

# Low Ambiguity First Algorithm: A New Approach to Knowledge-Based Word Sense Disambiguation

Dongjin Choi<sup>1</sup>, Myungwon Hwang<sup>2</sup>, Byeongkyu Ko<sup>1</sup>, Sicheon You<sup>3</sup>,  
and Pankoo Kim<sup>1</sup> (✉)

<sup>1</sup> Department of Computer Engineering, Chosun University, 375 Seoseok-dong,  
Dong-gu, Gwangju, Republic of Korea

{dongjin.choi84,byeongkyu.ko}@gmail.com, pkkim@chosun.ac.kr

<sup>2</sup> Korea Institute of Science and Technology Institute (KISTI), 245 Daehak-ro,  
Yuseong-gu, Daejeon, Republic of Korea  
mgh@kisti.re.kr

<sup>3</sup> School of Informatics and Product Design, Chosun University, 375 Seoseok-dong,  
Dong-gu, Gwangju, Republic of Korea  
syou@chosun.ac.kr

**Abstract.** The Word Sense Disambiguation (WSD) problem has been considered as one of the most important challenging task in Natural Language Processing (NLP) research area. Even though, many of scientists applied the robust machine learning, statistical techniques, and structural pattern matching approach, the performance of WSD is still not able to bit human results due to the complexity of human language. In order to overcome this limitation, currently, the knowledge base such as WordNet has gained high popularity among researchers due to the fact that this knowledge base can extensively provide not only the definitions of nouns and verbs, but also the semantic networks between senses which were defined by linguists. However, knowledge bases are not fully dealing with entire words of human languages because maintaining and expanding the knowledge base is huge task which requires many efforts and time. Expanding knowledge base is not a big issue to concern however, a new approach is the major goal of this paper to solve WSD problem only based on limited knowledge resources. In this paper, we propose a method, named *low ambiguity first (LAF)* algorithm, which disambiguates a polysemous word with a low ambiguity degree first with given disambiguated words, based on the structural semantic interconnections (SSI) approach. The LAF algorithm is based on the two hypotheses that first, adjacent words are semantically relevant than other words far way. Second, word ambiguity can be measured by frequency differences between synsets of the given word in WordNet. We have proved these hypotheses in the experiment results, the LAF algorithm can improve the performance of traditional WSD results.

**Keywords:** Word sense disambiguation · Natural language processing · WordNet · Low ambiguity first

# 1 Introduction

Word Sense Disambiguation (WSD) is one of the most important, complicated, and challenging task in the computational linguistics research area [1]. WSD is a task to find correct senses for the given words that have multiple senses but their appearances are the same (similarly, a polysemous type of word). For example, a noun *bat* represents not only for a *chiropteran*<sup>1</sup> but also *rackets* for squash, and *baseball bat*. When this polysemous word is appeared in a sentence, people are able to understand the meaning of the given polysemous word by referring co-occurred words but computers are not [8]. In order to compute human languages, many researchers have been studying for long time to discover the best approaches to obtain a good result, but it is still an ongoing problem.

Early approaches were started to make a corpus which was manually tagged senses of polysemous words from the small number of sentences [4]. After expanding and developing this small corpus, dictionaries had been lunched to public. It provided vast amount of definitions for the target language so, people started to apply the dictionaries to WSD task. The most famous dictionary-based approach is the *Lest* algorithm which was introduced by *Michael Lesk* in 1986 [2]. However, it has a limitation that WSD results are depended on the dictionaries.

A supervised and unsupervised method had been applied to overcome this limitation in WSD task. The supervised method can be considered as a classification task by using collocation, bag-of-words, n-gram<sup>2</sup>, and context words a feature [5]. This method can apply many kinds of pattern recognition and machine learning approaches such as, Decision list, Naive Bayes classifier, k-Nearest Neighbors (kNN) algorithm, and Support Vector Machines (SVMs). However, it requires corpus which includes tagging information for words, but a problem is that tagging task has to be done manually, so it requires many times and costs. The unsupervised method is based on the assumption that the same sense of a words will occur in similar contexts. Therefore, it can be called as Word Sense Discrimination, in other words, this method is not able to distinguish specific senses for given words from a target sentence [6].

Currently, the most popular and powerful approach to WSD task is based on Knowledge dictionaries [7] such as WordNet<sup>3</sup>. Especially, Structural Semantic Interconnections (SSI) algorithm [3] is the most well-known approach to WSD task. This algorithm creates structural specifications of the possible senses for each target word to disambiguate in a context. And it selects the best hypothesis sense according to a grammar which describing relations between sense specifications. Even though, this SSI algorithm is powerful algorithm based on the strong knowledge-base, there is a limitation to overcome. This paper will be focusing on this limitation provide proposals to overcome this weakness.

<sup>1</sup> Nocturnal mouselike mammal with forelimbs modified to form membranous wings and anatomical adaptations for echolocation by which they navigate.

<sup>2</sup> is a contiguous sequence of n items from a given sequence of text or speech.

<sup>3</sup> is a lexical database for the human languages provides definitions and relations among synonyms developed by Cognitive Science Laboratory of Princeton University.

In this paper, we propose a new WSD algorithm, named *Low Ambiguity First (LAF)*, which is based on the hypotheses that it is able to calculate ambiguity of words by using WordNet, and the word with low ambiguity degree must be disambiguated first. Moreover, adjacent words are semantically relevant than other words far way. These hypotheses are the proposals to overcome weaknesses of the SSI algorithm. We believe that the LAF algorithm can improve precision performance of WSD.

The reminder of this paper is organized as follows: In Sect. 2, we describe the SSI algorithm in details and point out its weaknesses. Section 3 is the main part of this paper that we present the low ambiguity first algorithm and explain how it works with examples. Also, the word ambiguity measurement will be illustrated. Finally, Sect. 4 concludes this paper with future works.

## 2 Related Works

Structural Semantic Interconnections (SSI) algorithm is a method to disambiguate polysemous words by creating structural specifications of the candidate senses for each word and select the most appropriate sense by using the structural grammar. The structural grammar is a possible relevant relations between structural specifications precisely, semantic interconnections among graphs. This SSI algorithm can be described as following variables:

- $T = [t_1, \dots, t_n]$  where,  $t$  is list of co-occurring terms to be disambiguated and  $n$  is a total number of noun types of word in the given sentence.
- $S_1^t, S_2^t, \dots, S_k^t$  are structural specifications of the possible concepts for the given  $t$ , where  $k$  is a total number of the possible concepts.
- $I = [S^{t_1}, \dots, S^{t_n}]$  is a list of the disambiguated senses (precisely, semantic interpretation of  $T$ ), where  $S^{t_i}$  is the chosen sense for the given  $t$  or the null element that the  $t$  is not yet disambiguated.
- $P = [t_i | S^{t_i} = null]$ , where  $P$  is a list of pending terms to be disambiguated.
- $G = (E, N, S_G, P_G)$ , where  $G$  is a context-free grammar,  $E$  is edge labels to indicate semantic relations between possible senses.  $N$  is a path between concepts and  $S_G$  is a start symbol of  $G$ .  $P_G$  is set of productions includes about 40 productions.

The SSI algorithm only considers noun types of word as a term to be disambiguated. Therefore, the list of  $t$  will be initialized with noun types of words from the given sentence. The WordNet definitions for the given  $t$  will be considered as a possible concepts ( $S_j^t$ ) for the  $t$ . If the target term  $t$  is a monosemous<sup>4</sup> word,  $I$  will be updated with  $S^{t_1}$ . If there are no monosemous terms nor initial synsets, the algorithm will choose the most probable sense based on the frequency of word senses.  $I$  will be updated as long as the SSI algorithm can find semantic relations between senses of  $I$  and possible senses of  $t$  in  $P$  by using  $G$ .

---

<sup>4</sup> having only single meaning or sense.

Let us assume that there is a sentence to be disambiguated as follows. *Sentence = Retrospective is an exhibition of a representative selection of an artists life work and art exhibition is an exhibition of art objects (paintings or statues).* The initial values of each variable will be updated as following:

$T = [\textit{retrospective}, \textit{work}, \textit{object}, \textit{exhibition}, \textit{life}, \textit{statue}, \textit{artist}, \textit{selection}, \textit{representative}, \textit{painting}, \textit{art}]$

$I = [\textit{retrospective}\#1, -, -, -, -, -, -, -, -]$

$P = [\textit{work}, \textit{object}, \textit{exhibition}, \textit{life}, \textit{statue}, \textit{artist}, \textit{selection}, \textit{representative}, \textit{painting}, \textit{art}]$

At first,  $I$  will be updated with the senses of monosemous words in the list of  $P$  as follows:

$I = [\textit{retrospective}\#1, \textit{statue}\#1, \textit{artist}\#1]$

$P = [\textit{work}, \textit{object}, \textit{exhibition}, \textit{life}, \textit{selection}, \textit{representative}, \textit{painting}, \textit{art}]$

The  $I$  will be enriched until the senses of  $I$  and possible senses of  $t$  in  $P$  have semantic interconnections (such as, *kind-of*, *has-kind*, *part-of* and *has-part* relations). Therefore, the final statuses of the lists are as follows:

$I = [\textit{retrospective}\#1, \textit{statue}\#1, \textit{artist}\#1, \textit{exhibition}\#2, \textit{object}\#1, \textit{art}\#1, \textit{painting}\#1, \textit{life}\#12]$

$P = [\textit{work}, \textit{selection}, \textit{representative}]$

As we can see in this SSI algorithm, there are two limitations to overcome. First, there are no criteria to measure which word needs to disambiguate earlier than others.  $P$  is a pending list to prepare terms to be disambiguated. However, the problem is that all the ambiguity of words is different from each other. Some words are mostly used by the first sense among other senses. Some words are uncertain due to their high ambiguities. Therefore, we need a method to measure the *Word Ambiguity (WA)* to decide which word has lower ambiguity than other words. So, we can reduce a possibility to make wrong choices in each disambiguation step.

Another limitation is based on the hypothesis that *the adjacent words are semantically relevant to each other*. In other word, a nearest word might have higher possibility to make strong semantic relations than a distant word. If this hypothesis is correct, we need to alter this SSI algorithm to consider structural locations of words in the given sentence.

In this paper, we demonstrate our hypothesis and prove that the SSI algorithm can be improved by concerning semantic relations among adjacent words. Also, we present the word ambiguity measurement by using WordNet sense frequency and propose Low Ambiguity First (LAF) algorithm in Sect. 3.

### 3 Low Ambiguity First Algorithm

This section presents the Log Ambiguity First algorithm (LAF), a knowledge-based Word Sense Disambiguation by applying the hypothesis that word ambiguity can be measured by frequency of senses in WordNet and the adjacent word are semantically relevant to each other. The variables of LAF algorithm is the same as the SSI algorithm as described in Sect. 2. However, items of pending list  $P$  will be ordered by their word ambiguity values.

#### 3.1 A Measurement for Word Ambiguity

A word Ambiguity (WA) is a criteria to measure complexity of word senses in the pending list  $P$ . It is important to decide which words disambiguate earlier than other words due to the fact that the SSI algorithm disambiguates senses of  $t$  in  $P$  by using the senses in  $I$  which already have been founded in the previous step. In other word, in each step, the best senses are chosen according to the current  $I$  and  $P$ , therefore, the order in which senses are chosen may affect the final result.

Let us assume that we have pending list  $P$  as follow:  $P = [group, Friday, investigation, Atlanta, primary\_election, evidence, irregularity]$ . According to the SSI algorithm, the  $I$  will be enriched by the monosemous words from the  $P$ . Therefore, we've got  $I = [-, Friday\#1, -, -, primary\_election\#1, -, -]$  and  $P = [group, investigation, Atlanta, evidence, irregularity]$  where,

$$\begin{aligned}
 S^{group}_3 &= [(2350)group\#1, (9)group\#2, (3)group\#3], \\
 S^{investigation}_2 &= [(16)investigation\#1, (8)investigation\#2], \\
 S^{Atlanta}_2 &= [(7)Atlanta\#1, (1)Atlanta\#2], \\
 S^{evidence}_3 &= [(54)evidence\#1, (24)evidence\#2, (7)evidence\#3], \\
 S^{irregularity}_4 &= [(3)irregularity\#1, (2)irregularity\#2, (1)irregularity\#3, \\
 & (irregularity\#4)]
 \end{aligned}$$

The word ambiguity can be measured by following Eq. 1 based on the frequencies of senses from each terms defined in WordNet.

$$WordAmbiguity(t) = \prod_{i=1}^{k-1} \prod_{j=i+1}^k \log \frac{frequency(S_i^t)}{frequency(S_j^t) + 1} \tag{1}$$

where,  $frequency(S_i^t)$  is the frequency of the  $i$ th sense of the given term  $t$  as illustrated in the previous paragraph. For example, the word ambiguity of term ( $t = group$ ) can be obtained as follow:

$$WordAmbiguity(group) = \log \frac{2350}{10} \times \log \frac{2350}{4} \times \log \frac{9}{4} \approx 2.312$$

If the word ambiguity value close to the zero, it means that the given term ( $t$ ) has high ambiguity (semantic complexity). In contrast, if the value far apart from the zero, it means that the given term has low ambiguity. In other word,

the term with low ambiguity will be disambiguated earlier than other words. Therefore, the pending list  $P$  will be updated by the order of word ambiguity as shown in the Fig. 1.

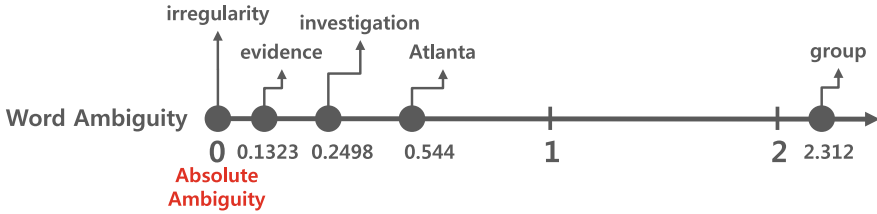


Fig. 1. Examples of Word Ambiguity corresponding to terms ( $t$ ) in the pending list  $P$ .

Therefore, the pending list  $P$  will be configured as follows:  $P = [group, Atlanta, investigation, evidence, irregularity]$ . This is the major difference between the SSI and LAF algorithm that the LAF algorithm will disambiguate low ambiguity word first. Detailed algorithm and explanations will be described in the next section.

### 3.2 An Adjacent Word May Have Stronger Relations than Other Words

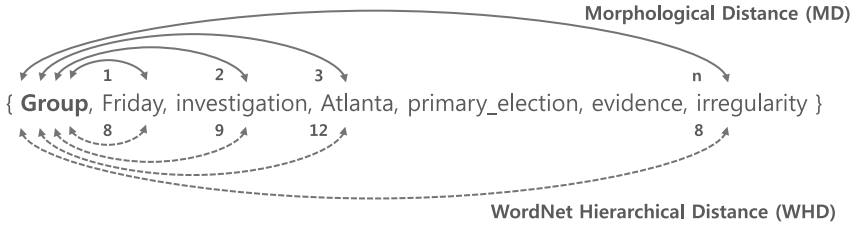
This section describes and demonstrates the hypothesis that an adjacent word of the target term ( $t$ ) may have stronger semantic relations than other words which far apart from the target term. In order to discover an evidence to prove this hypothesis can be applied in the SSI algorithm, we defined a Morphological Distance (MD) and WordNet Hierarchical Distance (WHD) as shown in Fig. 2. The MD is a morphological distance between target term ( $t$ ) and its adjacent words. And WHD is the shortest path distance between the target term and its adjacent words based on WordNet hierarchy.

Let us assume that we have a sentence as follows: *Sentence = [The Fulton-County-Grand-Jury said Friday an investigation of Atlanta’s recent primary-election produced no evidence that any irregularities took\_place]* which is the first sentence of the Brown1<sup>5</sup> of SemCor<sup>6</sup> data corpus. After preprocessing steps, we can obtain terms to be disambiguated as shown in Fig. 2. Therefore, the MD and WHD between the target term ( $t = group$ ) will be calculated as follows:

- $MD(Group, Friday) = 1, MD(Group, investigation) = 2, \dots,$   
 $MD(Group, irregularity) = 6$
- $WHD(Group, Friday) = 8, WHD(Group, investigation) = 9, \dots,$   
 $WHD(Group, irregularity) = 8$

<sup>5</sup> The Brown University Standard Corpus of Present-Day American English.

<sup>6</sup> A SemCor corpus is a manually sense-tagged corpora created by the WordNet project research team in Princeton University.



**Fig. 2.** Examples of the morphological distance and WordNet hierarchical distance between terms in pending list.

According to the examples in Fig. 2, the WHD was increased when the MD was getting bigger. However, the WHD was decreased even though the MD is getting bigger after certain point. Therefore, we need to discover this point that the tendency of WHD will be reversed, by comparing the MD and WHD from all sentences in Brown1 corpus. As a result, we obtained the average of WHD corresponding to the MD as shown in the following Table 1.

**Table 1.** Averages and Standard Deviations of WHD corresponding to MDs between terms in Brown1 corpus.

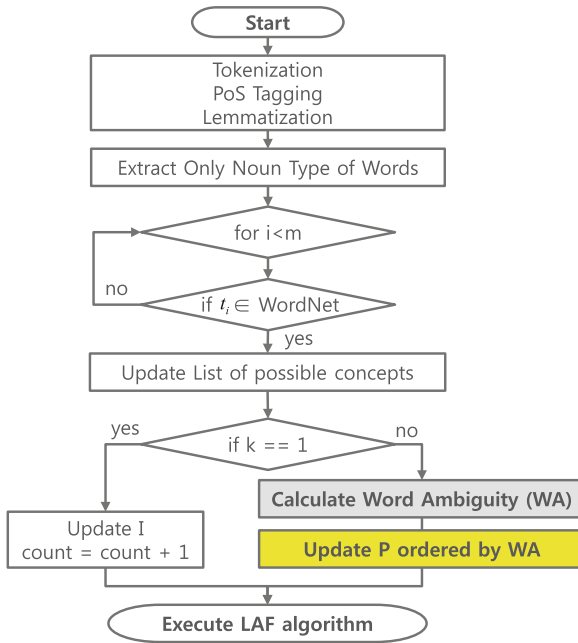
MD	Number of words	Ave. of WHD	Standard Deviation of WHD
MD = 1	5,161	10.16857	3.84991
MD = 2	5,133	10.32359	3.79279
MD = 3	4,330	10.41293	3.72705
MD = 4	3,420	10.19591	3.77212
MD = 5	2,661	10.24728	3.75285
MD = 6	1,983	10.29097	3.70380
MD = 7	1,436	10.03412	3.74383
MD = 8	1,068	10.08427	3.68752
MD = 9	752	9.98271	3.68088
MD = 10	531	10.45574	3.50725

As we can see in this table, the average value of WHD was increased until MD is equal to three. However, the average value of WHD was decreased when the MD is 4. These results can indicate that if the morphological distance between target term (*t*) and its adjacent words is less than 4, there will be a tendency that morphologically close words have a high possibility to make strong semantic relations. The most of the words (more than 80 percent) in Brown1 corpus are located in condition that the MD is one to six. Because it is hard to find a sentence with more than six kinds of nouns in the natural human language.

We hereby have proved that the adjacent words of the target term are likely to have shorter WHD than a word with higher MD under the certain point.

### 3.3 Low Ambiguity First Algorithm

In the previous sections, we have demonstrated two hypotheses that will be applied in the LAF algorithm, in order to improve the traditional SSI algorithm. Before we start to apply the LAF algorithm, we need to initialize the variables which were required for saving co-occurring terms, structural specifications of the possible concepts, list of the disambiguated senses, and pending list as described in Sect. 2. However, in the LAF algorithm, the pending list  $P$  will be updated after measuring the word ambiguity followed by the Eq. 1, and weighting function will be applied to calculate semantic similarity between unknown senses and known senses, changed by the morphological distance.



**Fig. 3.** Initializing processes for executing the LAF algorithm.

The proposed system will be developed by Python language and Pseudo-code for an implementation of the LAF algorithm described in the followings. This algorithm will take four kinds of list as input data which were the initialized list after preprocessing tasks shown in the Fig. 3. This algorithm will update list  $I$  and  $P$  until the system cannot find no further matching senses between items of  $P_C$  and  $I$ , by using the shortest path distance based on WordNet. The weighting function will be determined by the values of  $alpha$ ,  $beta$ , and  $theta$ , respectively. These parameters are the constant values for applying the hypothesis that described in the Sect. 3.2.



```

Low Ambiguity First Algorithm(T, I, P_C, P){
    import nltk
    T[m] = list of co-occurring terms(t)
    I[m] = list of disambiguated senses for the given t
    P_C[m][k] = list of possible concepts for the given t
    P[m'] = list of pending terms to be disambiguated
    count = counting variable for items of list I

    for(i=1;i<m';i++){
        for(j=1;j<count;j++){
            for(x=1;x<k;x++){
                // similarity will be based on the shortest path
                // distance in WordNet
                find similarity(P_C[i][x], I[j])
                return the most semantically close sense P_C[i][x]
                    and its value.
            if count == 1:
                update I, P
                count = count + 1
                break
            else:
                continue
        // alpha, beta, and theta are the constant values to
        // apply adjacent words have strong semantic relations
        // where, alpha + beta + theta = 1
        alpha * similarity(P_C[i][x], I[j])
        beta * similarity(P_C[i][x], I[j'])
        theta * similarity(P_C[i][x], I[j''])
        find the most relevant P_C[i][x]
        update I, P
        count = count + 1
        return P_C[i][x]
    }
}

```

In this proposed LAF algorithm, we are able to apply two hypotheses described in the previous Sects. 3.1 and 3.2, in order to overcome the limitations of traditional SSI algorithm. According to a base experiment<sup>7</sup> to verify reliability of the proposed algorithm, we have founded that our algorithm can improve the SSI algorithm. However, we are not going to describe experimental results in this paper due to the fact that the amount of testing data was not big enough. The major goal of this paper is the proposal of the LAF algorithm and demonstrations of the hypotheses. The experimental results will be introduced in the future works.

---

<sup>7</sup> We simply run a comparison test by using small amount of sentences.

## 4 Conclusions and Future Works

In this paper, we propose a new approach to overcome weaknesses of the traditional SSI algorithm which is the most popular knowledge-based WSD algorithm. People started to apply statistical approaches for disambiguating polysemous words however, the performance of these methods are still required improvements. Even though, knowledge base which is a machine readable knowledge database had been applied to WSD, there is a limitation that the current knowledge-base cannot cover entire senses of human language so far. The more we hold a rich knowledge base, the more we gained high performance for the WSD. However, enriching and maintaining knowledge base require many costs and time. Therefore we proposed a method only by using the limited current resources based on the two hypotheses. We demonstrated these hypotheses in this paper and proposed the Low Ambiguity First algorithm which is able to overcome weaknesses of the SSI algorithm. Experiments are still ongoing however, we believe that the proposed method can improve the performance of WSD.

**Acknowledgments.** This research was supported by SW Master's course of hiring contract Program grant funded by the Ministry of Science, ICT and Future Planning (H0116-15-1013) and the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (NRF-2014H1C1A1066494).

## References

1. Ide, N., Veronis, J.: Introduction to the special issue on word sense disambiguation. *Comput. Linguist.* **24**(1), 2–40 (1998)
2. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: *SIGDOC 1986: Proceedings of the 5th Annual International Conference on Systems Documentation*, pp. 24–26. ACM, New York (1986)
3. Navigli, R., Velardi, P.: Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(7), 1075–1086 (2005)
4. Weiss, S.F.: Learning to disambiguate. *Inform. Storage Retrieval* **9**(1), 33–41 (1973)
5. Jurafsky, D., Martin, J.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River (2000)
6. Pedersen, T.: Unsupervised corpus-based methods for WSD. In: Agirre, E., Edmonds, P. (eds.) *Word Sense Disambiguation: Text, Speech and Language Technology*, vol. 33, pp. 133–166. Springer, New York (2006)
7. Hwang, M., Choi, C., Kim, P.: Automatic enrichment of semantic relation network and its application to word sense disambiguation. *IEEE Trans. Knowl. Data Eng.* **23**(6), 845–858 (2011)
8. Choi, D., Kim, P.: Identifying the most appropriate expansion of acronyms used in wikipedia text. *Softw. Pract. Experience* (2014). doi:[10.1002/spe.2006](https://doi.org/10.1002/spe.2006)