợp nhất (có liên quan nhất) cho từ "đưa" được in đậm, gạch dưới

1) Về phòng chống tội phạm ma túy , trong ba năm qua số người nghiện tăng ở 53 / 64 tỉnh thành , chỉ 40% người nghiện được **đưa** đi cai nghiện tập trung .

○ Mang cho ai cái gì đó. Vd: đưa ra giải pháp, đưa tiền cho ai,...

● Dẫn dắt, hướng dẫn, điều khiển, làm cho đi tới một nơi, đạt tới một đích nhất định. Đưa trâu ra đồng.

○ Không chọn được

2) Báo TT có thể là đầu tàu , qua đó rộng rãi độc giả , nhất là các doanh nghiệp , có thể lên tiếng và **đưa** ra biện pháp trợ giúp .

● Mang cho ai cái gì đó. Vd: đưa ra giải pháp, đưa tiền cho ai,...

○ Dẫn dắt, hướng dẫn, điều khiển, làm cho đi tới một nơi, đạt tới một đích nhất định. Đưa trâu ra đồng.

○ Không chọn được

3) Tai nạn giao thông chết người mà **đưa** ra chỉ tiêu thì nghe xót xa quá , nhưng nếu không đưa ra chỉ tiêu phấn đấu tình trạng sẽ không tiến triển tốt .

● Mang cho ai cái gì đó. Vd: đưa ra giải pháp, đưa tiền cho ai,...

○ Dẫn dắt, hướng dẫn, điều khiển, làm cho đi tới một nơi, đạt tới một đích nhất định. Đưa trâu ra đồng.

○ Không chọn được

4) Tai nạn giao thông chết người mà đưa ra chỉ tiêu thì nghe xót xa quá , nhưng nếu không **đưa** ra chỉ tiêu phấn đấu tình trạng sẽ không tiến triển tốt .

● Mang cho ai cái gì đó. Vd: đưa ra giải pháp, đưa tiền cho ai,...

○ Dẫn dắt, hướng dẫn, điều khiển, làm cho đi tới một nơi, đạt tới một đích nhất định. Đưa trâu ra đồng.

○ Không chọn được

Figure 4.1: An example of sense tagging page for the target word *'đưa'*

Table 4.4: List of ambiguous verbs and their senses

ID	Target word	Senses	Occurrences
V1	mang	To bring, to take something to somebody/somewhere	66
		To contain some characteristics of something	34
V2	đưa	To give something to somebody	45
		To help somebody do something	55
V3	lấy	To use something for doing something	40
		To get married	46
V4	chuyển	To send (an email, postcard, document,. . .)	30
		To change (state)	48
V5	tiếp	To welcome somebody	13
		To continue doing something	28
V6	nhận	To accept, admit to something	55
		To recognize someone	45
V7	mất	To lose something, someone	84
		To die	20
V8	xem	To look at	91
		To think	32
V9	bắt	To arrest somone	83
		To force somebody doing something	16
Average number of sentences per target word			**92.33**

Table 4.5: List of ambiguous nouns and their senses

ID	Target word	Senses	Occurrences
N1	nhà	House	87
		Family	44
N2	nước	Water	69
		Country	81
N3	đường	A path that connects two locations (street)	100
		A way to do something	27
N4	đầu	A tip, an end	36
		The beginning	70
N5	biển	The sea	7
		sign, plate	95
N6	thứ	kind, sort, category	33
		place, position	72
N7	giờ	an hour	44
		now	64
N8	tiếng	language	68
		sound	82
N9	chiều	dimension	25
		afternoon	72
N10	tên	name	78
		a word used to indicate a person (impolite)	22
N11	hàng	product	95
		line	13
Average number of sentences per target word			**116.73**

Table 4.6: List of ambiguous adjectives and their senses

ID	Target word	Senses	Occurrences
A1	lớn	big	137
		old	13
A2	nhỏ	small	71
		young	35
A3	phải	something right	87
		right hand side	11
A4	khó	difficult	71
		poor	6
A5	dài	long (distance)	73
		long (time)	15
A6	trên	above	13
		more than, over	57
A7	trước	before	93
		in front of	16
A8	tốt	good in quality (product)	54
		nice, honest (person)	22
A9	nặng	heavy (weight)	21
		serious (illness)	34
Average number of sentences per target word			**92.11**

In RW task, the experiments are conducted using only RW corpus. The whole procedure for each type of feature is summarized below.

1. Split the RW corpus into 90% training set and 10% test set.

2. Run feature selection in the training set (Section 3.3).

3. Build the training feature vectors from the training set based on feature selection's parameter.

4. Build the test feature vectors from the test set

5. Calculate the classification accuracy.

6. Change the training and test set and repeat step 2-5 (10-fold cross validation). Calculate the average accuracy on 10 times trial.

4.4 Pseudoword and Real Word Task

In Pseudoword and Real word task (PW-RW task hereafter), we use PW corpus for training WSD classifiers, then classifiers are tested using RW corpus. This task is conducted in order to evaluate the effectiveness of pseudoword technique when it is applied to real WSD. Since the target words are shared in our PW and RW tasks and a pseudo-sense (V_1 or V_2) in PW task corresponds to a sense (s_1 or s_2) in RW task, WSD classifiers trained from PW corpus could be applicable for RW task. The attractive advantages of this approach is that no sense tagged corpus is required for supervised learning of WSD systems. The whole procedure for each feature type of this task is summarized below.

1. Run feature selection using the PW corpus (Section 3.3).
2. Build the training feature vectors from the PW corpus based on feature selection's parameter.
3. Build the test feature vectors from the RW corpus
4. Calculate the classification accuracy.

Chapter 5

Evaluation

In this chapter, experiments on three tasks described in Chapter 4 are conducted. The first task (PW task) is applied to discover the effectiveness of different kinds of features without sense tagged corpus. Then, the RW task is applied with the sense tagged corpus. Finally, the PW-RW task is carried out. The accuracy differences among feature types are studied across those three systems for comparison and discussion.

The chapter includes following sections

5.1 Experiment Setup

5.2 Results of Pseudoword Task

5.3 Results of Real Word Task

5.4 Results in Pseudoword and Real Word Task

5.1 Experiment Setup

We conducted 3 experiments for 3 tasks described in Chapter 4 as follows:

1. PW task: using PW corpus for training and test.
2. RW task: using RW corpus for training and test.
3. PW-RW task: using PW corpus for training WSD classifiers, then classifiers are tested on RW corpus.

For each experiment, we firstly evaluate the effectiveness of individual features, then the feature combinations. LibSVM [4] is used as SVM classifiers since it is an open source tool and easy to use. The baseline method used in the experiments is the most frequent sense method. That is, all test instances of a target word are tagged with the most frequent sense appeared in the training data. For example, all test data of the word V1.mang are classified as '*To bring, to take something to somebody or somewhere*' (s_1) because this sense dominated the other. In PW-RW task, we use two baselines. The first baseline is the system which always chooses the most frequent sense of PW corpus, the second baseline is the system choosing the most frequent sense of RW corpus. Comparison between these two baselines also enable us to verify how well pseudoword can simulate real word WSD.

The evaluation criteria for WSD systems is the accuracy of the sense classification defined as in (5.1).

$$acc = \frac{\text{number of correct instances}}{\text{total number of instances}} \tag{5.1}$$

5.2 Results of Pseudoword Task

5.2.1 Effectiveness of Individual Features

Firstly, we applied each feature separately to see the effectiveness of it. Table 5.1, 5.2 and 5.3 show the results of four individual features: BOW, POS, Collocation and Syntactic features, which were extracted from the PW corpus based on the method in Section 3. Table 5.1 shows results for pseudo-verbs, Table 5.2 shows results for pseudo-nouns, Table 5.3 shows results for pseudo-adjectives and Table 5.4 shows average results fof accuracies for verbs, nouns, adjectives and all target words, respectively. The numbers in parentheses denote the differences of accuracies compared to the baseline. The bold number in each word indicates the best accuracy achieved for it. Figure 5.1 shows the results in Table 5.1, 5.2 and 5.3, while 5.2 shows results in Table 5.4 in charts.

Table 5.1: Accuracy of individual features for pseudo-verbs

Target word	Baseline	BOW	POS	Collocation	Syntactic
V1.mang	72.67	**84.33**	70.67	78.62	66.33
V2.đưa	54	**91.33**	68.17	79	55.83
V3.lấy	82.28	**95.42**	83.57	91.45	88.91
V4.chuyển	59.74	**90.28**	73.51	79.6	72.67
V5.tiếp	62.26	**90**	80.79	73.24	75.26
V6.nhận	62.92	**82.08**	46.67	78.33	65.42
V7.mất	88.53	**91.58**	82.6	89.74	87.27
V8.xem	64.21	88.76	76.34	**90.52**	71.6
V9.bắt	86	**89.75**	88	86	**89.75**
Average	70.29	**89.28** **(+18.99)**	74.48 (+4.19)	82.94 (+12.65)	74.78 (+4.49)

Table 5.2: Accuracy of individual features for pseudo-nouns

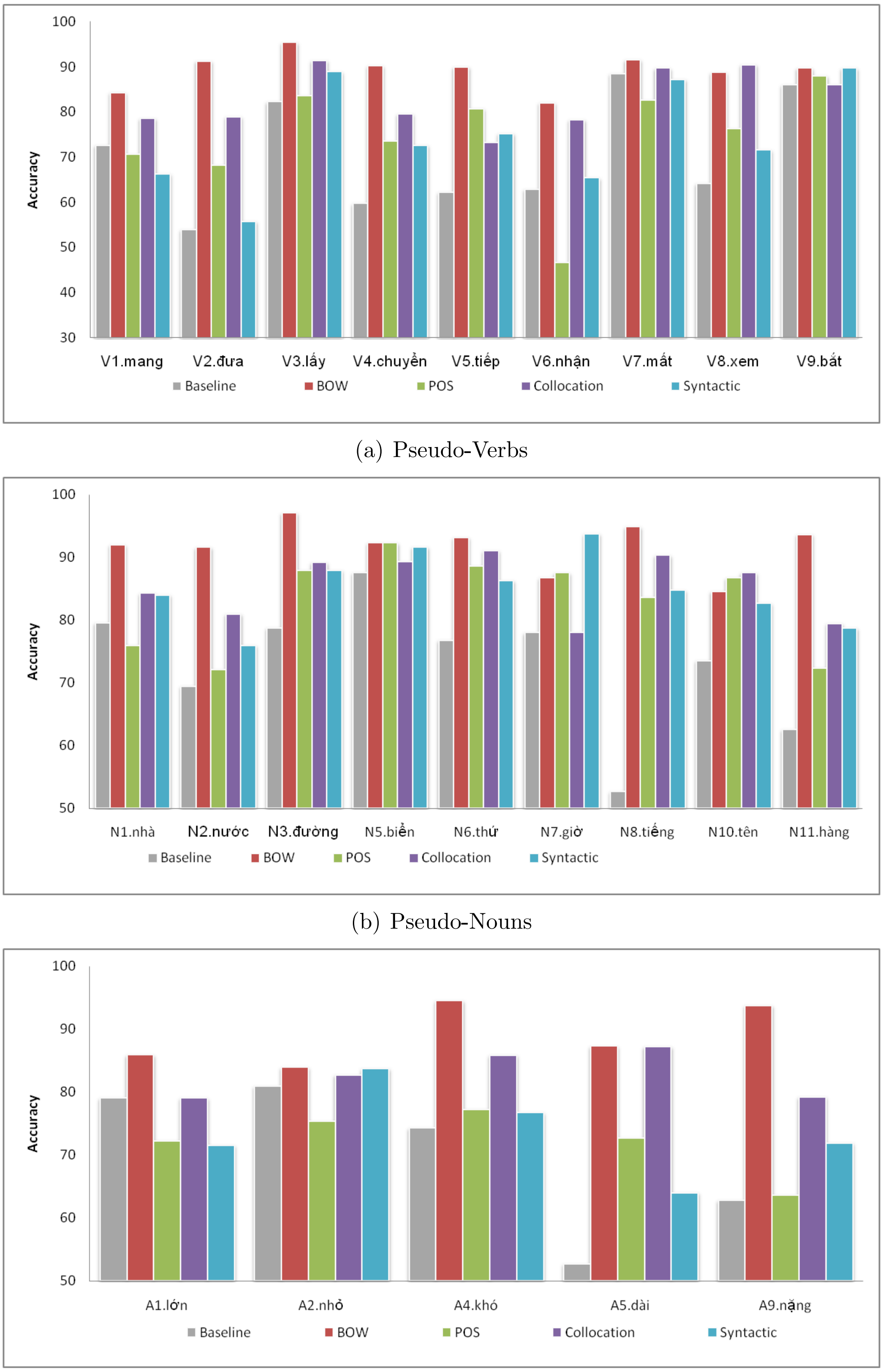
Target word	Baseline	BOW	POS	Collocation	Syntactic
N1.nhà	79.58	**92**	75.97	84.26	84
N2.nước	69.47	**91.6**	72.07	80.97	75.98
N3.đường	78.76	**97.08**	87.95	89.18	87.96
N5.biển	87.53	**92.3**	92.28	89.26	91.65
N6.thứ	76.79	**93.15**	88.63	91.07	86.31
N7.giờ	77.98	86.72	87.5	77.98	**93.75**
N8.tiếng	52.66	**94.85**	83.56	90.32	84.74
N10.tên	73.53	84.57	86.71	**87.52**	82.66
N11.hàng	62.55	**93.64**	72.37	79.46	78.73
Average	73.2	**91.77** **(+18.57)**	83 (+9.8)	85.56 (+12.36)	85.09 (+11.89)

Table 5.3: Accuracy of individual features for pseudo-adjectives

Target word	Baseline	BOW	POS	Collocation	Syntactic
A1.lớn	79.05	**85.89**	72.26	79.05	71.55
A2.nhỏ	80.91	**83.91**	75.36	82.73	83.73
A4.khó	74.28	**94.52**	77.26	85.87	76.74
A5.dài	52.66	**87.27**	72.65	87.21	64.01
A9.nặng	62.8	**93.75**	63.63	79.23	71.9
Average	69.94	**89.07** **(+19.13)**	72.23 (+2.29)	82.82 (+12.88)	73.59 (+3.65)

Table 5.4: Average accuracy of individual features for pseudo-verbs, pseudo-nouns, pseudo-adjectives and all pseudowords

	Baseline	BOW	POS	Collocation	Syntactic
Verb	70.29	**89.28** **(+18.99)**	74.48 (+4.19)	82.94 (+12.65)	74.78 (+4.49)
Noun	73.2	**91.77** **(+18.57)**	83 (+9.8)	85.56 (+12.36)	85.09 (+11.89)
Adjective	69.94	**89.07** **(+19.13)**	72.23 (+2.29)	82.82 (+12.88)	73.59 (+3.65)
All words	71.15	**90.04** **(+18.89)**	76.57 (+5.42)	83.77 (+12.62)	77.82 (+6.67)



(a) Pseudo-Verbs

(b) Pseudo-Nouns

(c) Pseudo-Adjectives

Figure 5.1: Accuracy of individual features for pseudo-words

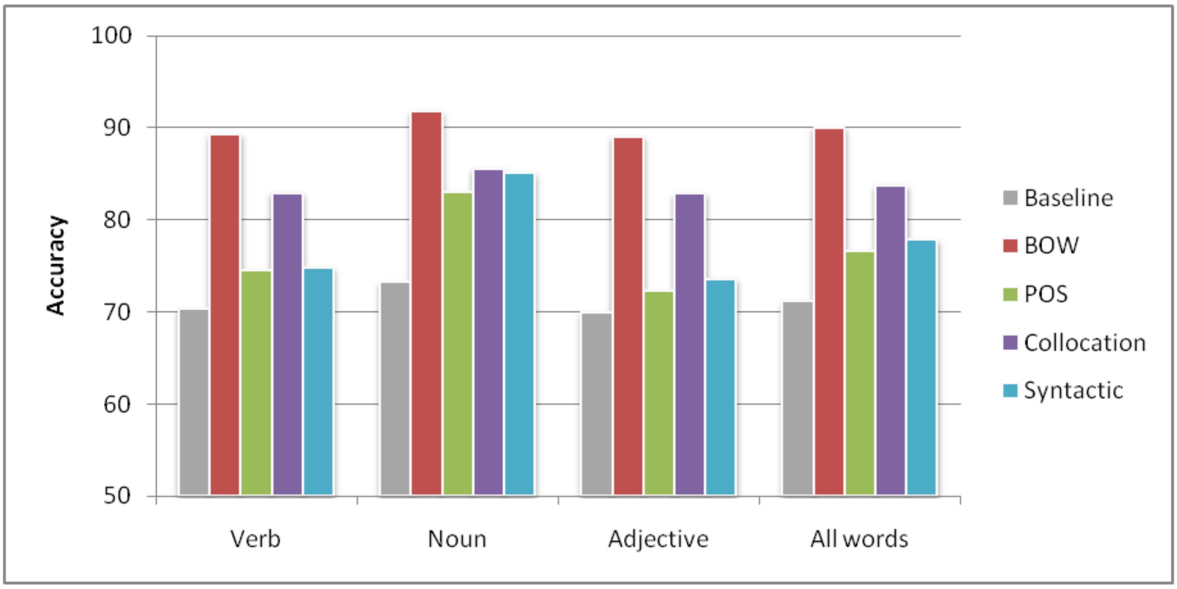


Figure 5.2: Average accuracy of each feature type for pseudowords

5.2.2 Effectiveness of Feature Combination

Since POS feature was not effective for all verbs, nouns and adjectives (accuracy was lower than the baseline), we did not include POS feature in the feature combinations. Table 5.5, 5.6 and 5.7 show the accuracy of the WSD systems with different combination of features for pseudo-verbs, pseudo-nouns and pseudo-adjectives. Figure 5.3 shows the same results in charts. Table 5.8 shows the average accuracies of each feature combination over each part-of-speech as well as all target words. The results of this table are drawn in chart graph in Figure 5.4.

Table 5.5: Accuracy of feature combinations for pseudo-verbs

Target word	Baseline	BOW + Collocation	BOW + Syntactic	Collocation + Syntactic	All 3 features
V1.mang	72.67	81.48	**87.19**	78.86	82.9
V2.đưa	54	**95**	91.67	80.67	93.33
V3.lấy	82.28	91.45	**93.99**	92.56	92.56
V4.chuyển	59.74	**94.82**	91.61	85.53	93.48
V5.tiếp	62.26	90	92.18	81.56	**93.72**
V6.nhận	62.92	**84.58**	**84.58**	81.25	**84.58**
V7.mất	88.53	90.36	**91.5**	88.53	90.36
V8.xem	64.21	94.21	90.8	90.87	**95.92**
V9.bắt	86	86	**88.5**	86	86
Average	70.29	89.77 (+19.48)	90.22 (+19.93)	85.09 (+14.8)	**90.32 (+20.03)**

Table 5.6: Accuracy of feature combinations for pseudo-nouns

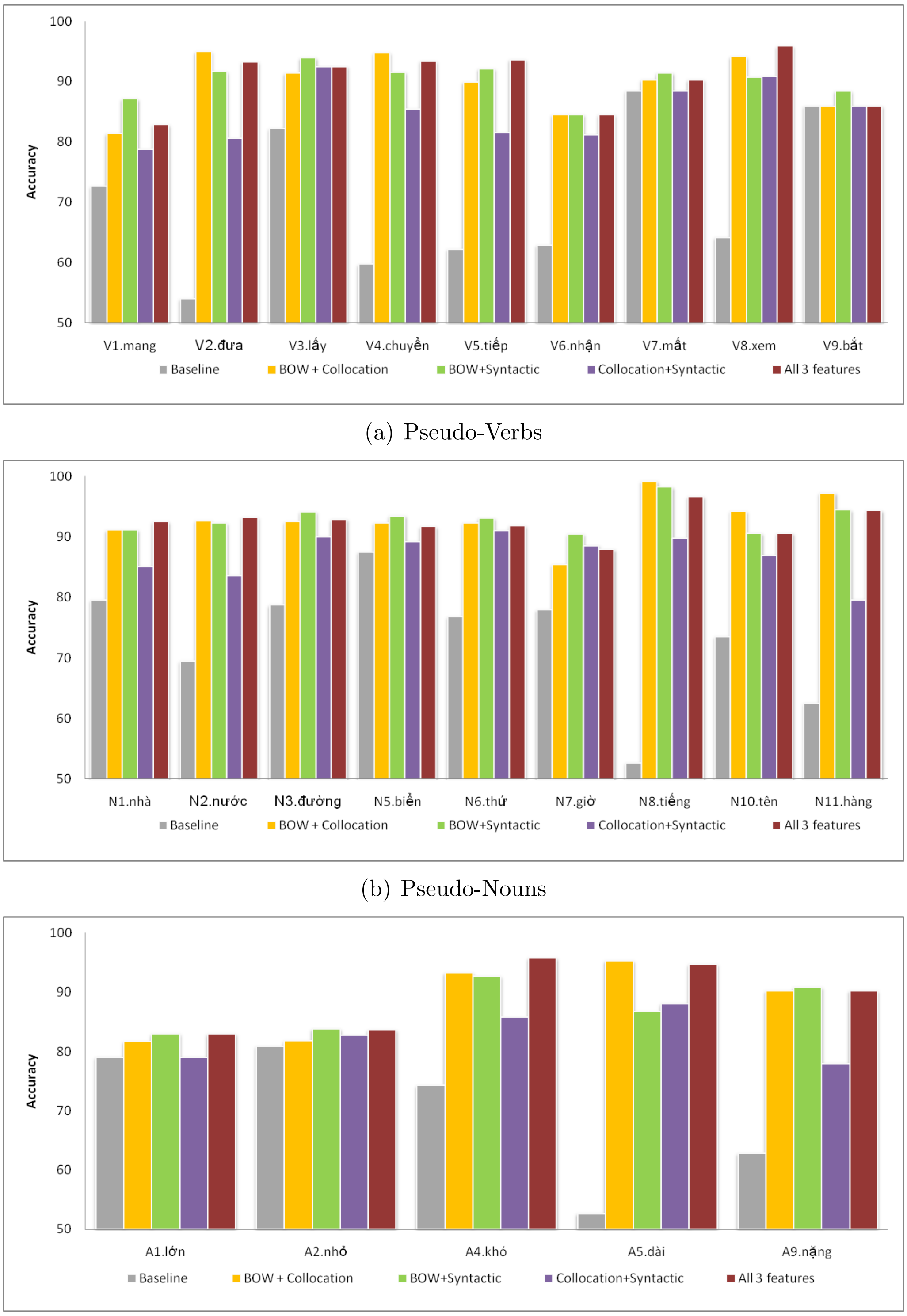
Target word	Baseline	BOW + Collocation	BOW + Syntactic	Collocation + Syntactic	All 3 features
N1.nhà	79.58	91.14	91.18	85.08	**92.54**
N2.nước	69.47	92.68	92.32	83.56	**93.28**
N3.đường	78.76	92.51	94.16	90.03	**92.93**
N5.biển	87.53	92.34	**93.52**	89.26	91.72
N6.thứ	76.79	92.32	**93.15**	91.07	91.9
N7.giờ	77.98	85.48	**90.48**	88.57	87.98
N9.chiều	52.66	**99.23**	98.26	89.81	96.66
N10.tên	73.53	**94.28**	90.62	86.95	90.62
N11.hàng	62.55	**97.27**	94.46	79.64	94.36
Average	73.2	93.03 (+19.83)	**93.13 (+19.93)**	87.11 (+13.91)	92.44 (+19.24)

Table 5.7: Accuracy of feature combinations for pseudo-adjectives

	Baseline	BOW + Collocation	BOW + Syntactic	Collocation + Syntactic	All 3 features
A1.lớn	79.05	81.73	82.98	79.05	**82.98**
A2.nhỏ	80.91	81.82	**83.82**	82.73	83.64
A4.khó	74.28	93.3	92.68	85.8	**95.8**
A5.dài	52.66	**95.33**	86.7	88.07	94.71
A9.nặng	62.8	90.24	**90.83**	77.98	90.24
Average	69.94	88.48 (+18.54)	87.4 (+17.46)	82.72 (+12.78)	**89.47 (+19.53)**

Table 5.8: Average accuracy of feature combinations for pseudo-verbs, pseudo-nouns, pseudo-adjectives and all pseudowords

Target word	Baseline	BOW + Collocation	BOW + Syntactic	Collocation + Syntactic	All 3 features
Verb	70.29	89.77 (+19.48)	90.22 (+19.93)	85.09 (+14.8)	**90.32 (+20.03)**
Noun	73.2	93.03 (+19.83)	**93.13 (+19.93)**	87.11 (+13.91)	92.44 (+19.24)
Adjective	69.94	88.48 (+18.54)	87.4 (+17.46)	82.72 (+12.78)	**89.47 (+19.53)**
All words	71.15	90.43 (+19.28)	90.25 (+19.1)	84.98 (+13.83)	**90.74 (+19.59)**



(a) Pseudo-Verbs

(b) Pseudo-Nouns

(c) Pseudo-Adjectives

Figure 5.3: Accuracy of feature combinations for pseudo-words

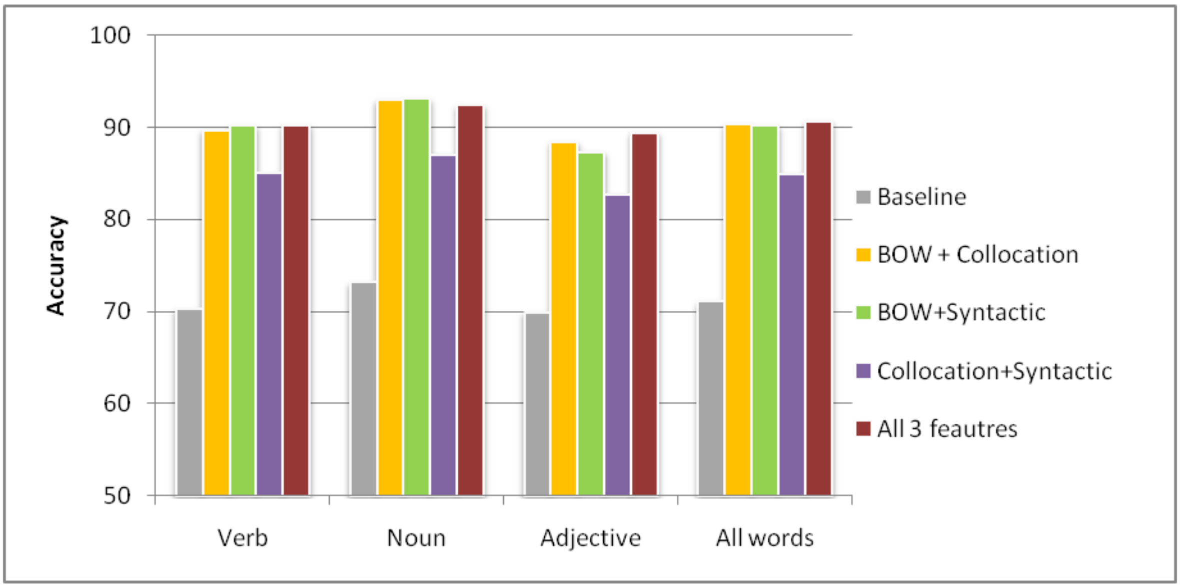


Figure 5.4: Average accuracy of each feature combination for pseudowords

5.2.3 Discussion

In the first set of experiments (Subsection 5.2.1), in overall, WSD classifiers that used BOW feature overcome all other three features in most of the target words. BOW always achieved higher accuracy than baseline and performed stably compared to the other feature types. It is reasonable to realize that BOW can capture the most contextual information of a target word. As a human usually does when facing an ambiguous word, BOW classifiers utilize the context around the target word to find the key words that help disambiguate it.

On the other hand, POS feature only contains the grammatical information of several words around the target word but not the *"meaning"* of these words. Moreover, the words to be disambiguated are in the same class of part-of-speech in our task. So, their surrounding POS may not be clearly discriminative. The results of POS feature are always the lowest in comparison with the others, even with the baseline. We can find cases where accuracy of POS classifier is lower than the baseline for all POS categories of target words, for example, V6.nhận, N1.nhà, A1.lớn and A2.nhỏ.

Collocation feature also gave the relatively high results for all part-of-speech. This is because usage of two target words in two classes are different, so their collocations are much more different. However, Collocation still couldn't better than BOW in most cases.

In average, when applying individual feature in pseudoword WSD, BOW is the most effective feature, following by Collocation, Syntactic and POS feature.

In the second set of experiments (Subsection 5.2.2), WSD classifiers with combined features gave much higher results comparing to individual features for all verbs and nouns. All systems got over baseline accuracies. The most effective feature combination is BOW + Syntactic for verbs, BOW + Collocation for adjectives and all 3 features fon nouns.

The combination without BOW is the worst effective since it doesn't take the advantage of referring wide range lexical information around the target word as BOW does. On the other hand, combinations increase the importance of the contextual words which have syntactic relations to the target word, or considering the word order, together with rich lexical information in the whole sentence. However, all feature types combination couldn't outperform the combination with just 2 features (BOW + Collocation or BOW + Syntactic) in some cases, such as V1.mang, V2.đưa, V3.lấy, V4.chuyển, V7.mất, V9.bắt, N3.đường, N5.biển, N6.thứ, N7.giờ, N8.tiếng, N10.tên, N11.hàng, A2.nhỏ and A5.dài.

Seeing Table 5.5, 5.6 and 5.7, the best feature combinations vary for individual target word. It might indicates that effective combination of features are different according to target words. It is desirable to automatically choose the best combination of features when a target word is given. This is one of our future work.

5.3 Results of Real Word Task

5.3.1 Effectiveness of Individual Features

Firstly, we applied each feature separately to see the effectiveness of it. Table 5.9, 5.10 and 5.11 show the results of four individual features: BOW, POS, Collocation and Syntactic features, which were extracted from the RW corpus based on the method in Section 3. Table 5.9 shows results for verbs, Table 5.10 shows results for nouns, Table 5.11 shows results for adjectives and Table 5.12 shows average results for each category of words as well as all words. The numbers in parentheses denote the differences of accuracies compared to the baseline. The bold number in each word indicates the best accuracy achieved for it. Figure 5.5 shows the results in Table 5.9, 5.10 and 5.11 in charts, while 5.6 show the results in Table 5.12.

Table 5.9: Accuracy of individual features for target verbs

Target word	Baseline	BOW	POS	Collocation	Syntactic
V1.mang	66.12	**85.88**	78.63	67.94	69.54
V2.đưa	55.05	**93.74**	68.59	78.99	59.9
V3.lấy	53.34	**97.64**	87.08	88.47	93.06
V4.chuyển	61.43	**86.07**	62.5	67.86	70.89
V5.tiếp	68.83	**81.17**	77.67	**81.17**	77
V6.nhận	55.05	94.85	74.75	**97.07**	74.65
V7.mất	80.73	**87.46**	78.82	82.55	84.36
V8.xem	74.07	**91.09**	73.06	83.23	75.02
V9.bắt	84.1	88.1	76.27	84.1	**92.94**
Average	66.52	**89.55** **(+23.03)**	75.26 (+9.74)	81.26 (+14.74)	77.48 (+10.96)

Table 5.10: Accuracy of individual features for target nouns

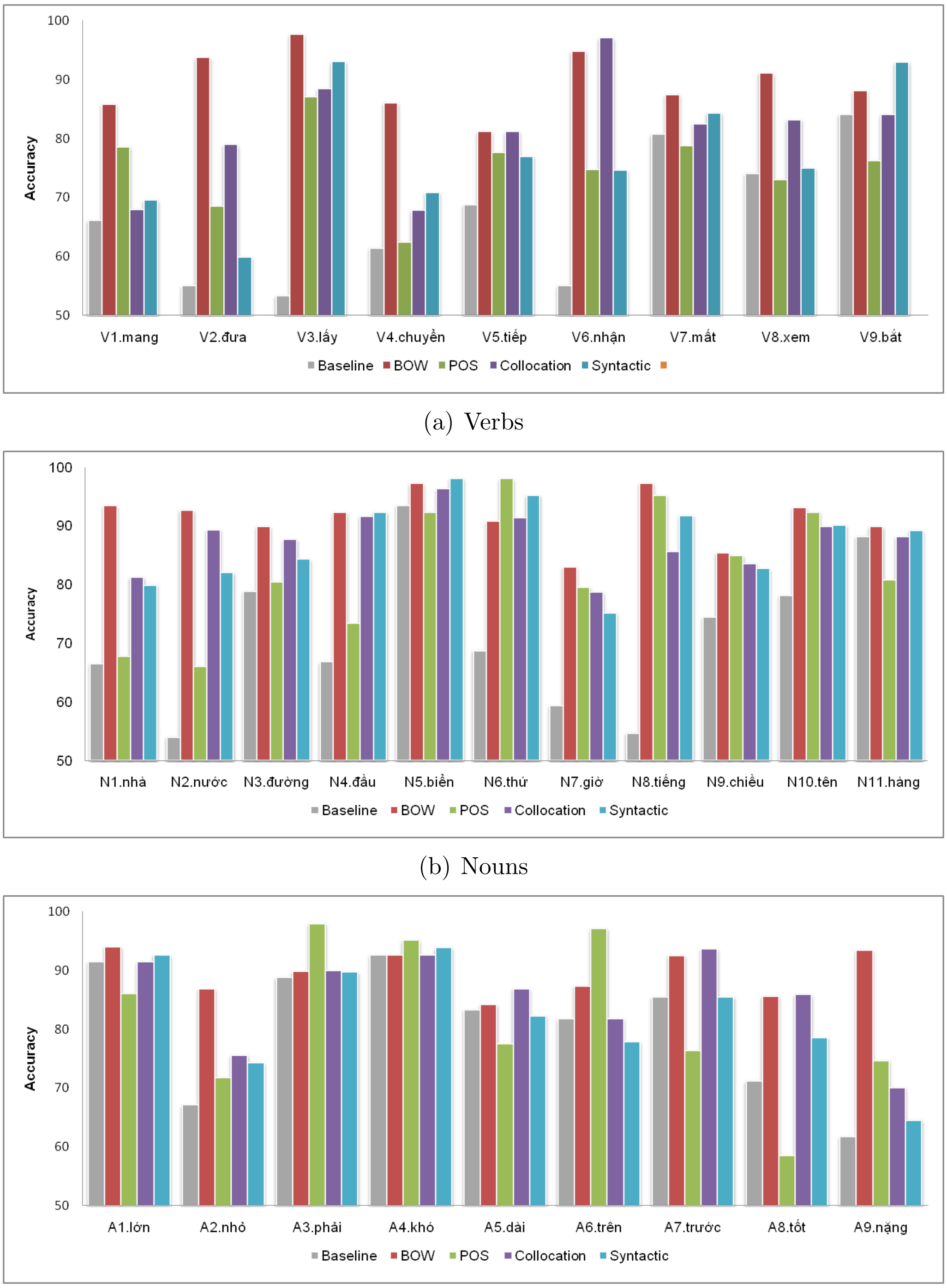
Target word	Baseline	BOW	POS	Collocation	Syntactic
N1.nhà	66.49	**93.4**	67.74	81.21	79.91
N2.nước	54	**92.66**	66.02	89.32	82.02
N3.đường	78.84	**89.87**	80.45	87.63	84.29
N4.đầu	66.82	**92.27**	73.37	91.64	**92.27**
N5.biển	93.46	97.18	92.34	96.27	**98.09**
N6.thứ	68.7	90.77	**98**	91.42	95.17
N7.giờ	59.33	**83**	79.5	78.67	75.17
N8.tiếng	54.68	**97.24**	95.19	85.61	91.76
N9.chiều	74.44	**85.38**	84.92	83.59	82.68
N10.tên	78.1	**93.14**	92.27	89.92	90.05
N11.hàng	88.18	**89.85**	80.77	88.18	89.18
Average	71.18	**91.34** **(+20.16)**	82.78 (+11.6)	87.59 (+16.41)	87.33 (+16.15)

Table 5.11: Accuracy of individual features for target adjectives

Target word	Baseline	BOW	POS	Collocation	Syntactic
A1.lớn	91.44	**94.02**	86.09	91.44	92.64
A2.nhỏ	67.12	**86.86**	71.83	75.59	74.26
A3.phải	88.85	89.85	**97.98**	89.96	89.74
A4.khó	92.64	92.64	**95.14**	92.64	93.89
A5.dài	83.31	**84.17**	77.53	86.92	82.31
A6.trên	81.78	87.38	**97.08**	81.78	77.86
A7.trước	85.54	92.53	76.41	**93.61**	85.54
A8.tốt	71.19	85.59	58.53	**85.91**	78.63
A9.nặng	61.72	**93.48**	74.71	70.05	64.57
Average	80.4	**89.61** **(+9.21)**	81.7 (+1.3)	85.32 (+4.92)	82.16 (+1.76)

Table 5.12: Average accuracy of individual features for verbs, nounds, adjectives and all target words

Target word	Baseline	BOW	POS	Collocation	Syntactic
Verb	66.52	**89.55 (+23.03)**	75.26 (+9.74)	81.26 (+14.74)	77.48 (+10.96)
Noun	71.18	**91.34 (+20.16)**	82.78 (+11.6)	87.59 (+16.41)	87.33 (+16.15)
Adjective	80.4	**89.61 (+9.21)**	81.7 (+1.3)	85.32 (+4.92)	82.16 (+1.76)
All words	72.7	**90.17 (+17.47)**	79.91 (+7.21)	84.72 (+12.02)	82.32 (+9.62)



(a) Verbs

(b) Nouns

(c) Adjectives

Figure 5.5: Accuracy of individual features for target words

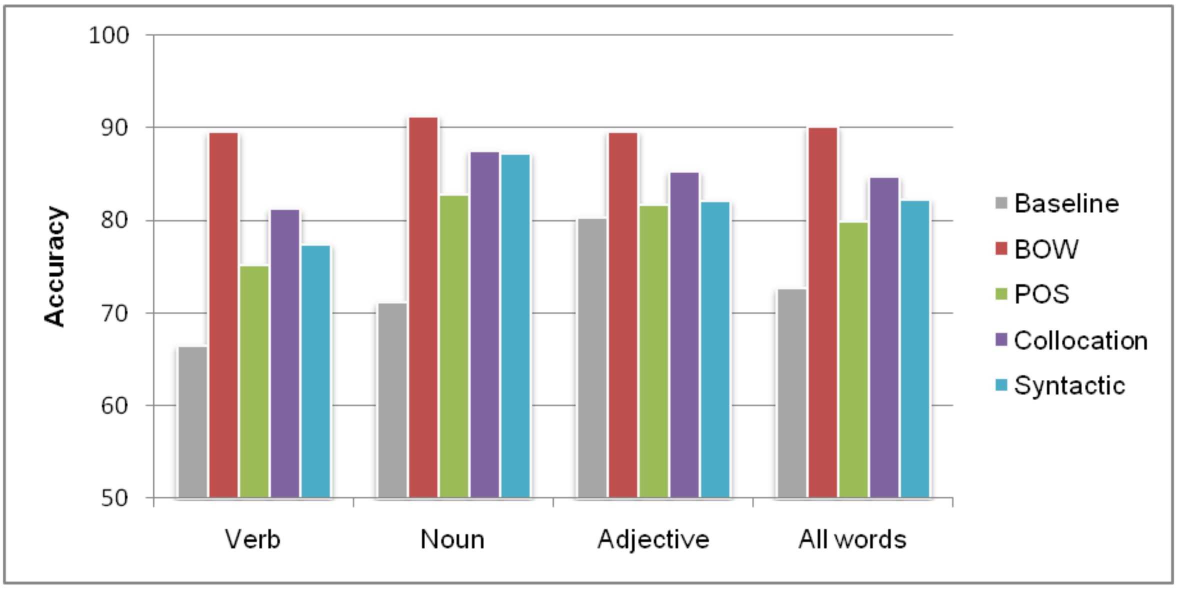


Figure 5.6: Average accuracy on each feature type for target words

5.3.2 Effectiveness of Feature Combination

Since POS feature was not effective for all verbs, nouns and adjectives (accuracy was lower than the baseline), we did not include POS feature in the feature combinations. Table 5.13, 5.14 and 5.15 show the accuracy of the WSD systems with different combination of features for verbs, nouns and adjectives. Figure 5.7 shows the same results in charts. Table 5.16 shows the average accuracies of each feature combination over each part-of-speech as well as all target words. The result is demonstated in Figure 5.8.

Table 5.13: Accuracy of feature combinations for ambiguous verbs

Target word	Baseline	BOW + Collocation	BOW + Syntactic	Collocation + Syntactic	All 3 features
V1.mang	66.12	84.57	**89.7**	77.51	88.38
V2.đưa	55.05	**89.29**	88.18	82.32	88.18
V3.lấy	53.34	**100**	98.89	93.06	**100**
V4.chuyển	61.43	88.39	**88.75**	73.04	84.82
V5.tiếp	68.83	83.67	84.5	84.5	**87**
V6.nhận	55.05	**97.07**	93.94	**97.07**	**97.07**
V7.mất	80.73	87.27	**91.18**	83.46	88.27
V8.xem	74.07	87.28	**88.64**	83.23	87.16
V9.bắt	84.1	87.1	**94.14**	92.23	93.23
Average	66.52	89.4 (+22.88)	**90.88 (+24.36)**	85.16 (+18.63)	90.46 (+23.93)

Table 5.14: Accuracy of feature combinations for ambiguous nouns

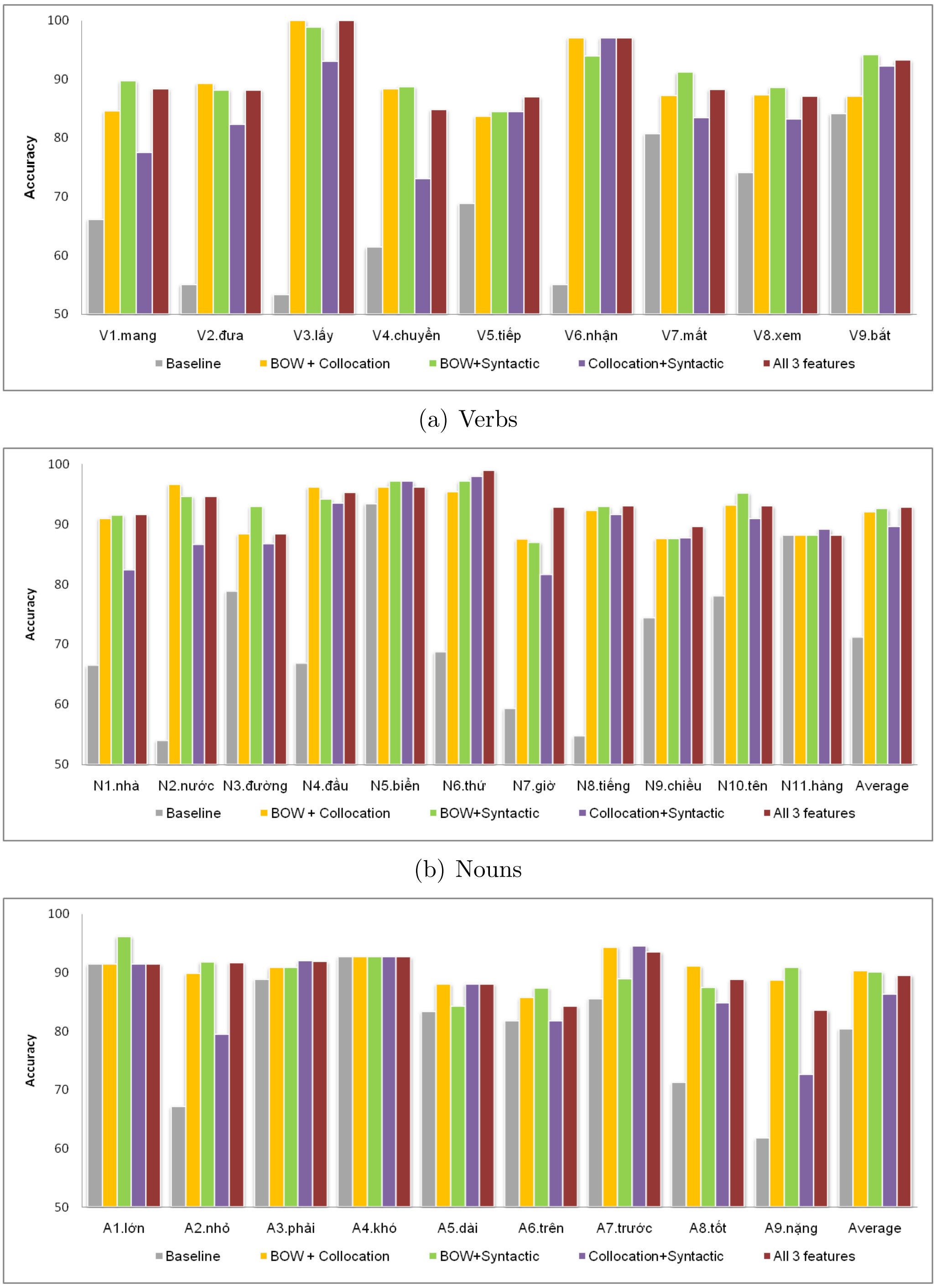
Target word	Baseline	BOW + Collocation	BOW + Syntactic	Collocation + Syntactic	All 3 features
N1.nhà	66.49	91.03	91.56	82.4	**91.61**
N2.nước	54	**96.67**	94.66	86.7	94.62
N3.đường	78.84	88.4	**93.01**	86.79	88.4
N4.đầu	66.82	**96.18**	94.27	93.55	95.36
N5.biển	93.46	96.27	**97.18**	**97.18**	96.27
N6.thứ	68.7	95.44	97.17	98	**99**
N7.giờ	59.33	87.5	87	81.67	**92.83**
N8.tiếng	54.68	92.33	92.95	91.71	**93.05**
N9.chiều	74.44	87.61	87.61	87.81	**89.61**
N10.tên	78.1	93.25	**95.25**	91.03	93.05
N11.hàng	88.18	88.18	88.18	**89.18**	88.18
Average	71.18	92.08 (+20.9)	92.62 (+21.44)	89.64 (+18.46)	**92.91 (+21.73)**

Table 5.15: Accuracy of feature combinations for ambiguous adjectives

Target word	Baseline	BOW + Collocation	BOW + Syntactic	Collocation + Syntactic	All 3 features
A1.lớn	91.44	91.44	**96.17**	91.44	91.44
A2.nhỏ	67.12	89.79	**91.77**	79.42	91.7
A3.phải	88.85	90.85	90.85	**91.96**	91.85
A4.khó	**92.64**	**92.64**	**92.64**	**92.64**	**92.64**
A5.dài	83.31	**88.03**	84.28	**88.03**	**88.03**
A6.trên	81.78	85.71	**87.38**	81.78	84.28
A7.trước	85.54	94.35	88.96	**94.52**	93.52
A8.tốt	71.19	**91.13**	87.48	84.8	88.77
A9.nặng	61.72	88.67	**90.9**	72.57	83.57
Average	80.4	**90.29 (+9.89)**	90.05 (+9.65)	86.35 (+5.95)	89.53 (+9.13)

Table 5.16: Average accuracy of feature combinations for verbs, nouns, adjectives and all target words

Target word	Baseline	BOW + Collocation	BOW + Syntactic	Collocation + Syntactic	All 3 features
Verb	66.52	89.4 (+22.88)	**90.88 (+24.36)**	85.16 (+18.63)	90.46 (+23.93)
Noun	71.18	92.08 (+20.9)	92.62 (+21.44)	89.64 (+18.46)	**92.91 (+21.73)**
Adjectives	80.4	**90.29 (+9.89)**	90.05 (+9.65)	86.35 (+5.95)	89.53 (+9.13)
All words	72.7	90.59 (+17.89)	**91.18 (+18.48)**	87.05 (+14.35)	90.97 (+18.27)



(a) Verbs

(b) Nouns

(c) Adjectives

Figure 5.7: Accuracy of feature combinations for target words

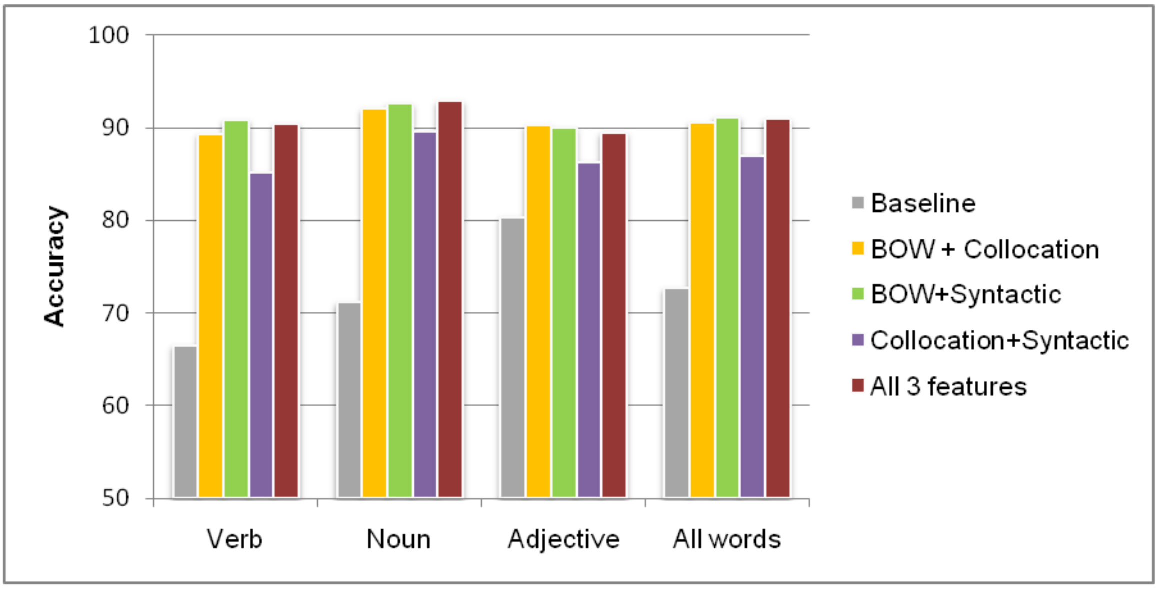


Figure 5.8: Average accuracy on feature combinations for target words

5.3.3 Discussion

In the first set of experiments (Subsection 5.3.1), all WSD classifiers of individual features performed well. All of them are better than the baseline method. For almost all words, BOW outperformed the other three features. As we discussed in Subsection 5.2.3, BOW is the best feature for training WSD classifiers since it contains the most lexical information around the target word.

On the other hand, the results of POS feature are always the lowest in comparison with the others, even with the baseline. We can find cases where accuracy of POS classifier is lower than the baseline for all POS categories of target words, for example, V9.xem, V10.bắt, N11.hàng, A5.dài, A7.trước and A8.tốt.

There are several cases that Collocation feature gave equal or higher accuracies than BOW, such as in V5.tiếp, V6.nhận, A5.dài, A7.trước and A8.tốt. In those cases, maybe just 4 words around the target word are effective rather than all words in the sentence without any collocation. However, in overall, Collocation couldn't outperform BOW feature.

Syntactic feature is not so effective for adjectives since we only use 4 syntactic relations for an adjective. Hence, there is not much contextual information for training SVM classifiers. However, Syntactic feature works well on verbs and nouns.

In average, when applying individual feature in Vietnamese WSD, BOW is the most effective feature, following by Collocation, Syntactic and POS feature.

In the second set of experiments (Subsection 5.3.2), WSD classifiers with combined features gave much higher results compared to individual features for all verbs and nouns. All systems got over baseline accuracies. The most effective feature combination is BOW + Syntactic for verbs, BOW + Collocation for adjectives and all 3 features for nouns.

Table 5.17: List of feature types

ID	Features
1	BOW
2	POS
3	Collocation
4	Syntactic
5	BOW + Collocation
6	BOW + Syntactic
7	Collocation + Syntactic
8	BOW + Collocation + Syntactic

The combination without BOW is the worst effective since it doesn't take the advantage of wide range lexical information around the target word as BOW does. On the other hand, the combinations with BOW increase the importance of the contextual words which have syntactic relations to the target word, or consider the word order, together with rich lexical information in the whole sentence. However, all 3 feature types combination couldn't outperform the combination with just 2 features (BOW + Collocation or BOW + Syntactic) in some cases of verbs and nouns, such as V1.mang, V2.đưa, V4.chuyển, V7.mất, V8.xem, V9.bắt, N2.nước, N3.đường, N4.đầu, N5.biển and N10.tên.

Similarily in PW task, seeing Table 5.13, 5.14 and 5.15, the best feature combinations also vary for individual target word. It might indicates that effective features or effective combination of features are different according to target words. It is desirable to automatically choose the best combination of features when a target word is given.

5.4 Results in Pseudoword and Real Word Task

This section reports results in PW-RW task. In this task, 8 feature sets shown in Table 5.17 are used for training WSD classifiers. The first four utilize one feature type (individual feature), while remains utilize two or three feature types (feature combination). Results are shown in Table 5.18 and 5.19. The bold number in each word indicates the best accuracy achieved for it. Figure 5.9 shows the average accuracies of verbs, nouns, adjectives as well as all target words.

Table 5.18: Accuracies in PW-RW task of individual features

Target word	Baseline 1	Baseline 2	BOW	POS	Collocation	Syntactic
V1.mang	66	66.12	**80**	69	67	63
V2.đưa	45	55.05	**78**	49	73	66
V3.lấy	46.51	53.34	**94.19**	53.49	58.14	58.14
V4.chuyển	38.46	61.43	**78.21**	51.28	47.44	52.56
V5.tiếp	68.29	68.83	**78.05**	43.9	68.29	63.41
V6.nhận	55	55.05	**73**	54	55	52
V7.mất	19.23	80.73	**87.08**	18.81	11.12	8.23
V8.xem	73.98	**74.07**	67.48	60.98	64.23	66.67
V9.bắt	83.84	**84.1**	83.84	78.79	79.8	88.89
Verb average	55.15	66.52	**79.98**	53.25	58.22	57.66
N1.nhà	33.59	66.49	**93.89**	48.09	43.51	46.56
N2.nước	54	54	**60.67**	36.67	47.33	36
N3.đường	21.26	78.84	**95.28**	48.82	38.58	51.18
N5.biển	93.14	**93.46**	84.31	76.47	93.14	81.37
N6.thứ	31.43	68.7	**89.52**	77.14	55.24	40.95
N7.giờ	59.26	59.33	**76.85**	67.59	65.74	76.85
N8.tiếng	74.23	54.68	**82.47**	72.16	72.16	78.35
N10.tên	22	78.1	**84**	52	29	61
N11.hàng	12.04	88.18	**94.44**	67.59	90.74	82.41
Noun average	44.55	71.31	**84.6**	60.73	59.49	61.63
A1.lớn	8.67	**91.44**	52	34	12.67	14
A2.nhỏ	33.02	67.12	**86.23**	50	38.68	37.74
A4.khó	7.79	**92.64**	85.71	58.44	46.75	25.97
A5.dài	17.05	83.31	**84.09**	59.09	76.14	71.59
A9.nặng	61.82	61.72	**67.27**	43.64	47.27	50.91
Adjective average	25.67	**79.25**	75.06	49.03	44.3	40.04
All words average	41.79	72.36	**79.88**	54.34	54	53.11

Table 5.19: Accuracies in PW-RW task of feature combinations

Target word	Baseline 1	Baseline 2	Unigram+ Collocation	Unigram +Syntactic	Collocation +Syntactic	All 3 features
V1.mang	66	66.12	80	**84**	66	**84**
V2.đưa	45	55.05	**76**	75	68	74
V3.lấy	46.51	53.34	**89.53**	84.88	56.98	87.21
V4.chuyển	38.46	61.43	79.49	79.49	56.41	**80.77**
V5.tiếp	68.29	68.83	**78.05**	75.61	68.29	73.17
V6.nhận	55	55.05	**74**	70	55	73
V7.mất	19.23	80.73	**86.12**	76.5	11.12	77.46
V8.xem	73.98	**74.07**	68.29	72.36	68.29	73.17
V9.bắt	83.84	84.1	**86.87**	88.89	85.86	83.84
Verb average	55.15	66.52	**79.82**	78.53	59.55	78.51
N1.nhà	33.59	66.49	**90.93**	**90.93**	48.18	90.17
N2.nước	54	54	**70.67**	60	37.33	69.33
N3.đường	21.26	78.84	88.29	**89.86**	42.62	88.29
N5.biển	93.14	**93.46**	85.29	85.29	85.29	85.29
N6.thứ	31.43	68.7	**92.38**	81.9	53.33	85.71
N7.giờ	59.26	59.33	74.07	**84.26**	77.78	80.56
N8.tiếng	74.23	54.68	80.41	87.63	77.32	**88.66**
N10.tên	22	78.1	**83**	59	64	59
N11.hàng	12.04	**88.18**	87.44	82.81	68.93	82.81
Noun average	44.55	71.31	**83.61**	80.19	61.64	81.09
A1.lớn	8.67	**91.44**	48	60.67	54	62
A2.nhỏ	33.02	67.12	**89.34**	88.35	35.85	86.62
A4.khó	7.79	**92.64**	88.31	79.22	62.34	83.12
A5.dài	17.05	83.31	82.95	79.55	68.18	**84.09**
A9.nặng	61.82	61.72	**70.91**	61.82	40	61.82
Adjective average	25.67	**79.25**	75.90	73.92	52.07	75.53
All words average	41.79	72.36	**79.78**	77.55	57.75	78.38

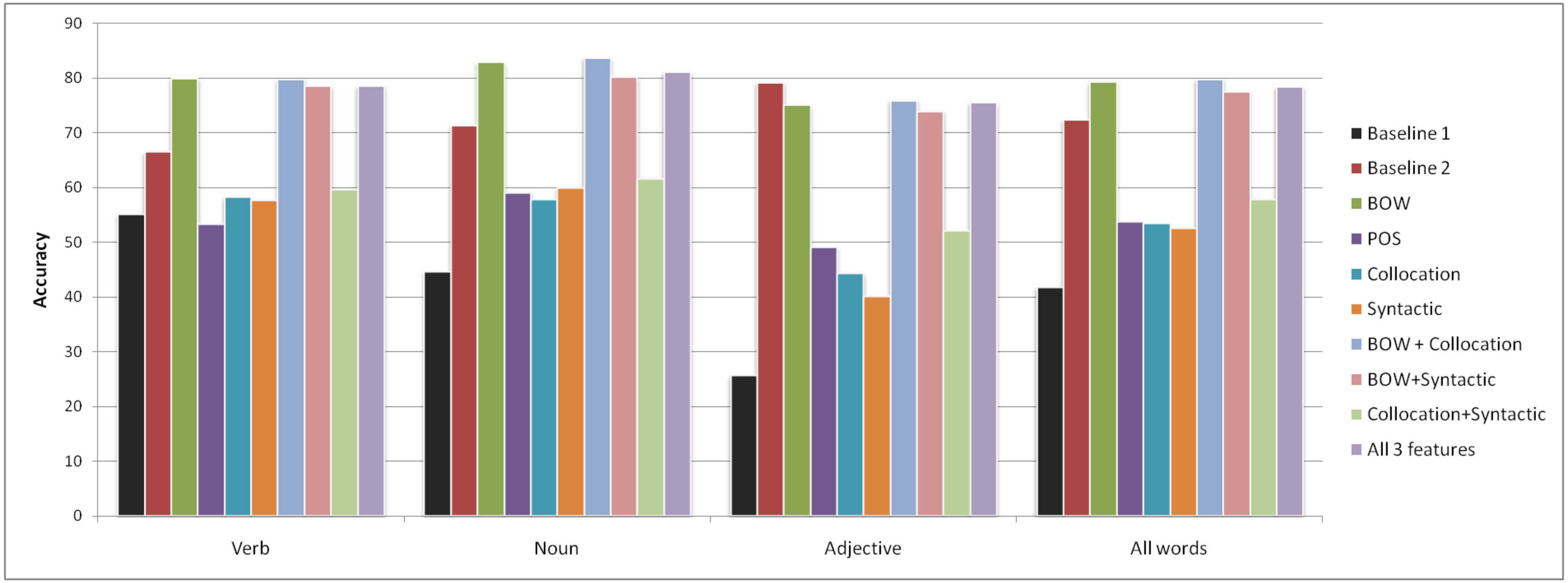


Figure 5.9: Average accuracy on 8 feature types for verb, noun and adjective

Comparing results in RW task (shown in Table 5.9, 5.10, 5.11 and 5.18) and PW-RW task (shown in Table 5.13, 5.14, 5.15 and 5.19), in overall, accuracies of WSD systems in RW-PW task were worse than that in RW task. Average results of three tasks on 8 feature sets shown in table 5.17 are drawn in charts in Figure 5.10. The figure clearly shows that PW-RW results is worse than RW results in all feature sets. It indicates that it is not suitable to use pseudoword technique to train WSD classifiers for real words.

It seems that WSD classifiers trained from PW corpus could not be appropriate to apply to real WSD in Vietnamese, although two words of pseudo-senses are not randomly chosen but related with real senses. The first reason is that pseudowords are not actually real words, so there are certain differences among extracted features from PW corpus and features from RW corpus. The second reason is that the most frequent sense of pseudowords in some case totally different with the real most frequent sense. This can be empirically observed by seeing that there are great gaps between Baseline 1 and Baseline 2 for some target words in Table 5.19. Therefore, the training data for the least frequent sense in PW corpus couldn't learn the behavior of that sense in the RW corpus (which is the most frequent sense indeed).

However, in Figure 5.9, there are some feature sets achieved higher accuracies than the baseline 2 for verbs and nouns. Therefore, although pseudoword technique is still not comparable to SVM classifiers trained from RW corpus, its results are acceptable to be applied in WSD task for verb and noun when a sense tagged corpus is not available. It would be interesting to verify that applying pseudoword technique in WSD is better than unsupervised WSD or not. This is one of our future work.

Next, we compare results on PW and RW task in average. Table 5.20 summarizes the best feature sets for each category of target word in PW and RW task. Table 5.20(a) shows the best individual feature for verbs, nouns and adjectives. All are same in PW and RW task, which means that pseudoword technique is appropriate to choose the best individual features in average. On the other hand, the best combined feature sets for verbs, nouns and adjectives shown in Table 5.20(b) are totally different with each other. Therefore, pseudoword technique might be inappropriate to choose the best combination of features for verbs, nouns and adjectives.

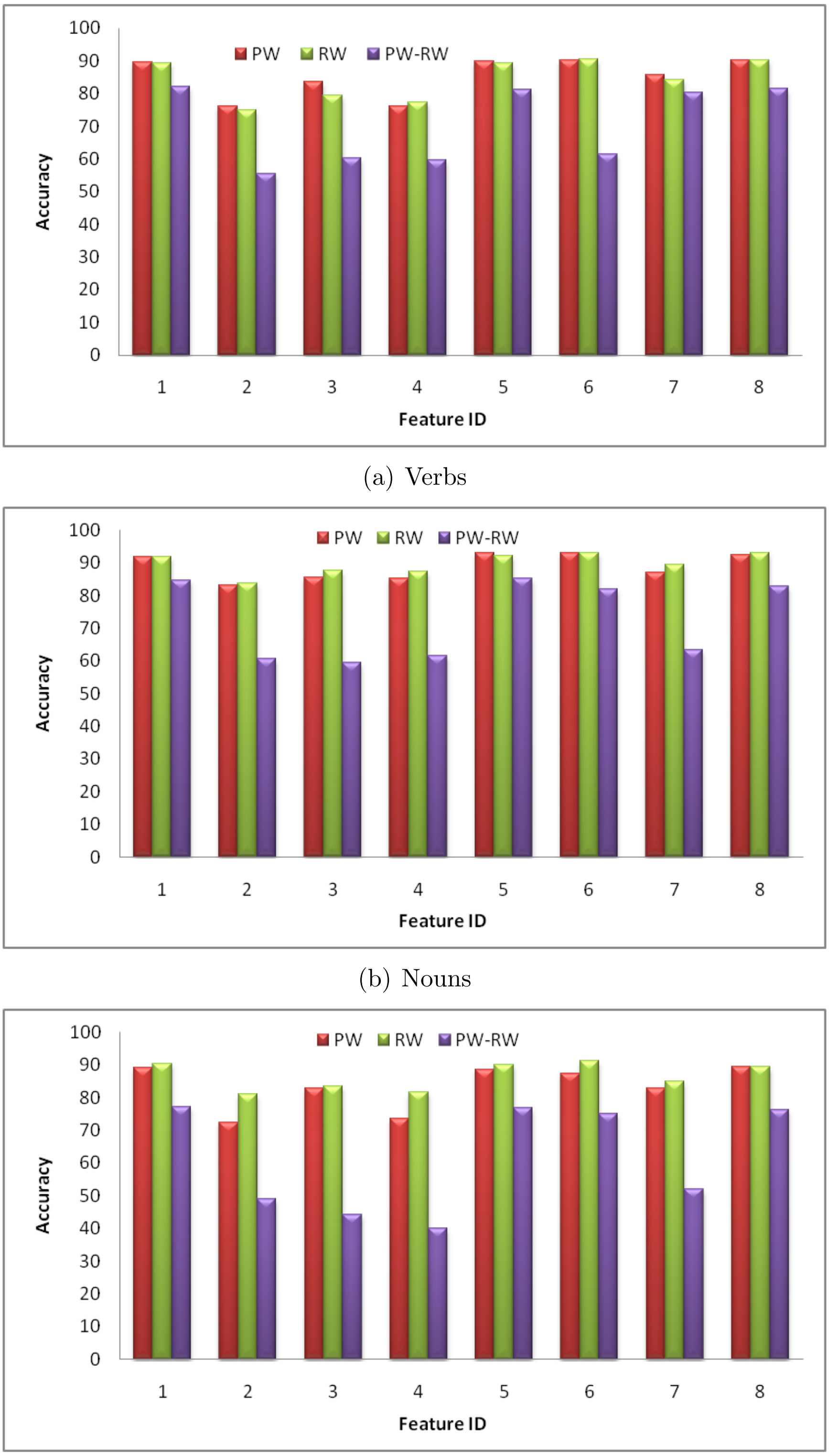
Table 5.20: Best feature sets for verbs, nouns and adjectives in average

(a) Individual feature

	PW	RW
V	BOW	BOW
N	BOW	BOW
A	BOW	BOW

(b) Feature combinations

	PW	RW
V	All 3 features	BOW+Syntactic
N	BOW+Syntactic	All 3 features
A	All 3 features	BOW + Collocation



(a) Verbs

(b) Nouns

(c) Adjectives

Figure 5.10: Average results of three tasks on different feature sets for verb, noun and adjective

Table 5.21, 5.22 and 5.23 show the orders of average accuracies of individual features and feature combinations in three tasks for verb, noun and adjective, respectively. The bold numbers indicate the cases where the ranks of feature set in PW and RW task are same. All ranks of individual features are same in PW and RW, which means that pseudoword technique is good for evaluating the effectiveness of individual features. However, for feature combination, the order of effectiveness of feature combinations in PW task is rather different with that in RW task. Therefore, pseudoword technique might be inappropriate to evaluate the effectiveness of feature combinations.

Table 5.21: Orders of average accuracies of different feature sets for verbs

(a) Individual feature

Task	BOW	POS	Collocation	Syntactic
PW	**1**	**4**	**2**	**3**
RW	**1**	**4**	**2**	**3**

(b) Feature combinations

Task	BOW+Collocation	BOW+Syntactic	Collocation+Syntactic	All 3 features
PW	**3**	2	**4**	1
RW	**3**	1	**4**	2

Table 5.22: Orders of average accuracies of different feature sets for nouns

(a) Individual feature

Task	BOW	POS	Collocation	Syntactic
PW	**1**	**4**	**2**	**3**
RW	**1**	**4**	**2**	**3**

(b) Feature combinations

Task	BOW+Collocation	BOW+Syntactic	Collocation+Syntactic	All 3 features
PW	2	1	**4**	3
RW	3	2	**4**	1

Table 5.23: Orders of average accuracies of different feature sets for adjectives

(a) Individual feature

Task	BOW	POS	Collocation	Syntactic
PW	**1**	**4**	**2**	**3**
RW	**1**	**4**	**2**	**3**

(b) Feature combinations

Task	BOW+Collocation	BOW+Syntactic	Collocation+Syntactic	All 3 features
PW	**2**	3	**4**	1
RW	**2**	1	**4**	3

Next, we compare the best feature set in PW and RW task for each target word. Table 5.4 reveals the number of target word where the best (or one of the best) feature set are same in PW and RW tasks. The 'Individual' column indicates the case that the best individual feature sets are same, while 'Combined' column indicates the case of combined feature sets. It is shown that the number of target words where the best individual feature or feature combination in PW task agreed with those in RW task is not much for noun, whereas it is high for verb and adjective. So, pseudoword technique is inappropriate to choose the best combination of features when the target word is noun. Otherwise, it is rather appropriate to choose the best individual features and the best combination for each target word in verb and adjective categories.

Table 5.24: The best feature comparison for each target word

POS	Number of pseudowords	Individual	Combined
V	9	7	6
N	9	5	2
A	5	4	4

The reason why there are too few target nouns sharing the best feature sets in PW and RW tasks might be because nouns are used in a wide range of domains compared to verbs and adjectives in the corpus. For example, in Table 4.4, the first sense of V4.chuyển is '*to send*'. This sense can only be used in the text related to email, postcard or document. Similarly, in Table 4.6, the second sense of A9.nặng is '*serious*'. This sense can only be used in a context that has topic about health and disease. On the other hand, domains for using nouns are very large. For example, in Table 4.5, the second sense of the word N7.giờ is '*now*'. This sense can be used in various topics, such as sports, news, literature, etc. However, since the corpus is small, its pseudoword cannot cover all possible contexts in which the real word migh appears.

To sum, on average, pseudoword technique is appropriate to choose the best individual features but inappropriate for feature combinations. For each target word, pseudoword technique is only appropriate to choose the best individual features and the best combination when the target word is not a noun but verb or adjective.

Chapter 6

Conclusion

6.1 Contribution

WSD is an important task in natural language processing. This research aims to explore effective features for Vietnamese WSD. Since WSD can be considered as a classification task, we applied SVM, a machine learning technique, to find which features are important in Vietnamese WSD. Two corpora have been built for training and test:

1. RW corpus in which around 3000 sentences of ambiguous words are manually senses tagged.

2. PW corpus in which about 2500 samples of pseudo-senses are automatically collected.

We have experimented three tasks to evaluate the effectiveness of each individual features and feature combinations. We compared the results of experiments with and without a sense tagged corpus and found that PW task gave result similar to RW task in some cases. This result shows that pseudoword technique is a potential technique for exploring effective features for Vietnamese WSD, especially for verbs and adjectives.

On the other hand, we also discovered that pseudoword technique is not good in comparison with SVM classifiers trained from RW corpus. However, the results of applying pseudoword technique to train a real WSD for verbs and nouns are mostly higher than the baseline. It might indicates that pseudoword technique could be applied to disambiguate real ambiguous verbs and nouns when there is no sense tagged corpus available.

6.2 Future Work

There are some disadvantages in this research. For example, the data sparseness is problematic for training classification models, and the assumption of two senses per target word may not be realistic. Therefore, it is interesting to investigate the effective features for WSD multi-class classifiers in accompany with increasing the corpus size and the number of senses to be disambiguated.

Besides, through the experiments, we see that the best feature combinations vary for individual target word. It might indicates that effective features or effective combination of features are different according to target words. In future, we would like to try choosing the best combination of features automatically when a target word is given.

Since our experiment showed that pseudoword technique is not as good as the ordinary supervised corpus based method but still higher than the most frequency baseline, it would be interesting if we could verify that applying pseudoword technique in WSD is better than unsupervised or not.

Another interesting work is comparing the effective features between Vietnamese WSD and English WSD to explore the differences and similarity between these languages in WSD task.

Appendix A

Algorithm for Syntactic Relation Extraction

A.1 Syntactic Relation for Verb

We extracted 7 syntactic relations for verb.

Let w be the target word, whose POS tag is V or V-H (H indicates a head word of a phrase) in a syntactic tree.

Let T be the syntactic tree of the sentence including w.

The algorithms to find syntactic relational word with w in T are described below.

Algorithm 1: Extract Subj-N

```
Input: $T$: syntactic tree, $N_{VP}$: a $VP$ node in $T$ that contains $w$
Output: the noun which is the subject of $w$
$N_{Curr} \Leftarrow N_{VP}$ ;                                   // VP is current node
while $N_{Curr}$ does not yet reach $ROOT_T$ do
    foreach Sibling $X$ of $N_{Curr}$ do
        if non-terminal symbol of $X$ is $NP$-$SUB$ then
            $N_t \Leftarrow X$
            return Head word of $N_t$
        else
            $N_{Curr} \Leftarrow (N_{Curr} \rightarrow Parent)$
return $NULL$
```

Algorithm 2: Extract DOB-N

```
Input: $T$: syntactic tree, $N_{VP}$: a $VP$ node in $T$ that contains $w$
Output: the noun which is the direct object of $w$
$N_{Curr} \Leftarrow N_{VP}$ ;                                // VP is current node
foreach $Child\ X\ of\ N_{Curr}$ do
    if $non\text{-}terminal\ symbol\ of\ X\ is\ NP\text{-}DOB$ then
        $N_t \Leftarrow X$
        return $Head\ word\ of\ N_t$
return $NULL$
```

Algorithm 3: Extract IOB-N

```
Input: $T$: syntactic tree, $N_{VP}$: a $VP$ node in $T$ that contains $w$
Output: the noun which is the indirect object of $w$
$N_{Curr} \Leftarrow N_{VP}$ ;                                // VP is current node
foreach $Child\ X\ of\ N_{Curr}$ do
    if $non\text{-}terminal\ symbol\ X\ is\ NP\text{-}IOB$ then
        $N_t \Leftarrow X$
        return $Head\ word\ of\ N_t$
return $NULL$
```

Algorithm 4: Extract Head-V

```
Input: $T$: syntactic tree, $N_{VP}$: a $VP$ node in $T$ that contains $w$
Output: the verb which is the head of $w$
$N_{Curr} \Leftarrow (N_{VP} \rightarrow Parrent)$ ;           // VP is current node
foreach $Child\ X\ of\ N_{Curr}$ do
    if $X\ is\ pre\text{-}terminal\ of\ which\ POS\ is\ V\text{-}H\ and\ X \neq N_{VP}$ then
        return $word\ of\ X$
return $NULL$
```

Algorithm 5: Extract Mod-V

```
Input: $T$: syntactic tree, $N_{VP}$: a $VP$ node in $T$ that contains $w$
Output: the verb that modifies $w$
$N_{Curr} \Leftarrow N_{VP}$ ;                                // VP is current node
foreach $Child\ X\ of\ N_{Curr}$ do
    if $non\text{-}terminal\ symbol\ of\ X\ is\ VP$ then
        $N_t \Leftarrow X$
        return $Head\ word\ of\ N_t$
return $NULL$
```

Algorithm 6: Extract Mod-A

```
Input: $T$: syntactic tree, $N_{VP}$: a $VP$ node in $T$ that contains $w$
Output: the adjective that modifies $w$
$N_{Curr} \Leftarrow N_{VP}$ ;                                  // VP is current node
foreach Child $X$ of $N_{Curr}$ do
    if non-terminal symbol of $X$ is $AP$ then
        $N_t \Leftarrow X$
        return Head word of $N_t$
return NULL
```

Algorithm 7: Extract Mod-P

```
Input: $T$: syntactic tree, $N_{VP}$: a $VP$ node in $T$ that contains $w$
Output: the preposition that modifies $w$
$N_{Curr} \Leftarrow N_{VP}$ ;                                  // VP is current node
foreach Child $X$ of $N_{Curr}$ do
    if non-terminal symbol of $X$ is $PP$ then
        $N_t \Leftarrow X$
        return Head word of $N_t$
return NULL
```

The last step of the above algorithms is to extract a head of a phrase N_t. This is easy since head words of phrases are distinguished by attaching '-H' to their POSs (such as V-H, N-H) in Vietnamese Treebank.

A.2 Syntactic Relation for Noun

We extracted 7 syntactic relations for noun.

Let w be the target word where POS is *N* or *N-H* (*H* indicates a head of a phrase) in a syntactic tree.

Let T be the syntactic tree of the sentence including w.

The algorithms to find syntactic relational word with w in T are described below.

Algorithm 8: Extract OB-V

```
Input: $T$: syntactic tree, $N_{NP}$: a $NP$-$DOB$ or $NP$-$IOB$ node in $T$ that contains $w$
Output: the verb that is modified by $w$ where $w$ is its object
$N_{Curr} \Leftarrow (N_{NP} \rightarrow Parent)$ // Find VP node dominating NP-DOB or NP-IOB
foreach Child $X$ of $N_{Curr}$ do
    if $X$ is pre-terminal of which POS is $V$-$H$ then
        return word of $X$
return NULL
```

Algorithm 9: Extract Head-N

```
Input: $T$: syntactic tree, $N_{NP}$: a $NP$ node in $T$ that contains $w$
Output: the noun that is head of $w$
$N_{Curr} \Leftarrow (N_{NP} \rightarrow Parent)$ ;                  // Find NP node dominating $N_{NP}$
foreach Child $X$ of $N_{Curr}$ do
    if $X$ is pre-terminal of which POS is N-H and $X \neq N_{NP}$ then
        return word of $X$
return NULL
```

Algorithm 10: Extract Mod-A

```
Input: $T$: syntactic tree, $N_{NP}$: a $NP$ node in $T$ that contains $w$
Output: the adjective that modifies $w$
$N_{Curr} \Leftarrow N_{NP}$
foreach Child $X$ of $N_{Curr}$ do
    if non-terminal symbol of $X$ is AP then
        $N_t \Leftarrow X$
        return Head word of $N_t$
return NULL
```

Algorithm 11: Extract Mod-N

```
Input: $T$: syntactic tree, $N_{NP}$: a $NP$ node in $T$ that contains $w$
Output: the noun that modifies $w$
$N_{Curr} \Leftarrow N_{NP}$
foreach Child $X$ of $N_{Curr}$ do
    if non-terminal symbol of $X$ is $NP$ then
        $N_t \Leftarrow X$
        return Head word of $N_t$
return NULL
```

Algorithm 12: Extract Mod-P

```
Input: $T$: syntactic tree, $N_{NP}$: a $NP$ node in $T$ that contains $w$
Output: the preposition that modifies $w$
$N_{Curr} \Leftarrow N_{NP}$
foreach Child $X$ of $N_{Curr}$ do
    if non-terminal symbol of $X$ is $PP$ then
        $N_t \Leftarrow X$
        return Head word of $N_t$
return NULL
```

Algorithm 13: Extract Subj-Verb

```
Input: $T$: syntactic tree, $N_{NP}$: a $NP\text{-}SUB$ node in $T$ that contains $w$
Output: the predicate verb of $w$ when $w$ is a subject
$N_{Curr} \Leftarrow (N_{NP} \rightarrow Parent)$
foreach Child $X$ of $N_{Curr}$ do
    if non-terminal symbol of $X$ is $VP$ then
        $N_t \Leftarrow X$;                          // Find a VP sibling of NP-SUB
        return Head word of $N_t$
return NULL
```

The last step of the above algorithms is to extract a head of a phrase N_t. This is easy since head words of phrases are distinguished by attaching '-H' to their POSs (such as V-H, N-H) in Vietnamese Treebank.

A.3 Syntactic Relation for Adjective

We extracted 4 syntactic relations for adjective.

Let w be the target word whose POS tag is A or A-H (H indicates a head of a phrase) in a syntactic tree.

Let T be the syntactic tree of the sentence including w.

The algorithms to find syntactic relational word with w in T are described below.

Algorithm 14: Extract Subj-N

```
Input: $T$: syntactic tree, $N_{AP}$: a $AP\text{-}PRD$ node in $T$ that contains $w$
Output: the noun that is subject of $w$ where $w$ is a predicate
$N_{Curr} \Leftarrow N_{AP}$
while $N_{Curr}$ does not yet reach $ROOT_T$ do
    foreach Sibling $X$ of $N_{Curr}$ do
        if non-terminal symbol of $X$ is $NP\text{-}SUB$ then
            $N_t \Leftarrow X$
            return Head word of $N_t$
        else
            $N_{Curr} \Leftarrow (N_{Curr} \rightarrow Parent)$
return NULL
```

Algorithm 15: Extract S-V

```
Input: $T$: syntactic tree, $N_{AP}$: a $AP\text{-}SUB$ node in $T$ that contains $w$
Output: the predicative verb of $w$ where $w$ is a subject
$N_{Curr} \Leftarrow (N_{AP} \rightarrow Parent)$
foreach $Child\ X\ of\ N_{Curr}$ do
    if non-terminal symbol of $X$ is $VP$ then
        $N_t \Leftarrow X$;                       // Find a VP sibling of AP-SUB
        return Head word of $N_t$
return NULL
```

Algorithm 16: Extract Head-V

```
Input: $T$: syntactic tree, $N_{AP}$: a $AP$ node in $T$ that contains $w$
Output: the verb that is modified by $w$
$N_{Curr} \Leftarrow (N_{AP} \rightarrow Parent)$ ;          // Find VP node dominating $N_{AP}$
foreach $Child\ X\ of\ N_{Curr}$ do
    if $X$ is pre-terminal of which POS is $V\text{-}H$ then
        return word of $X$
return NULL
```

Algorithm 17: Extract Head-N

```
Input: $T$: syntactic tree, $N_{AP}$: a $AP$ node in $T$ that contains $w$
Output: the noun that is modified by $w$
$N_{Curr} \Leftarrow (N_{AP} \rightarrow Parent)$ ;          // Find NP node dominating $N_{AP}$
foreach $Child\ X\ of\ N_{Curr}$ do
    if $X$ is a pre-terminal of which POS is $N\text{-}H$ then
        return word of $X$
return NULL
```

The last step of the above algorithms is to extract a head of a phrase N_t. This is easy since head words of phrases are distinguished by attaching '-H' to their POSs (such as V-H, N-H) in Vietnamese Treebank.

References

[1] E. Agirre and D. Martinez. Learning class-to-class selectional preferences. In *ConLL '01: Proceedings of the 2001 workshop on Computational Natural Language Learning*, pages 1–8, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[2] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proceedings of the 16th conference on Computational linguistics*, pages 16–22, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[3] C. Cardie. Empirical methods in information extraction. *AI Magazine*, 18:65–79, 1997.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] E. Charniak. Statistical techniques for natural language parsing. *AI Magazine*, 18:33–44, 1997.

[6] H. T. Dang, C. Y. Chia, M. Palmer, and F. Chiou. Simple features for chinese word sense disambiguation. *Proceedings of the 19th International Conference On Computational Linguistics*, 1:1–7, 2002.

[7] D. Dinh. Building a training corpus for word sense disambiguation in english-to-vietnamese machine translation. In *COLING-02 on Machine translation in Asia*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[8] W. A. Gale, K. W. Church, and D. Yarowsky. One sense per discourse. In *HLT '91: Proceedings of the workshop on Speech and Natural Language*, pages 233–237, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[9] W. A. Gale, K. W. Church, and D. Yarowsky. Work on statistical methods for word sense disambiguation. *AAAI Fall Symposium Series, Probabilistic Approaches to Natural Language*, pages 54–60, 1992.

[10] N. Ide and J. Veronis. Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, 24:2–40, 1998.

[11] A. Kaplan. An experiment study of ambiguity and context. *Mechanical Translation*, 2:39–46, 1955.

[12] K. Knight. Automating knowledge acquisition for machine translation. *AI Magazine*, 18:81–96, 1997.

[13] Y. K. Lee and H. T. Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 10:41–48, 2002.

[14] Lesk and Michael. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, New York, NY, USA, 1986. ACM.

[15] Z. Lu, H. Wang, J. Yao, T. Liu, and S. Li. An equivalent pseudoword solution to chinese word sense disambiguation. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 457–464, Morristown, NJ, USA, 2006. Association for Computational Linguistics.