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

Title	Study on Supervised Learning of Vietnamese Word Sense Disambiguation Classifiers
Author(s)	Nguyen, Minh Hai; Shirai, Kiyoaki
Citation	Journal of Natural Language Processing, 19(1): 25-50
Issue Date	2012-03
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/10658
Rights	Copyright (C) 2012 The Association for Natural Language Processing. Minh Hai Nguyen, Kiyoaki Shirai, Journal of Natural Language Processing, 19(1), 2012, 25-50.
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Study on Supervised Learning of Vietnamese Word Sense Disambiguation Classifiers

Minh Hai Nguyen[†] and Kiyoaki Shirai[††]

It is said that Vietnamese is a language with highly ambiguous words. However, there has been no published Word Sense Disambiguation (WSD hereafter) research on this language. This current research is the first attempt to study Vietnamese WSD. Especially, we would like to explore the effective features for training WSD classifiers and verify the applicability of the 'pseudoword' technique to both investigating effectiveness of features and training WSD classifiers. Three tasks have been conducted, using two corpora which were built manually based on Vietnamese Treebank and automatically by applying pseudowords technique. Experiment results showed that Bag-Of-Word feature performs well for all three categories of words (verbs, nouns, and adjectives). However, its combination with POS, Collocation or Syntactic features can not significantly improve the performance of WSD classifiers. Moreover, the experiment results confirmed that pseudoword is a suitable technique to explore the effectiveness of features in disambiguation of Vietnamese verbs and adjectives. Furthermore, we empirically evaluated the applicability of the pseudoword technique as an unsupervised learning method for real Vietnamese WSD.

Key Words: *Word Sense Disambiguation, Vietnamese, Supervised Machine Learning, Feature for WSD, Pseudoword*

1 Introduction

WSD plays an important role in natural language processing applications, such as machine translation, information retrieval, speech processing, etc. So far, this problem has been studied for English, Japanese and many other languages for more than half a century, and many effective knowledge sources as well as disambiguation methods have been discovered. Vietnamese is said to be a language including many highly ambiguous words. For example, the word 'bien' 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, to the best of our knowledge, there is no research on Vietnamese WSD. Vietnamese is an isolating language with some general characteristics as follows:

- Words do not have morphological forms. Vietnamese has a number of tense markers to

[†] School of Information Science, Japan Advanced Institute of Science and Technology. nhminh@jaist.ac.jp
[††] School of Information Science, Japan Advanced Institute of Science and Technology. kshirai@jaist.ac.jp

indicate the tense of a sentence. Therefore, the grammatical relationship is expressed by word order and auxiliary words.

- Word boundary is not obviously determined by blank.
- There are many 'classifiers' which come before nouns like Chinese.
- Vietnamese also has the same basic SVO word order as English.

In this study, one of our goals is to carry out the first attempt to establish a WSD method for Vietnamese. Since approaches based on supervised machine learning achieved great success in WSD, the present authors are also interested in it. Especially, this paper will discuss the following two issues:

- What are effective features in Vietnamese WSD?
  Various types of features for WSD were proposed in previous work. Our question here is, "What kinds of features are effective for disambiguation of word senses in Vietnamese?"
- Is pseudoword technique applicable for Vietnamese WSD?
  For supervised learning of WSD classifiers, a sense-tagged corpus is required as training data. However, there is no Vietnamese sense-tagged corpus available to the public. Pseudoword technique is often used to evaluate supervised WSD methods when no training data is available. Two words w_1 and w_2 are regarded as an imaginary word (pseudoword) p, then machine learning methods are applied to train classifiers which predict if the original word of p in texts is w_1 or w_2. The performance of trained classifiers can be evaluated without heavy human intervention. Our interest is whether the pseudoword technique is useful for Vietnamese WSD or not.

Considering the above issues, this paper has three goals. The first one is to empirically explore effective features for Vietnamese WSD. Supervised WSD classifiers with several kinds of features are trained, then their performance is compared. Effectiveness of feature combination is also considered. The second is to check the applicability of the pseudoword technique. This paper will investigate the possibility of the pseudoword technique for finding the most effective features. The last goal is, as an alternative to unsupervised methods, we explore a method to apply the pseudoword technique for training WSD classifiers when no sense-tagged corpus is available.

In the next section, we will discuss some work related to our research. Then, we describe the development of our system for Vietnamese WSD in Section 3. Section 4 introduces three tasks which were conducted in this research. Section 5 shows results and some discussion. Finally, we summarize the research and indicate future work in Section 7.

2 Related work

The first experiment by Kaplan proved that just one or two words on both sides of an ambiguous word can be evidence to disambiguate that word (Kaplan 1955). Later, more useful information from context was discovered by numerous works in WSD. Yarowsky introduced simple set of features (context around the ambiguous words) in accent restoration task (Yarowsky 1994). This led to many other improved sets of features, such as syntactic dependencies (Martínez, Agirre, and Màrquez 2002; Dang, Chia, Palmer, and Chiou 2002; Yarowsky and Florian 2002), or cross language evidence (Gale, Church, and Yarowsky 1992a). Beside the approaches utilizing the evidence provided by the surrounding context of the ambiguous word, there are many other researches which take advantage of knowledge bases without using any corpus evidence, such as approaches using dictionaries, thesauri, and lexical knowledge bases (Lesk and Michael 1986; Agirre and Martinez 2001). These knowledge sources have been used in various ways to improve WSD systems in English. Numerous studies have also been devoted to WSD in languages other than English. However, Vietnamese WSD has not been studied so far. Vietnamese is a language with characteristics different from those of English. For example, words in Vietnamese are not separated by empty spaces, an adjective can be a subject of a sentence, etc. It is necessary to investigate the effective features for Vietnamese WSD.

According to the knowledge sources used in sense disambiguation, methods in WSD are classified as knowledge-based, unsupervised corpus-based, supervised corpus-based, and combinations of these (Agirre and Edmonds 2006). Among these methods, the approach to supervised learning is the hot topic, since it has been 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 poses challenges to the supervised learning approach for WSD. For Vietnamese WSD, the problem is serious, since no sense-tagged corpus is available to the public. Dinh attempted to construct a sense-tagged corpus in Vietnamese by using English semantically-tagged corpus and bilingual English-Vietnamese texts (Dinh 2002). However, he mainly annotated English texts, in order to disambiguate English words to be applied in an English-Vietnamese machine translation system. And there was no evaluation of WSD based on his corpus, either.

Gale et al. introduced a technique called 'pseudowords' to overcome the obstacles of supervised methods (Gale, Church, and Yarowsky 1992b). However, two words to be combined as a pseudoword in Gale's experiments are randomly chosen. Thus pseudowords may have different linguistic characteristics from real ambiguous words. Lu et al. presented 'equivalent' pseudowords (Lu, Wang, Yao, Liu, and Li 2006), in which they built up pseudowords based on real ambigu-

ous words. However, they only performed evaluation on pseudowords, and have no comparison between pseudowords and real ambiguous words. The task of classifying two different words may be easier than distinguishing two senses of the same word. Therefore, our research aims to empirically evaluate the validity of the 'pseudoword' method for Vietnamese WSD.

3 Our method

In this section, we describe our method to disambiguate word senses. SVM is used as a machine learning algorithm which is introduced in Subsection 3.1. Features used in the SVM classifiers are also explained in Subsection 3.2.

3.1 Support Vector Machine as classifier for WSD

Support Vector Machine (SVM) (Corinna and Vladimir 1995) learns a linear discriminant hyperplane that separates two classes of data represented as high-dimensional vectors. In this research, the number of senses for an ambiguous word is limited to two, since it is rather difficult to prepare a large scale corpus covering all senses of an ambiguous word[1]. The linear kernel is used for training WSD classifiers, because in high dimensional space (when the number of features is large), we expect that mapping data to a higher dimensional space does not improve performance. We actually found that other kernels gave poorer results than linear kernel in our preliminary experiment.

3.2 Feature set

For each target instance w, we encode its surrounding context as a feature vector. The feature set F of w is denoted as in (1), where f_i represents a feature.

$$F = \{f_1, f_2, ..., f_n\} \tag{1}$$

In our experiment, the feature vector is weighted according to the context of target instances in the training corpus (Eq. (2)), where ω_i is a weight of f_i. Methods for defining f_i and ω_i will be described in detail for each type of feature.

$$\vec{f} = (\omega_1, \omega_2, ..., \omega_n) \tag{2}$$

[1] This assumption is obviously not realistic. However, we believe that results of the experiments reported in this paper would provide somewhat reliable information about Vietnamese WSD.

3.2.1 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,

$$\frac{Hoang\ hon}{(sunset; Noun)}\ \frac{tren}{(on; Preposition)}\ \frac{\boldsymbol{bien}}{(sea; Noun)}\ \frac{that}{(so; Adverb)}\ \frac{dep}{(beautiful; Adjective)}$$

the BOW of the target word '*bien*' is {*hoang hon, dep*}. Therefore, f_i corresponds to a word appearing in the context of a target word.

Function words[2], proper nouns, numbers and punctuation marks are not used as features, since they would not be effective clues for WSD. For BOW feature, F is a set of all possible words appearing in the context of target instances in the training corpus. For each sentence l containing a target instance w in the training corpus, f_i is weighted as in Eq. (3).

$$\omega_i = \begin{cases} t_i^1 & \text{if } f_i \text{ appears in } l \text{ and sense of } w \text{ is } s_1 \\ t_i^2 & \text{if } f_i \text{ appears in } l \text{ and sense of } w \text{ is } s_2 \\ 0 & \text{if } f_i \text{ does not appear in } l \end{cases} \tag{3}$$

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. While f_i is weighted as in Eq. (4) in the test data, since the sense of w is unknown[3].

$$\omega_i = \begin{cases} (t_i^1 + t_i^2)/2 & \text{if } f_i \text{ appears in } l \\ 0 & \text{if } f_i \text{ does not appear in } l \end{cases} \tag{4}$$

3.2.2 POS

This feature encodes part-of-speech of each word in a context window c around the target instance w as in Eq. (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 context after the target word. Similarly, p_i is negative for words in the context before the target word. If p_i exceeds the sentence boundary, P_i is denoted by the null symbol ϵ. For POS feature, F is a set of 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

[2] Function word is defined by POS. In our method, classifier, unit noun, pronoun, quantifier, adverb, preposition, connector, interjection, introductory word (a kind of particle), abbreviation and untagged word are regarded as function words.

[3] We also tried weighting both test and training data as in Eq. (4). However, the accuracy was 81.2, which was worse than our weighting method (94.0; the accuracy of the classifier with only BOW feature for all words shown in Table 4). In Eq. (3), association between a BOW feature and a sense is considered in the training phase. It seems useful to improve the accuracy.

corpus, f_i is weighted by ω_i as in Eq. (6). Note that POS categories used in our classifiers are coarse, such as A (Adjective), V (Verb), N (Noun) and E (Preposition).

$$f_i = (p_i, P_i) \tag{5}$$

$$\omega_i = \begin{cases} 1 & \text{if POS of the word at the position } p_i \text{ is } P_i; \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

3.2.3 Collocation

Collocation feature (COL feature hereafter) encodes a sequence of words (n-grams) that co-occurs with the target word. Let w_i denote the *i-th* word to the right (or left if i is negative) of the target instance w_0. If the *i-th* word exceeds the sentence boundary, $w_i = \epsilon$. A collocation string is defined as in Eq. (7).

$$C_{l,r} = w_l w_{l+1} ... w_r \tag{7}$$

For each target instance in the corpus, we extracted 9 collocation strings: $C_{-1,0}$; $C_{0,1}$; $C_{-2,0}$; $C_{-1,1}$; $C_{0,2}$; $C_{-3,0}$;$C_{-2,1}$; $C_{-1,2}$; $C_{0,3}$. Each feature f_i is extracted as in Eq. (8), where l_i and r_i are the start and end positions of a collocation string ($1 < r_i - l_i < 4, l_i = -3, ..., 0, r_i = 0, ..., 3$). Unlike the case of BOW, we do not remove punctuation symbols or numbers in the collocations. For the COL feature, F is a set of 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. (9).

$$f_i = (l_i, r_i, C_{l_i,r_i}) \tag{8}$$

$$\omega_i = \begin{cases} 1 & \text{if } C_{l_i,r_i} \text{ is found in } l; \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

3.2.4 Syntactic

Syntactic relations can be extracted from an annotated syntactic tree, such as subject-verb, verb-object, etc. In this paper, target words are supposed to verbs, nouns or adjectives. For each category of target word, we used different features according to Vietnamese grammar. Since characteristics of Vietnamese are different from English, the extracted features are not the same as in the previous approaches based on syntactic relations of English. For example, an adjective can be subject of a sentence in Vietnamese, while it is impossible in English. Table 1 shows the list of syntactic feature (SYN feature hereafter) used in our WSD classifiers. In Table 1, 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 a feature word.

Table 1 List of syntactic features.

Syntactic feature for verbs	
Subj-N	The word that is subject of the target verb w.
DOB-N	The direct object of w.
IOB-N	The indirect object of w.
Head-V	The verb that is modified by w.
Mod-V	The verb that modifies w.
Mod-A	The adjective that modifies w.
Mod-P	The preposition that modifies w.
Syntactic feature for nouns	
OB-V	The verb that is modified by the target noun w where w is its object.
Head-N	The noun that is a head of w.
Head-P	The head preposition of the prepositional phrase including w.
Mod-A	The adjective that modifies w.
Mod-N	The noun that modifies w.
Mod-P	The head preposition of the prepositional phrase that modifies w.
Subj-V	The predicative verb of w where w is a subject.
Syntactic feature for adjectives	
Subj-N	The subject of the target adjective w where w is a predicate.
S-V	The predicative verb of w where w is a subject.
Head-V	The verb that is modified by w.
Head-N	The noun that is modified by w.

The SYN feature vector is constructed in the same manner as in POS and Collocation features. 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 (10). For Syntactic feature, 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 target instance w in the corpus, f_i is weighted as in Eq. (11).

$$f_i = (sl_i, t_i) \tag{10}$$

$$\omega_i = \begin{cases} 1 & \text{if } w \text{ and } t_i \text{ are in the syntactic relation } sl_i \text{ in } l \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

In addition to 4 types of features, the feature combinations are considered as in Table 2. In feature combination, feature vectors for target instances are built by just concatenating vectors for individual features.

Table 2 Combined feature sets.

2-feature-combination
BOW+POS, BOW+COL, BOW+SYN, POS+COL, POS+SYN, COL+SYN (example of feature vector: $F_{combine} = \{F_{BOW}, F_{COL}\}$)
3-feature-combination
BOW+POS+COL, BOW+POS+SYN, BOW+COL+SYN, POS+COL+SYN (example of feature vector: $F_{combine} = \{F_{BOW}, F_{COL}, F_{SYN}\}$)
4-feature-combination
BOW+POS+COL+SYN (example of feature vector: $F_{combine} = \{F_{BOW}, F_{POS}, F_{COL}, F_{SYN}\}$)

4 Tasks

This section describes three tasks which were conducted to explore the effective features for learning Vietnamese WSD classifiers, as well as to evaluate pseudoword technique. Since there is no sense-tagged corpus for Vietnamese WSD, two kinds of sense-tagged corpora were built based on Vietnamese Treebank (Nguyen, Vu, Nguyen, Nguyen, and Le 2009), a corpus which contains around 10,000 sentences manually annotated with syntactic trees. Details of these two corpora are explained in the succeeding sections.

4.1 Real Word task

We first conducted the ordinary WSD experiments in order to investigate which features are effective for Vietnamese WSD classifiers. We called this task Real Word task (RW task hereafter). Since there is no sense-tagged corpus for Vietnamese WSD, in order to train SVM classifiers, a manually sense-tagged corpus named 'RW corpus' is built using Vietnamese Treebank (Nguyen et al. 2009)[4]. The tagging process was conducted as follows: we first choose 9 verbs, 11 nouns and adjectives for target words. These words are chosen considering the following conditions: it is a high frequency word in Vietnamese Treebank, it is ambiguous and both senses of it are expected to appear sufficiently in the Treebank. For each target word, about 100 sentences were chosen for sense tagging, resulted in around 3,000 sentences for all verbs, nouns and adjectives. Two Vietnamese native speakers were invited to judge independently which sense a target word had in those sentences. Chosen senses are those defined in VDict Vietnamese dictionary[5]. Average number of senses for target words in VDict is 3.1. However, not all but only two coarse grained

[4] Vietnamese Treebank contains about 10,000 sentences which come from news articles. Vietnamese Treebank has already been available at http://vlsp.vietlp.org:8080/demo/?page=resources. At the same site, some other Vietnamese language resources (such as machine readable dictionary and bilingual corpus) are available.

[5] VDict—An online Vietnamese-Vietnamese dictionary `http://vdict.com/`, online accessed 2009-11-01.

senses for each target word are annotated. The inter-annotator aggreement is 90.63%. For the disagreed sentences, two annotators discussed together and determined the final sense. We call the above sense tagged corpus 'RW corpus'. The average numbers of sentences for verbs, nouns and adjectives are 92.3, 116.7 and 92.1, respectively. Full lists of chosen target words and their senses are shown in Figure 1.

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
N1	nhà	house	87
		family	44
N2	nước	water	69
		country	81
N3	đường	street, route	100
		a way to do something	27
N4	biển	the sea	7
		sign, plate	95
N5	thứ	kind, sort, category	33
		place, position	72
N6	giờ	an hour	44
		now	64
N7	chiều	dimension	25
		afternoon	72
N8	tên	name	78
		a word used to indicate a person (impolite)	22
N9	hàng	product	95
		line	13
N10	đầu	a tip, an end	36
		the beginning	70
N11	tiếng	an hour	68
		sound	82
A1	lớn	big	137
		old	13
A2	nhỏ	small	71
		young	35
A3	khó	difficult	71
		poor	6
A4	dài	long (distance)	73
		long (time)	15
A5	nặng	heavy (weight)	21
		serious (illness)	34
A6	trên	above	13
		more than, over	57
A7	trước	before	93
		in front of	16
A8	phải	something right	87
		right hand side	11
A9	tốt	good in quality (product)	54
		nice, honest (person)	22

Fig. 1 List of ambiguous words and their senses

4.2 Pseudoword task

Although using ordinary WSD classifiers can give us more reliable results, the problem is a sense tagged corpus is not easily built. Therefore, we applied the pseudoword technique to automatically develop a sense-tagged corpus, and trained WSD classifiers from it. We call this task Pseudoword task (PW task). The main goal of this task is to evaluate the applicability of pseudoword technique for exploring effective features of WSD by comparing results between RW and PW tasks.

Let us suppose V_1 and V_2 are two different words. Pseudoword V_1-V_2 is an 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 by 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 a sentence. We call the obtained corpus 'PW corpus'. Although it is not a real WSD, a pseudo-sense tagged corpus can be easily created without any human intervention.

In many previous studies applying pseudoword technique to evaluate WSD methods, two words 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. (Lu et al. 2006).

Let us suppose w is a target word. We use VDict 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. Disambiguation of the pseudoword V_1-V_2 would simulate the disambiguation of the original target word w. For example, the Vietnamese verb '*mang*' has two meanings: "to bring something" and "to contain some characteristic of something". Then '*dem*' (bring) and '*chua*' (contain) are selected as pseudo-senses of '*mang*'. We chose 9 verbs, 9 nouns, and 5 adjectives as target words in PW task, which are the subset of target words in RW task. Some target words in RW task are discarded in PW task because of the lack of data in our corpus. Figure 2 reveals the target words and their two pseudo-senses of verbs, nouns and adjectives, respectively. Note that in order to increase the number of training and test instances, a pseudo-sense is sometimes represented as a set of synonymous words such as V2, N1 and A1. The figure also includes pseudo-senses of the 2 remaining target nouns and 4 adjectives in RW task, whose IDs are shown in italics. Since we could not get enough training data for these pseudo-senses, we removed them from target

ID	Target word	Pseudo-sense	Occurrences
V1	mang	đem (bring)	47
		chứa (contain)	18
V2	đưa	trao (give);trao tặng (give as present);chuyển giao (hand over)	26
		hướng dẫn (guide);điều khiển (control)	22
V3	lấy	sử dụng (use)	68
		cưới (marry);kết hôn (get married)	15
V4	chuyển	gửi (send)	129
		thay đổi (change);đánh đổi (swap);đổi (alter)	87
V5	tiếp	đón (go to meet)	48
		tiếp tục (continue)	79
V6	nhận	chấp nhận (accept);công nhận (certify);chứng nhận (certify);nhận lời (accept)	49
		xác nhận (confirm);phân biệt (distinguish)	29
V7	mất	mất mát (lost);mất mùa (have a poor crop);mất ngủ (lack of sleep);mất tích (missing)	19
		chết (die)	146
V8	xem	nhìn (look)	190
		nghĩ (think)	106
V9	bắt	ép (force)	12
		giữ (keep)	72
N1	nhà	nhà cửa (house);nhà đất (land);nhà máy (factory);nhà trọ (lodging-house);nhà xưởng (mill)	74
		gia đình (family)	288
N2	nước	con nước (tide);mặt nước (water face);nước mắm (fish source);nước mắt (tear);nước mặn (sea water);nước ngọt (water);nước ngầm (ground water);nước sạch (clean water);sông nước (river)	95
		xã hội (society);đất nước (country);nhà nước (state);nước ngoài (abroad);nước nhà (home country)	216
N3	đường	đường phố (street);đường bộ (road);đường mòn (trail)	51
		hướng (direction);cách (way to do something)	189
N4	biển	bảng (sign)	21
		sông (river)	147
N5	thứ	loại (type)	55
		hạng (rating)	17
N6	giờ	giờ phút (hours);phút giây (moment);phút (minute)	73
		hiện (now);hiện giờ (at the moment)	11
N7	chiều	hướng (direction);chiều hướng (tendency)	17
		chiều tối (evening);đêm tối (night);tối (evening);buổi sáng (morning)	59
N8	tên	tên tuổi (name)	17
		kẻ (a guy)	46
N9	hàng	gian hàng (booth);mặt hàng (item);hàng hiệu (luxury);hàng quán (shop);hàng hóa (goods)	40
		hàng ngũ (line);dòng (flow)	67
N10	đầu	đỉnh (top);mút (tip);chop (tip)	28
		khởi đầu (begin);mở đầu (opening)	8
N11	tiếng	ngôn ngữ (language)	4
		giọng (voice)	44
A1	lớn	lớn lao (great);rộng lớn (large);to lớn (big)	16
		khôn lớn (grow up);lớn khôn (grown);lớn tuổi (old);tuổi già (old)	59
A2	nhỏ	nhỏ bé (small);nhỏ nhắn (little);nhỏ nhặt (minor);nhỏ nhỏ (tiny);nhỏ nhoi (tiny)	20
		trẻ (young);trẻ trung (young);non trẻ (infant)	85
A3	khó	dễ (easy)	42
		nghèo (poor)	121
A4	dài	xa (far)	71
		lâu (long);lâu dài (long-term)	79
A5	nặng	nặng nề (heavy);nặng nhọc (heavy);trĩu nặng (heavy)	28
		nghiêm trọng (serious);quan trọng (important)	47
A6	trên	này (this);đây (this);kia (that)	0
		hơn (more than)	42
A7	trước	xa xưa (old);cổ xưa (ancient);xưa (ancient)	19
		tiền hậu (front and back)	0
A8	phải	đúng (true);có lý (reasonable)	121
		tả hữu (left right)	0
A9	tốt	tươi (fresh);lành (good)	25
		tốt bụng (kindhearted);từ tế (kind)	11

Fig. 2 List of pseudowords and their pseudo-senses

words of PW task[6]. The PW corpus comprises 1,162 sentences for verbs, 1,483 sentences for nouns and 568 sentences for adjectives. The average samples of pseudo-verbs, pseudo-nouns and

[6] It is possible to prepare a lot of Vietnamese texts to obtain more training examples for these words. However, manually annotated syntactic trees are used to derive SYN features. Thus we used Vietnamese Treebank, which is a relatively small corpus.

pseudo-adjectives are 129.1, 164.8 and 113.6, respectively. The number of adjective instances is less than verbs and nouns because the frequency of ambiguous adjectives in the corpus is low. Also, since the adjectives have fine-grained senses, it is more difficult to disambiguate them.

4.3 Pseudoword and Real Word task

We will present a method to train WSD classifiers without sense-tagged corpora in this subsection. 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 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 advantage of this approach is that no sense-tagged corpus is required for supervised learning of WSD systems.

5 Evaluation

For each experiment, we first evaluate the effectiveness of each feature separately, then the feature combinations. LIBSVM (Chang and Lin 2001) is used for training SVM classifiers. Experiments in RW task and PW task are conducted by 10-fold cross validation. For PW-RW task, PW corpus is used as training set and RW corpus is used as test set.

The Baseline used in the experiments is the most frequent sense method. That is, all test instances of a target word are determined to be the most frequent sense appearing in the training data.

The evaluation criteria for WSD systems is the accuracy of sense classification defined as in Eq. (12).

$$acc = \frac{\text{number of correct instances}}{\text{total number of instances}} \tag{12}$$

In each task, 15 feature sets are used for training WSD classifiers. The first four utilize one feature type, while the others utilize two, three, or four feature types (feature combination). In following subsections, accuracies of trained WSD classifiers for individual target words are reported. Average accuracies for verbs, nouns, adjectives and all target words are also shown. For the results of individual target words, not all but only the first and second ranked feature combinations are shown.

5.1 Results of Real Word task

Table 3 shows results for each target word, while Table 4 shows the average accuracies for verbs, nouns, adjectives and all target words in RW task. Results of SVM classifiers are verified by McNemar's test ($p < 0.05$). $*$ means the case that it significantly outperforms Baseline. The bold number indicates the best accuracy achieved when one feature type is used, or when two or more feature types are used. If † is attached, the system significantly outperforms the second

Table 3 Accuracy in RW task for each target word.

Word	Baseline	BOW	POS	COL	SYN	Comb. 1st	Comb. 2nd
V1	66.0	**98.0** *†	77.0	68.0	70.0	**97.0** * (BOW+COL)	94.0 * (BOW+SYN)
V2	55.0	**93.0** *†	66.0	80.0 *	61.0	**94.0** * (BOW+COL)	92.0 * (BOW+COL+SYN)
V3	53.5	**96.5** *	88.4 *	88.4 *	93.0 *	**100.0** * (All Features)	**100.0** * (BOW+POS+SYN)
V4	61.5	**92.3** *†	73.1	67.9	70.5	**91.0** * (BOW+COL)	87.2 * (BOW+POS+COL)
V5	68.3	**85.4** *	80.5	80.5	75.6	**90.2** * (All Features)	**90.2** * (BOW+POS+SYN)
V6	55.0	**99.0** *	72.0 *	97.0 *	75.0 *	**98.0** * (BOW+COL)	97.0 * (BOW+COL+SYN)
V7	80.8	**92.3** *	82.7	82.7	84.6	**94.2** * (All Features)	92.3 * (BOW+POS+COL)
V8	74.0	**93.5** *†	77.2	82.9 *	74.8	**92.7** * (BOW+SYN)	89.4 * (All Features)
V9	83.8	87.9	80.8	83.8	**92.9** *	**94.9** * (BOW+POS+SYN)	**94.9** * (BOW+SYN)
N1	66.4	**98.5** *†	73.3	80.9 *	80.2 *	**96.9** * (BOW+COL)	**96.9** * (BOW+POS)
N2	54.0	**93.3** *	71.3 *	89.3 *	82.0 *	**96.7** * (BOW+POS+COL)	96.0 * (All Features)
N3	78.7	**100.0** *†‡	83.5	87.4 *	84.3	**94.5** * (BOW+SYN)	93.7 * (BOW+POS+SYN)
N4	93.1	93.1	96.1	96.1	**98.0**	**98.0** (POS+SYN)	97.1 (BOW+POS+SYN)
N5	68.6	91.4 *	**97.1** *	91.4 *	95.2 *	**100.0** * (All Features)	**100.0** * (BOW+POS+COL)
N6	59.3	**94.4** *†	81.5 *	78.7 *	75.9 *	**97.2** * (BOW+COL+SYN)	95.4 * (All Features)
N7	74.2	85.6 *	**91.8** *	83.5 *	82.5	**95.9** * (BOW+POS)	91.8 * (BOW+COL+SYN)
N8	78.0	**96.0** *	83.0	90.0 *	90.0 *	**97.0** * (BOW+SYN)	96.0 * (All Features)
N9	88.0	**90.7**	89.8	88.0	88.9	**90.7** (BOW+SYN)	89.8 (BOW+POS+SYN)
N10	66.7	**95.2** *	72.4	91.4 *	92.4 *	**97.1** * (BOW+SYN)	96.2 * (BOW+POS+SYN)
N11	54.7	**97.3** *	93.3 *	86.0 *	92.0 *	**97.3** * (BOW+COL+SYN)	**97.3** * (BOW+SYN)
A1	91.3	**98.7** *†	88.0	91.3	92.7	**96.0** * (BOW+SYN)	95.3 (BOW+POS+SYN)
A2	67.0	**92.5** *†	64.2	75.5 *	74.5	**97.2** * (BOW+SYN)	96.2 * (BOW+POS+SYN)
A3	92.2	92.2	**94.8**	92.2	93.5	**96.1** (POS+SYN)	93.5 (POS+COL+SYN)
A4	83.0	**86.4**	76.1	**86.4**	81.8	**88.6** (BOW+SYN)	**88.6** (BOW+POS)
A5	61.8	**94.5** *†	65.5	69.1	65.5	**94.5** * (BOW+POS+COL)	**94.5** * (BOW+SYN)
A6	81.4	90.0 *	**97.1** *	81.4	77.1	**100.0** * (BOW+POS+SYN)	**100.0** * (BOW+POS)
A7	85.3	**97.2** *	79.8	93.6 *	85.3	**94.5** * (BOW+POS+COL)	**94.5** * (BOW+COL+SYN)
A8	88.8	88.8	**95.9**	89.8	89.8	**99.0** * (BOW+POS+SYN)	98.0 * (POS+COL+SYN)
A9	71.1	**97.4** *†	55.3 *	85.5 *	78.9	**98.7** * (BOW+SYN)	97.4 * (BOW+COL)

Table 4 Average accuracy in RW task for verbs, nouns, adjectives and all target words.

	Baseline	BOW	POS	COL	SYN
Verb	66.9	**93.6** *†	77.3 *	81.6 *	77.5 *
Noun	69.8	**94.5** *†	84.3 *	87.4 *	87.1 *
Adj	81.7	**93.5** *†	80.5	86.1 *	83.6
All	72.3	**94.0** *†	81.2 *	85.4 *	83.4 *

	BOW+ POS	BOW+ COL	BOW+ SYN	POS+ COL	POS+ SYN	COL+ SYN	POS+ COL+ SYN	BOW+ COL+ SYN	BOW+ POS+ SYN	BOW+ POS+ COL	All Feat.
Verb	90.0 *	92.4 *	**92.7** *	82.2 *	79.1 *	85.3 *	85.7 *	92.1 *	90.7 *	91.0 *	91.9 *
Noun	**95.1** *	94.0 *	95.0 *	89.2 *	88.4 *	89.4 *	90.7 *	94.9 *	94.9 *	94.6 *	94.7 *
Adj	93.6 *	91.3 *	93.0 *	88.9 *	84.3	87.2 *	89.3 *	91.3 *	**94.6** *	93.2 *	92.6 *
All	93.2 *	92.8 *	**93.8** *	87.1 *	84.6 *	87.6 *	88.9 *	93.1 *	93.6 *	93.2 *	93.3 *

best system among one feature or combined feature groups. To clearly show the effectiveness of feature combination, ‡ is attached if the difference between the best single and combined feature is statistically significant[7].

First, we see that almost all WSD classifiers of single features except POS and SYN for adjectives, are significantly better than the Baseline method. When only a single feature is used, BOW was better than the other three features in almost all words. This is reasonable because BOW can capture the most contextual information of a target word. As a human usually does when facing an ambiguous word, BOW utilizes the context around the target word to find the key words that help disambiguate it. The POS feature only contains the grammatical information of several words around the target word, but not the *'meanings'* of these words. So, their surrounding POS may not be clearly discriminative. The results of POS feature are usually the lowest in comparison with the others, even with baseline. SYN feature is also not so effective for adjectives (only 1.9% higher than Baseline), since we only use 4 syntactic relations for an adjective. This may cause data sparseness for training SVM classifiers. However, SYN feature works well on verbs and nouns (with 10.6% accuracies higher than Baseline for verb and 17.3% for noun). On average, when applying a single feature in Vietnamese WSD, BOW is the most effective feature, followed by COL, SYN and POS feature.

In Table 3, WSD classifiers with combined feature sets got equal or higher results compared to individual features for some target words. In Table 4, the best feature combination outperforms the best single feature BOW for nouns and adjectives on average. However, BOW+SYN, which is the best feature combination for all words, are not higher than BOW. Note that the differences

[7] Tables 5, 6, 8 and 9 are also denoted in the same format.

between the best single and combined feature sets are insignificant (not marked by ‡), indicating that combining several features is not obviously better or worse than the use of only one type of feature. Increasing the number of feature types in feature combination could not lead to the improvement of accuracies. The 4 feature types combination is better than the combination of 2 or 3 features only for one verb (V7). Furthermore, the best feature combinations are different for individual target words, and differences between the best and second best of feature combination are insignificant (not marked by †) because of the relatively small size of the training corpus. Therefore, we cannot conclude what is the best feature combination for Vietnamese WSD from our result.

5.2 Results of Pseudoword task

Table 5 shows results of each pseudoword in PW task, and Table 6 shows the average accuracies

Table 5 Accuracy in PW task of each pseudoword.

Word	Baseline	BOW	POS	COL	SYN	Comb. 1st	Comb. 2nd
V1	72.3	**87.7***	70.8	78.5	64.6	**93.8*** (BOW+POS+SYN)	89.2* (BOW+SYN)
V2	54.2	**83.3***	68.8	79.2*	56.3	**91.7*** (BOW+POS+COL)	**91.7*** (BOW+COL)
V3	81.9	**94.0***	83.1	91.6*	89.2	**94.0*** (BOW+POS+SYN)	**94.0*** (POS+COL+SYN)
V4	59.7	**94.0***†	73.6*	79.6*	72.7*	**96.3*** (BOW+COL)	94.0* (BOW+COL+SYN)
V5	62.2	**89.0***	81.1*	73.2*	75.6*	**94.5*** (BOW+POS+COL)	**94.5*** (BOW+SYN)
V6	62.8	**84.6***	46.2*	78.2*	65.4	**87.2*** (BOW+POS+COL)	85.9* (All Features)
V7	88.5	**92.1***	82.4*	89.7	87.3	**91.5** (BOW+SYN)	90.3 (All Features)
V8	64.2	**94.6***	76.4*	90.5*	71.6*	**98.0***‡ (BOW+COL+SYN)	97.6* (BOW+COL)
V9	85.7	88.1	88.1	85.7	**89.3**	**89.3** (BOW+POS+SYN)	88.1 (BOW+SYN)
N1	79.6	**95.0***†	76.0	84.3*	84.0*	**93.9*** (BOW+COL)	93.6* (BOW+POS+COL)
N2	69.5	**95.5***†	72.0	81.0*	75.9	**95.2*** (BOW+COL+SYN)	94.2* (BOW+COL)
N3	78.8	**95.0***†	87.9*	89.2*	87.9*	**94.6*** (All Features)	**94.6*** (BOW+POS+COL)
N4	87.5	**92.9***	92.3	89.3	91.7*	**94.6*** (BOW+POS+COL)	94.0* (All Features)
N5	76.4	**91.7***	88.9	90.3*	86.1	**94.4*** (BOW+POS+SYN)	93.1* (All Features)
N6	77.6	88.2*	86.8	77.6	**93.4***	**94.7*** (BOW+POS+SYN)	**94.7*** (POS+SYN)
N7	52.6	**90.5***	83.6*	89.7*	84.5*	**94.8*** (BOW+SYN)	94.0* (BOW+COL)
N8	73.0	84.1*	**87.3**	**87.3***	82.5	**93.7*** (BOW+POS+COL)	**93.7*** (BOW+POS+SYN)
N9	62.6	**91.6***†	72.0	79.4*	78.5*	**97.2***‡ (BOW+COL)	95.3* (BOW+COL+SYN)
A1	78.7	**80.0**	72.0	78.7	70.7*	**84.0** (BOW+SYN)	81.3 (All Features)
A2	81.0	**83.8**	75.2	82.9	**83.8**	**88.6*** (BOW+SYN)	84.8 (BOW+POS+SYN)
A3	74.2	**93.9***†	77.3	85.9*	76.7	**96.3*** (BOW+COL+SYN)	93.9* (BOW+COL)
A4	52.7	86.0*	72.7*	**87.3***	64.0*	**94.0***‡ (BOW+COL)	93.3* (BOW+POS+COL)
A5	62.7	**89.3***	62.7	80.0*	72.0	**90.7*** (BOW+COL)	89.3* (All Features)

Table 6 Average accuracy in PW task of pseudo-verbs, pseudo-nouns, pseudo-adjectives and pseudowords all.

	Baseline	BOW	POS	COL	SYN
Verb	69.4	**91.5** *†	75.9 *	84.3 *	75.6 *
Noun	74.5	**93.3** *†	80.8 *	85.1 *	84.0 *
Adj	68.8	**87.5** *†	73.1	84.0 *	73.2 *
All	71.6	**91.6** *†	77.7 *	84.6 *	79.1 *

	BOW+ POS	BOW+ COL	BOW+ SYN	POS+ COL	POS+ SYN	COL+ SYN	POS+ COL+ SYN	BOW+ COL+ SYN	BOW+ POS+ SYN	BOW+ POS+ COL	All Feat.
Verb	88.4 *	92.3 *	91.2 *	86.5 *	77.6 *	86.6 *	87.1 *	**92.6** *	89.9 *	91.4 *	91.0 *
Noun	91.8 *	**93.2** *	92.7 *	87.5 *	83.0 *	86.5 *	88.3 *	93.0 *	92.6 *	93.1 *	92.9 *
Adj	84.9 *	89.6 *	87.0 *	86.4 *	75.0 *	84.0 *	86.3 *	**89.8** *	86.3 *	89.1 *	89.1 *
All	89.4 *	92.2 *	91.2 *	87.0 *	79.7 *	86.1 *	87.5 *	**92.3** *	90.5 *	91.8 *	91.6 *

for pseudo-verbs, pseudo-nouns, pseudo-adjectives and all target words.

We can see that results when only a single feature is used are similar to RW task, in which BOW feature gave the best performance. As we discussed in Subsection 5.1, BOW contains the most lexical information around the target word. Results of POS feature are not always the lowest in comparison with the others, however in some cases, they are lower than the Baseline (3 of 9 verbs, 1 of 9 nouns, 2 of 5 adjectives). COL feature also gave relatively high results for all parts-of-speech. This is because usages of two target words in two classes are different, so their collocations are very different. However, COL still could not perform better than BOW.

When two or more features are combined together, WSD classifiers gave better results compared to single features for 8 of 9 verbs, 6 of 9 nouns, and all adjectives. Table 6 showed that the most effective feature combination is BOW+COL+SYN for verbs and adjectives, while BOW+COL is most effective for nouns. However, the differences among feature combinations including BOW are not so great. The combinations without BOW are worse, since they do not take advantage of referring to the wide range of lexical information around the target word as BOW does. Similar to RW task, the best feature combinations in PW task vary for individual target words as shown in Table 5. This might be because our training corpus is not large enough.

5.2.1 Comparison of Effective Features in RW and PW task

If the best feature set found in PW task is same as one in RW task, it indicates that, even when we do not have a word sense tagged corpus, we can apply pseudoword technique to find the effective features for Vietnamese WSD. As shown in Table 6, on average, BOW is the most effective feature, followed by COL, SYN and POS features in PW task. The order is the same

Table 7 The best feature comparison for each target word.

POS	Single	Combined
V	9/9	4/9
N	4/9	2/9
A	4/5	2/5

as for the RW task (in Table 4). Thus investigation of effective features by pseudoword sense disambiguation is reasonable.

Looking deeper to the similarity between results of PW task and RW task helps us to verify the applicability of pseudoword technique for investigating effective features in more details. Table 7 reveals two numbers in the form of a/b: a is the number of target words where the best (or one of the best) feature set is the same in PW and RW tasks, while b is total number of target words shared in PW and RW tasks. The 'Single' column indicates the case in which the best single feature sets are the same, while 'Combined' column indicates the case of combined feature sets. As shown in the table, pseudoword is only appropriate for choosing the best single feature when the target word is a verb or an adjective, since the best single feature of all target verbs and 4 of 5 target adjectives in PW task agreed with those in RW task. It seems ineffective for choosing the best single feature for nouns, as well as the best feature combination for all categories.

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, the first sense of the ambiguous verb *'V4.chuyen'* is *'to send'*. This sense can only be used in text related to email, postcard or documents. Similarly, the second sense of the adjective *'A5.nang'* is *'serious'*. This sense can only be used in a context related to health and disease. However, domains for using nouns are very large. For example, the second sense of the ambiguous noun *'N6.gio'* 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 might appear.

5.3 Results of pseudoword and Real Word task

In this task, we use two baselines. The first baseline, MFS-PW, is the system which always chooses the most frequent sense in PW corpus, the second one, MFS-RW, is the system choosing the most frequent sense in RW corpus. Comparison between these two baselines also enables us to verify how well pseudoword can simulate real word WSD. Table 8 shows results for each target

Table 8 Accuracy in PW-RW task for each target word.

Word	MFS-PW	MFS-RW	BOW	POS	COL	SYN	Comb. 1st		Comb. 2nd	
V1	66.0	66.0	66.0	**70.0**	67.0	65.0	**81.0** *†‡	(POS+SYN)	71.0	(POS+COL+SYN)
V2	45.0	55.0	61.0	58.0	55.0	**66.0**	**70.0**	(BOW+COL+SYN)	66.0	(COL+SYN)
V3	46.5	53.5	52.3	47.7	55.8	**59.3**	**58.1**	(POS+SYN)	57.0	(POS+COL+SYN)
V4	38.5	61.5	52.6	**53.8**	43.6	51.3	**59.0**	(BOW+COL+SYN)	57.7	(BOW+POS+SYN)
V5	68.3	68.3	48.8	41.5	**68.3**	63.4	**73.2**	(BOW+COL)	65.9	(BOW+SYN)
V6	55.0	55.0	**56.0**	55.0	55.0	51.0	**64.0**	(POS+SYN)	58.0	(BOW+POS+SYN)
V7	19.2	80.8	19.2	**20.2**	19.2	19.2	**23.1**	(BOW+POS)	21.2	(POS+SYN)
V8	74.0	74.0	52.8	65.9	**72.4**	67.5	**69.1**	(BOW+POS+COL)	**69.1**	(POS+COL)
V9	83.8	83.8	83.8	78.8	83.8	**90.9**	**84.8**	(COL+SYN)	**84.8**	(BOW+SYN)
N1	33.6	66.4	38.2	39.7	35.1	**45.8**	**50.4**	(BOW+SYN)	49.6	(POS+SYN)
N2	54.0	54.0	47.3	38.7	**48.7**	32.0	**48.7**	(BOW+COL)	41.3	(BOW+POS+COL)
N3	21.3	78.7	29.9	46.5	24.4	**49.6**	**56.7** †	(POS+SYN)	48.8	(POS+COL+SYN)
N4	93.1	93.1	**93.1**	85.3	**93.1**	**93.1**	**96.1**	(POS+COL)	94.1	(POS+COL+SYN)
N5	31.4	68.6	32.4	**78.1** †	48.6	41.0	**78.1** †	(POS+SYN)	68.6	(POS+COL)
N6	59.3	59.3	59.3	64.8	63.0	**76.9** *†	**71.3** *	(BOW+POS+SYN)	70.4	(POS+SYN)
N7	74.2	74.2	73.2	75.3	75.3	**82.5**	**88.7** *	(BOW+POS+SYN)	87.6 *	(All Features)
N8	22.0	78.0	22.0	37.0	29.0	**44.0**	**50.0**	(POS+SYN)	44.0	(POS+COL+SYN)
N9	12.0	88.0	74.1	74.1	40.7	**77.8**	**80.6**	(BOW+COL+SYN)	79.6	(All Features)
A1	8.7	91.3	8.7	**22.7** †	9.3	13.3	**31.3** ‡	(POS+SYN)	26.7	(BOW+POS+SYN)
A2	33.0	67.0	33.0	35.8	33.0	**37.7**	**40.6**	(POS+SYN)	35.8	(BOW+POS+SYN)
A3	7.8	92.2	41.6	**57.1**	27.3	16.9	**63.6**	(POS+SYN)	62.3	(BOW+POS+SYN)
A4	17.0	83.0	37.5	54.5	**73.9**	69.3	**68.2**	(POS+COL+SYN)	**68.2**	(COL+SYN)
A5	61.8	61.8	40.0	36.4	**54.5**	50.9	**50.9**	(COL+SYN)	45.5	(POS+COL)

Table 9 Average accuracies in PW-RW task for verbs, nouns, adjectives and all words.

	MFS-PW	MFS-RW	BOW	POS	COL	SYN
Verb	55.1	66.9	55.0	55.7	57.6	**59.2**
Noun	43.9	72.4	51.1	58.2	49.6	**58.4**
Adj	21.6	81.1	28.4	**38.7**	34.7	34.0
All	43.3	72.2	47.8	53.3	49.4	**53.7**

	BOW+ POS	BOW+ COL	BOW+ SYN	POS+ COL	POS+ SYN	COL+ SYN	POS+ COL+ SYN	BOW+ COL+ SYN	BOW+ POS+ SYN	BOW+ POS+ COL	All Feat.
Verb	55.1	56.9	58.4	57.3	**59.1**	57.9	57.2	59.0	57.2	57.2	58.0
Noun	60.4	51.3	55.8	59.3	**63.8** †‡	55.4	59.9	55.0	60.3	56.6	58.0
Adj	39.7	31.3	35.1	37.6	**44.5** ‡	36.1	39.1	34.2	42.0	36.6	37.6
All	54.3	49.2	52.5	54.2	**58.2** †‡	52.4	54.7	52.2	55.5	52.7	53.8

word. Table 9 shows average results for verbs, nouns, adjectives and all target words[8].

Comparing results in RW task (Tables 3 and 4) and PW-RW task (Tables 8 and 9), we can

[8] In Tables 8 and 9, $*$ indicates that the system significantly outperforms MFS-RW.

see that accuracies of WSD systems in RW-PW task are worse than those in RW task in all feature sets. It seems that WSD classifiers trained from PW corpus could not perform as well as ones trained from RW corpus, although two words of pseudo-senses were 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 features extracted from PW corpus, and features from RW corpus. The second reason is that the most frequent sense of pseudowords in some cases totally different from the real most frequent sense. This can be empirically observed by seeing that there are great gaps between MFS-PW and MFS-RW in Table 8. For example, MFS-PW of *'V7.mat'* is 19.2% while its MFS-RW is 80.8%. Therefore, the training data for the least frequent sense in PW corpus could not learn the behavior of that sense in the RW corpus (which is the most frequent sense indeed). The worst case is adjectives where disagreement of the most frequent sense is found in 4 of 5 adjectives. This is also the reason why the accuracies for adjectives are much lower than for verbs and nouns.

As shown in Table 8, classifiers trained from PW corpus do not significantly outperform MFS-RW except for V1, N6 and N7 (marked by $*$). This might be because the training data (Vietnamese Treebank) used in our experiment is not so large. One way to enlarge the size of training data is to use not manually annotated but automatically analyzed syntactic trees for SYN features. However, no public syntactic parser for Vietnamese is currently available.

On average, in Table 9, systems without BOW feature achieved relatively better results. Although BOW works well on RW and PW task, it performs poorest compared to other feature sets. One of the reasons might be the mismatch of words appearing in the context of target words in PW and RW corpus. Many words in the test RW corpus might be 'unknown' in the training PW corpus, causing the decline of accuracy. Comparing BOW and POS, BOW would suffer from the mismatch, since the variety of words (feature space of BOW in other words) is much broader than that of POS. This assumption would be supported by the fact that POS is better than BOW in Table 9.

6 Discussion

In this section, we will discuss three issues: comparison between SVM and Naive Bayes model in 6.1, differences of effective WSD features for different languages in 6.2, and the previous work on the pseudoword technique in 6.3

Table 10 Average accuracies of Naive Bayes for all words in three tasks.

	MFS-PW	MFS-RW	BOW	POS	COL	SYN
RW Task	–	72.3	77.7$^{\triangleright}$	**81.5**	77.2$^{\triangleright}$	**81.5**$^{\triangleright}$
PW Task	71.6	–	74.5$^{\triangleright}$	**78.6**	74.9$^{\triangleright}$	77.4$^{\triangleright}$
PW-RW Task	43.3	72.2	44.8$^{\triangleright}$	**54.3**	43.6$^{\triangleright}$	51.0$^{\triangleright}$

	BOW+ POS	BOW+ COL	BOW+ SYN	POS+ COL	POS+ SYN	COL+ SYN	POS+ COL+ SYN	BOW+ COL+ SYN	BOW+ POS+ SYN	BOW+ POS+ COL	All Feat.
RW	80.1$^{\triangleright}$	78.0$^{\triangleright}$	79.0$^{\triangleright}$	79.4$^{\triangleright}$	**84.3**	78.8$^{\triangleright}$	80.7$^{\triangleright}$	78.7$^{\triangleright}$	80.9$^{\triangleright}$	79.2$^{\triangleright}$	79.9$^{\triangleright}$
PW	76.5$^{\triangleright}$	74.0$^{\triangleright}$	75.1$^{\triangleright}$	76.2$^{\triangleright}$	**80.9**	75.1$^{\triangleright}$	76.5$^{\triangleright}$	74.6$^{\triangleright}$	76.9$^{\triangleright}$	75.2$^{\triangleright}$	75.5$^{\triangleright}$
PW-RW	46.5$^{\triangleright}$	44.9$^{\triangleright}$	46.1$^{\triangleright}$	44.7$^{\triangleright}$	**51.8**$^{\triangleright}$	44.5$^{\triangleright}$	45.5$^{\triangleright}$	45.0$^{\triangleright}$	47.6$^{\triangleright}$	45.4$^{\triangleright}$	45.8$^{\triangleright}$

6.1 Naive bayes classifier

In order to compare our results with other learning method, we also performed experiments using a Naive Bayes classifier (NB hereafter) (Duda, Hart, and Stork 2001). Naive Bayes classifier is trained with the same feature set described in Section 4. Sense of a target word w_i is determined by Eq. (13).

$$S(w_i) = \arg\max_{S_k} \left[\ln P(S_k) + \sum_{x_j \in F_i} \ln P(x_j|S_k) \right] \tag{13}$$

where S_k is one of the two senses ($k = 0, 1$), $P(S_k)$ is the probability of sense S_k in the training corpus, x_j is a value of feature vector F_i, $P(x_j|S_k)$ is the conditional probability of x_j.

Table 10 shows results of Naive Bayes classifiers, i.e. averages of accuracies for all words in three tasks. The accuracies of NB are almost always worse than SVM. $\triangleright$ indicates that NB is significantly worse than SVM by McNemar's test ($p < 0.05$). One exceptional case is POS in PW-RW task where NB is better than SVM. However, the difference is not statistically significant. From these results, SVM might be a more appropriate learning algorithm for Vietnamese WSD. In RW task, BOW is the best single feature for SVM, while POS and SYN for NB. However, the accuracy of SVM using BOW feature is much higher than NB using POS or SYN. We still conclude that BOW is the best feature for Vietnamese.

6.2 Comparison of effective features for different languages

For English, several papers have reported empirical evaluation of different types of features. Lee and Ng evaluated a variety of features and supervised learning algorithms (SVM, Naive Bayes, AdaBoost and Decision Tree) on the SENSEVAL-2 and SENSEVAL-1 data (Lee and Ng 2002). Among 4 learning algorithms, SVM achieved the best performance. They compared features

similar to BOW, POS, COL and SYN in this paper, and reported that COL was the best feature type, followed by BOW, POS and SYN. When we implemented the SVM classifiers with the exactly same BOW, POS and COL feature proposed by (Lee and Ng 2002) and evaluated the performance of them for Vietnamese WSD, we found that COL was also the best (the average accuracy was 85.3 for all words), followed by SYN (83.4), POS (79.5) and BOW (79.3)[9]. On the other hand, when we used our own features described in Subsection 3.2, BOW was significantly better than COL for Vietnamese WSD as shown in Table 4. Our features seem more appropriate for Vietnamese WSD than Lee's ones, since the accuracy of our method was much better[10]. We may say that local collocations near the target word would be useful for English WSD, while words in the context in a wide range would be effective for Vietnamese.

Martínez et al. explored the contribution of syntactic features by training Decision List and AdaBoost on the SENSEVAL-2 English data (Martínez et al. 2002). The paper revealed that COL was more effective than SYN, although syntactic features contributed to the gain of WSD precision when they combined with COL and BOW. Mohammad and Pedersen have also reported similar results (Mohammad and Pedersen 2004). They trained Decision Tree on the data of SENSEVAL-2, SENSEVAL-1 and others, and showed that (1) COL was better feature than SYN, (2) simple ensemble of two classifiers using COL and SYN achieved the increase of the accuracy. As shown in Table 4, SYN was also less effective than COL for Vietnamese WSD. Seeing results of two feature combinations with SYN (BOW+SYN, POS+SYN and COL+SYN), SYN contributed to the gain of accuracies when it combined with POS and COL, but not with BOW since the performance of BOW was much better than SYN.

Murata et al. worked on the comprehensive study of supervised machine learning of Japanese WSD (Murata, Utiyama, Uchimoto, Ma, and Isahara 2003). They evaluated several machine learning methods (SVM, Naive Bayes, Decision List and ensembles of them) with several feature sets (COL, POS, SYN, BOW as well as topics of documents) on the data of SENSEVAL-2 Japanese dictionary task. The results of Naive Bayes classifiers, which was the best system except for ensembles of multiple learning algorithms, showed that the most effective feature was COL, followed by BOW, SYN and POS. Our results showed that BOW would be the most effective for Vietnamese WSD, but it might be less useful than COL in Japanese, like English.

[9] We could not implement the exactly same feature for SYN, because Lee and Ng used dependency trees to extract syntactic features, while we used constituent trees. Actually, they converted constituent trees to dependency trees, but did not show the conversion algorithm. Furthermore, they applied feature selection for SYN features, but detailed algorithm was not explained, either.

[10] Considering feature combination, our method was also better than Lee's method. When all 4 features were used, the accuracy of our method was 93.3, while Lee's method achieved 88.7. Without SYN feature (BOW+POS+COL), the accuracy is 93.2 for our method, while 86.9 for Lee's method.

Note that the above discussions are just rough comparisons between languages, since the feature sets used in previous work and ours are not exactly same. Furthermore, the effectiveness of features might be dependent not only on languages but also other factors, such as target words, sense definitions (fine or coarse grained), genres of texts and machine learning algorithms[11]. To more precisely explore differences of effective features among different languages, more sophisticated designs of experiments would be required. That is, we should prepare parallel corpora with annotations of senses, use bilingual or multilingual lexicons to define the same set of target words and their senses, train WSD classifiers using the same machine learning algorithm, and use the exactly same feature set. Such an experiment is beyond the scope of this paper, since currently we do not have the necessary language resources.

6.3 Previous work on pseudoword

Gale et al. introduced the 'pseudoword' technique at first in English (Gale et al. 1992b). They built a pseudo-ambiguous word by combining two or three randomly chosen unambiguous words and tried to disambiguate these two or three pseudo-senses. The unambiguous words came from definition sentences in a dictionary, and they were chosen so that the frequencies of pseudowords were equal. Although this is not a real WSD system, the idea of pseudoword helps to develop large amounts of training material. In the study of (Gaustad 2001), the author constructed experiments to compare the performance of Naive Bayes classifier for real ambiguous word and pseudoword. Pseudowords were created by choosing words with the same frequency ratios to that of real senses. The paper reported that accuracies of pseudoword disambiguation were different from that of real WSD, indicating that pseudoword technique would not be valid for evaluation of WSD systems.

In most previous work, semantic properties of senses were not considered for the choice of pseudowords. While Lu et al. proposed the method for Chinese WSD to automatically choose unambiguous pseudowords similar to real senses using a thesaurus (Lu et al. 2006). Furthermore, like our PW-RW task, pseudowords in an unannotated corpus were used to estimate the probabilities of Naive Bayes model for real WSD. The trained NB achieved good results, even higher than supervised classifiers trained from a relatively small amount of sense tagged corpus.

Our pseudoword technique is similar to (Lu et al. 2006), which considers semantic properties of pseudowords. One of the differences is that pseudowords were automatically chosen using the Chinese thesaurus in (Lu et al. 2006), while manually chosen in this paper. Lu's method seems

[11] For example, the order of the effectiveness of features were different according to the machine learning algorithms in (Murata et al. 2003).

preferable to ours, since manual choice of pseudowords might be arbitrary. Another difference is the size of the training corpus. As discussed in 5.3, pseudoword technique did not work well in our experiment of PW-RW task, while it worked well with a large amount of training data in (Lu et al. 2006). From another point of view, the lack of language resources and tools in Vietnamese, such as a thesaurus (for automatic selection of pseudowords) and a syntactic parser (to obtain a large training corpus with parse tree), might be an obstacle to application of pseudoword technique for Vietnamese WSD.

7 Conclusion

In this research, we have developed a WSD system for Vietnamese language on two corpora: RW corpus (which was manually built) and PW corpus (collected automatically). In RW task, the best average accuracy for all words is 94.0%. We have experimented using three tasks to evaluate the effectiveness of each feature and feature combinations with and without a sense-tagged corpus. For the first goal to explore effective features, we found that BOW is the most effective one. Combinations of BOW and other features enhance the performance of WSD system in some cases, but not significantly. For the other goal to check the applicability of the pseudoword technique, we found that it is useful to rank feature types according to effectiveness for WSD and find best single feature for individual target verbs and adjectives. In addition, pseudoword technique might be an alternative WSD approach when there is no training data.

However, 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 will be interesting to investigate the effective features for WSD multi-class classifiers along with increasing the corpus size. Also, we could not clearly find the best feature combination. More large-scaled sense tagged corpus enables us to explore the best feature combination for Vietnamese WSD. Effectiveness of other types of features should also be investigated. For example, Cai et al. used features about the topics of documents (Cai, Lee, and Teh 2007), which are derived by Latent Dirichlet Allocation (Blei, Ng, and Jordan 2003). They reported that topic features were effective for English, but not sure for Vietnamese. Although the results of our experiments in PW-RW task showed that pseudoword technique did not work well as unsupervised WSD method, it should be evaluated again with a larger corpus. Another interesting proposal is comparing the effective features between Vietnamese WSD and other languages in precise experiments as discussed in Subsection 6.2.

Reference

Agirre, E. and Edmonds, P. (2006). *Word Sense Disambiguation, Algorithms and Application*, Vol. 33. Text, Speech and Language Technology.

Agirre, E. and Martinez, D. (2001). "Learning Class-to-class Selectional Preferences." In *ConLL '01: Proceedings of the 2001 Workshop on Computational Natural Language Learning*, pp. 1–8. Association for Computational Linguistics.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, **3**, pp. 993–1022.

Cai, J. F., Lee, W. S., and Teh, Y. W. (2007). "Improving Word Sense Disambiguation Using Topic Features." In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1015–1023.

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

Corinna, C. and Vladimir, V. (1995). "Support-Vector Networks." *Machine Learning*, **20**, pp. 273–297.

Dang, H., Chia, C.-Y., Palmer, M., and Chiou, F.-D. (2002). "Simple Features for Chinese Word Sense Disambiguation." *Proceedings of the 19th International Conference On Computational Linguistics*, **1**, pp. 1–7.

Dinh, D. (2002). "Building a Training Corpus for Word Sense Disambiguation in English-to-Vietnamese Machine Translation." In *COLING-02 on Machine Translation in Asia*, pp. 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification, 2nd Edition.* Wiley-Interscience.

Gale, W. A., Church, K. W., and Yarowsky, D. (1992a). "One Sense per Discourse." In *HLT '91: Proceedings of the Workshop on Speech and Natural Language*, pp. 233–237, Morristown, NJ, USA. Association for Computational Linguistics.

Gale, W. A., Church, K. W., and Yarowsky, D. (1992b). "Work on Statistical Methods for Word Sense Disambiguation." In *AAAI Fall Symposium Series, Probabilistic Approaches to Natural Language*, pp. 54–60. AAAI Press.

Gaustad, T. (2001). "Statistical Corpus-Based Word Sense Disambiguation: Pseudowords vs Real Ambiguous Words." In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL/EACL 2001). Proceedings of the Student Research Workshop.*

Kaplan, A. (1955). "An Experiment Study of Ambiguity and Context." *Mechanical Translation*, **2**, pp. 39–46.

Lee, Y. K. and Ng, H. T. (2002). "An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 41–48.

Lesk and Michael (1986). "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*, pp. 24–26, New York, NY, USA. ACM.

Lu, Z., Wang, H., Yao, J., Liu, T., and Li, S. (2006). "An Equivalent Pseudoword Solution to Chinese Word Sense Disambiguation." In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 457–464, Sydney, Australia. Association for Computational Linguistics.

Martínez, D., Agirre, E., and Màrquez, L. (2002). "Syntactic Features for High Precision Word Sense Disambiguation." In *Proceedings of the 19th International Conference on Computational linguistics (COLING)*, pp. 626–632.

Mohammad, S. and Pedersen, T. (2004). "Combining Lexical and Syntactic Features for Supervised Word Sense Disambiguation." In *Proceedings of the Conference on Computational Natural Language Learning*, pp. 25–32.

Murata, M., Utiyama, M., Uchimoto, K., Ma, Q., and Isahara, H. (2003). "CRL at Japanese Dictionary-based Task of SENSEVAL-2—Comparison of Various Types of Machine Learning Methods and Features in Japanese Word Sense Disambiguation—." *Journal of Natural Language Processing*, **10** (3), pp. 115–133. (in Japanese).

Nguyen, P. T., Vu, X. L., Nguyen, T. M. H., Nguyen, V. H., and Le, H. P. (2009). "Building a Large Syntactically-Annotated Corpus of Vietnamese." In *Proceedings of the Third Linguistic Annotation Workshop*, pp. 182–185, Suntec, Singapore. Association for Computational Linguistics.

Yarowsky, D. (1994). "Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French." In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 88–95. ACL.

Yarowsky, D. and Florian, R. (2002). "Evaluating Sense Disambiguation across Diverse Parameter spaces." *Journal of Natural Language Engineering*, **8**, pp. 293–310.

Minh Hai Nguyen: received the B.S from University of Science, Vietnam in 2007 and M.S from Japan Advanced Institute of Science and Technology in 2010. She was a teaching assistant at University of Science, Vietnam from 2007 to 2009. She is currently a Ph.D. student at School of Information Science, Japan Advanced Institute of Science and Technology. Her current research field is natural language processing, specifically sentence retrieval methods and applications.

Kiyoaki Shirai: received the B.E., M.E. and Dr. Eng. from Tokyo Institute of Technology in 1993, 1995 and 1998, respectively. He was an assistant at the Graduate School of Information Science and Engineering, Tokyo Institute of Technology from 1998 to 2001. He is currently an associate professor at School of Information Science, Japan Advanced Institute of Science and Technology. His current research interests include natural language processing, especially corpus-based methods and their applications. He is a member of the Association for Natural Language Processing, the Information Processing Society of Japan and the Japanese Society of Artificial Intelligence.

(Received June 21, 2011)
(Revised August 29, 2011)
(Rerevised November 17, 2011)
(Accepted December 20, 2011)