Highlights

**Word Sense Disambiguation using Knowledge-based Word Similarity**
Sunjae Kwon, Dongsuk Oh, Youngjoong Ko

- Propose a new context selection strategy for knowledge-based word sense disambiguation.
- Propose a novel approach to generate knowledge-based word vector representation.
- Compared with existing word sense disambiguation systems, our system shows the highest performance.
Word Sense Disambiguation using Knowledge-based Word Similarity*

Sunjae Kwon\textsuperscript{a,1}, Dongsuk Oh\textsuperscript{b,1} and Youngjoong Ko\textsuperscript{c,*}

\textsuperscript{a}Information \& Electronics Research Institute, KAIST 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
\textsuperscript{b}Human-inspired AI Computing Research Center, Korea University, Seoul 02841, Republic of Korea
\textsuperscript{c}N Center, Department of Applied Data Science and Department of Software Sungkyunkwan University (Natural Science Campus), 2066, Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Republic of Korea

\textbf{ARTICLE INFO}

\textbf{Keywords:}
Natural Language Processing
Word Sense Disambiguation
Knowledge-based Word Vector Representation
Similarity-based Word Selection

\textbf{ABSTRACT}

In natural language processing, word-sense disambiguation (WSD) is an open problem concerned with identifying the correct sense of words in a particular context. To address this problem, we introduce a novel knowledge-based WSD system. We suggest the adoption of two methods in our system. First, we suggest a novel method to encode the word vector representation by considering the graphical semantic relationships from the lexical knowledge-base. Second, we propose a method for extracting the contextual words from the text for analyzing an ambiguous word based on the similarity of word vector representations. To validate the effectiveness of our WSD system, we conducted experiments on the five benchmark English WSD corpora (Senseval-02, Senseval-03, SemEval-07, SemEval-13, and SemEval-15). The obtained results demonstrated that the suggested methods significantly enhanced the WSD performance. Furthermore, our system outperformed the existing knowledge-based WSD systems and showed a performance comparable to that of the state-of-the-art supervised WSD systems.

\section{1. Introduction}

In natural language, a word can signify different concepts based on its context. For example, the word ‘star,’ according to the Oxford dictionary (Dictionaries, 2019-04), has 17 different meanings or ‘senses.’ Each sense of the word can be mapped to a certain concept. For example, in the sentence “He always wanted to be a Hollywood star.”, the word ‘star’ can be defined as “a very famous or talented entertainer or sportsperson.” On the contrary, ‘star’ in “The Milky Way galaxy contains between 200 and 400 billion stars.” implies “a fixed luminous point in the night sky, which is a large, remote and incandescent body like the Sun.” Humans can readily understand these senses by considering the collocated words and their context. This problem is called word-sense disambiguation (WSD), a subtask of natural language processing (NLP), which automatically categorizes words into their appropriate senses. Because WSD plays a critical role in the natural language understanding, there are numerous studies that have adopted the WSD module, including those on machine translation (Vickrey, Biewald, Teyssier and Koller, 2005; Gonzales, Mascarell and Sennrich, 2017), information extraction (Navigli, 2009), information retrieval (Kang, Na and Lee, 2004; Zhong and Ng, 2012) and so on.

There are two main approaches for WSD research: 1) supervised WSD, and 2) knowledge-based WSD. In the former, the correct sense of an ambiguous word is classified with a machine learning model trained with human annotated corpus. Especially, as this method is beneficial when a highly advanced neural network algorithm is used, however, there is a certain limitation on the number of words that can be handled because annotating all the words including coinages is cost intensive. (Chaplot and Salakhutdinov, 2018). Contrarily, the latter determines the actual sense of the ambiguous word by choosing one of the candidate senses in the lexical knowledge-bases (LKBs), such as...

\footnote{Corresponding author
youngjoong.ko@gmail.com (Y. Ko)
http://nlp.skku.edu (Y. Ko)
ORCID(s):
1 Both authors contributed equally to this work.}
WordNet (Miller, 1995) and BabelNet (Navigli and Ponzetto, 2012), to name a few. As knowledge-based WSD systems rarely depend on hand-labeled corpora, these systems are advantageous in that there are virtually no limitations to the number of words that can be analyzed. Thus, these systems are more pragmatic and are the commonly adopted knowledge-based WSD approaches (Moro, Raganato and Navigli, 2014; Chaplot and Salakhutdinov, 2018).

Generally, the knowledge-based WSD approaches define the answer sense of the ambiguous word by ranking the candidate senses on the LKBs (Navigli and Lapata, 2007; Mihalcea, 2007; Navigli and Lapata, 2010). One of the most established ways in knowledge-based WSD is the graph-based WSD approach that utilizes the graphical structure of the lexical knowledge-bases. Based on the contextual words, we can build a subgraph from the LKBs and rank the candidate senses by measuring the node connectivity of the connected subgraphs. In other words, the performance of the graph-based WSD approaches essentially relies on the heuristics of subgraph construction and graph node ranking. Previous studies on the graph-based WSD approaches have adopted the method of optimal search that concurrently analyzes all the ambiguous words (Navigli and Lapata, 2007). These approaches are beneficial as they allow us to obtain the optimal set among the candidate senses; however, this approach is extremely time consuming. In order to allay the time complexity problem, (Manion and Sainudiin, 2014) suggested an iterative subgraph reconstruction WSD mechanism that is a greedy algorithm-based sub-optimal searching approach. However, it still utilizes whole words in a document as contextual words for each ambiguous word so it may give rise to build over-complicated subgraph that contains unnecessary information.

In this paper, we attempt to resolve the aforementioned problem of the iterative subgraph reconstruction model. Our proposed WSD system is different from the previous ones in two ways. First, we suggest a novel word vector representation method using the graphical structure of LKBs. Specifically, we express the words as a set of senses and generate a vector representation of the words based on the subgraphs connecting the adjacent senses. Second, we suggest a method to refine the iterative subgraph reconstruction by selecting contextual words for analyzing ambiguous words. In this approach, the contextual words whose similarity with the target word is higher than the designated threshold, are selected, and this similarity is estimated by cosine similarity of the knowledge-based word vector representations (Lahitani, Permanasari and Setiawan, 2016).

The experiments were conducted on five different publicly accessible English WSD corpora (Senseval-02 (Palmer, Fellbaum, Cotton, Delfs and Dang, 2001), Senseval-03 (Snyder and Palmer, 2004), SemEval-07 (Pradhan, Loper, Diigach and Palmer, 2007), SemEval-13 (Navigli, Jurgens and Vannella, 2013) and SemEval-15 (Moro and Navigli, 2015)). For following a well-established experimental setting for the WSD of (Raganato, Camacho-Collados and Navigli, 2017), we used SemEval-07 as a development set and the other corpora as test sets. All of these five corpora are annotated with the WordNet inventory. Meanwhile, to demonstrate the versatility of the proposed WSD system, we conducted additional experiments using a SemEval-13 corpus tagged documents as the BabelNet sense inventory. Experimental results showed that the proposed methods significantly improved the performance of all the WSD corpora. In addition, comparison with the existing methods showed that our final WSD model outperformed the existing knowledge-based WSD models by achieving the best score on the Senseval-03, SemEval-13 and SemEval-15. In particular, our WSD system demonstrated successful performance in the case of ambiguous nouns. Finally, experimental results on the BabelNet annotated corpus indicated that the proposed system consistently achieves high performance regardless of the structure of the LKBs.

Our main contributions in this paper are as follows:

- We introduce a novel knowledge-based word vector representation that reflects semantic hierarchies of the words.
- We propose a new graph-based WSD system that selects context words of the target word by word similarity between the word vector representations.
- Our WSD system shows state-of-the-art performance on the several benchmarks. Especially, the results were comparable with the existing supervised WSD.

The rest of the paper is organized as follows. Section 2 provides a brief review of previous studies on WSD and LKBs. Section 3 describes the entire process of our WSD systems including our suggestions. The experimental settings and results are detailed in section 4. Conclusions and directions for future work are presented in section 5.
2. Related Work

In this section, we first present some important previous studies. Next, we concisely describe the structure and characteristics of BabelNet and WordNet, which are the LKBs used in this study for estimating the efficacy of our proposed system.

2.1. Word-sense Disambiguation

A classic method used for knowledge-based WSD is the Lesk algorithm (Lesk, 1986). In this approach, the definition of the sense of a word is compared with its context, and the sense with the maximum number of overlapping with the context is determined to be the correct answer. The Lesk algorithm has undergone finetuning in several subsequent studies. Kilgarriff and Rosenzweig (2000) introduced a simplified version of the original Lesk algorithm to overcome the time complexity problem. Banerjee and Pedersen (2003) improvised the Lesk algorithm using term weighting. Recently, Luo, Liu, Xia, Chang and Sui (2018) further suggested that for comparing sense definitions with context, the memory network module can be incorporated into a neural network-based supervised learning framework.

In general, Lesk algorithms disambiguate the sense of the ambiguous word by relying only on its context. Therefore, the ambiguous words are independently analyzed for their senses. In practice, however, as the sense of each word depends upon the senses of the other words (Chaplot, Bhattacharyya and Paranjape, 2015), it is essential to jointly determine the senses of all the words in the entire context. Graph-based WSD is an approach that states and analyzes an underlined contextual semantic structure by leveraging the semantic network of the LKB. Navigli and Lapata (2007) build the subgraph of the context and jointly optimize the graph connectivity of all the words. However, an optimal searching algorithm requires to exponentially increase time complexity that is far from practical approaches. Therefore, Manion and Sainudiin (2014) introduces an iterative subgraph reconstruction method for a greedy algorithm based WSD system. In particular, they tried to handle the uncertainty of words with a ‘Sudoku’ style WSD strategy to disambiguate the target words in an order of the number of their senses.

Meanwhile, using whole words in a document as the context of a target ambiguous word give rises not only the time exhausting but also performance degrading. Some heuristics that selectively construct the context of an ambiguous word were introduced. Agirre, de Lacalle and Soroa (2018) limit to the context with a window size of 20 words around an ambiguous word. Mihalcea (2005) limits the context as words in a same sentence. In addition, Chaplot et al. (2015) collects context words only from words in the same sentence and further decreases context using the dependency structures of the sentence.

Distributional word vector representations have been also widely exploited in previous WSD studies. Basile, Caputo and Semeraro (2014) have suggested to use word similarity calculated with the word vector representation constructed from a co-occurrences word matrix. Iacobacci, Pilehvar and Navigli (2016) compared several word vector representations using neural networks, derived from the textual context and knowledge base, because word vectors generated from texts have been shown better result. In Oele and van Noord (2018) adopted word vector representation to the Lesk algorithm comparing the context of a word and its candidate senses definitions.

Note that, our WSD system adopts the sub-optimal graph-based approach of Manion and Sainudiin (2014). However, different from their iterative subgraph reconstruction, we concentrate on the context selection for the target words. Especially, we suggest the context selection strategy by using threshold with word similarity. In addition, in order to effectively leverage our context selection strategy, we propose a method to generate word vector representations in a knowledge graph to determine word similarity.

2.2. Lexical Knowledge-bases

WordNet is an English LKBs and is the most widely used for the sense repository of the WSD systems (Navigli, 2009; Chaplot and Salakhutdinov, 2018; Agirre et al., 2018). It has a graphical structure with nodes that are comprised of nouns, verbs, adjectives, and adverbs grouped into sets of synonyms (synsets), each of which represents a distinct sense. Each synset contains a brief definition, called glossary, and examples of usage in the text. In this case, the synsets are interconnected with the semantic and lexical relationships.

BabelNet is also an LKB that was used in our experiments. It has a graphical structure similar to WordNet, but there are two main distinctive attributes unique to BabelNet. First of all, it is a multi-lingual LKBs covering more than 200 languages made up of babel synsets that provide multi-lingual synonyms. In addition, it is a large-scale semantic knowledge graph, and it is an integration of several LKBs including Wikipedia, WordNet and so on.

Notably, our proposed WSD system is mainly evaluated using WordNet. However, in order to clarify that our
approaches are not solely dependent on the WordNet structure, an additional evaluation using the BabelNet was performed.

3. Proposed WSD system

In this chapter, we explain the detailed structure and the methodologies adopted by our WSD system. Fig. 1 is an overview of our proposed WSD model. First, we introduce a novel approach for calculating word similarity with word vector representation from knowledge-graph that was exploited in our WSD system. Subsequently, we explain our iterative thresholded subgraph reconstruction approach for WSD detailing the way in which each word similarity information can be effectively combined in the graph-based WSD.

3.1. Word similarity calculation via word vector representation from a knowledge graph structure

Generally, words can be defined depending on the context that they appear in. However, not all contextual words can be considered to have the same significance when determining the meaning of a particular word. It means that if we can accurately estimate the importance of words for WSD task, then we can identify the contextual words required to analyze the given ambiguous word, thereby leading to a more sophisticated analysis.

In order to effectively calculate word similarity, we assume that the document has a common semantic theme, which in turn affects the choice of each word (Blei, Ng and Jordan, 2003). Additionally, we also assume that semantic relatedness, which is information contained in the knowledge graph, with ambiguous words are crucial for disambiguation. We also assume that the word is not by nature a set of senses, but has a more complex hierarchy composed of a set of semantic themes comprising similar senses in a word. Considering the abovementioned points, we suggest the following method to train the knowledge graph-based word vector representation.

Fig. 2 depicts an example of a method for extracting information, by explaining the semantic meaning of a word, on the knowledge graph interconnecting the various senses of the word. In the example Fig. 2 (a), there are four divergent senses for the word ‘star’. Star#1 indicates “A celestial body of hot gases.”, Star#2 means “Any celestial body visible from the Earth at night.”, Star#3 denotes “An actor who plays a principal role.”, and Star#4 is “A person who is widely known.”. To interlock the senses, we extend and surround them by 2-level depth as illustrated in Fig. 2 (b). Subsequently, interlinked senses are gathered into a single subgraph where the senses share a common semantic theme. In the example, star#1 and star#2 are interconnected to form a theme related to astronomical phenomena, and star#3 and star#4 can be clustered into a theme related to celebrities. Next, as illustrated in Fig. 2 (c), in order to conveniently handle these themed subgraphs, we represent the subgraphs into a sequence of semantic relationships using a graph search algorithm, and we call this as a semantic relational path of the theme. Any graph search algorithms can be used to extract the semantic relational path. In our paper, we adopt the depth first search (DFS) and breadth first search (BFS) algorithms (Kozen, 1992). Finally, the semantic relational paths of the themes are concatenated to form
Figure 2: An example of the semantic relational path of the word ‘star’. Figure a represents the initial state. Figure b illustrates the subgraphs of themes by extending 2-level depth from each of the senses of the word. Figure c denotes the semantic relational path of the word extracted using the DFS algorithm. In the figure, ellipses and arrows denote the concepts and relations, respectively.

a semantic relational path of the word.

Through the abovementioned process, we realize the semantic relational path that expresses a word using the elements of the knowledge graph. This path represents a word hierarchical structure that has the following three layers: 1) semantic triples that involves two senses (subject and object) and relationship between the nodes, 2) subgraph of the sense connected with the other senses with the same context, and 3) the words that are represented in the form of a set of subgraphs. In this case, if we assume the relationship to be a word, subgraph to be a sentence, and the set of subgraphs to be a document, then the semantic relational path of a word can be considered to be a pseudo document. In our system, we can obtain the word vector representation of words by encoding the pseudo documents that are in the knowledge-graph with a neural network, and we use the Doc2Vec algorithm (Le and Mikolov, 2014), which is known to be effective for document encoding. With the Doc2Vec algorithm, we map the pseudo documents into a distributional vector space and cluster similar pseudo documents into a peripheral space. In other words, if there are several concepts overlapping the semantic relational path of the words, the vector representation of the words will be analogous. We estimate the similarity the words through the cosine similarity of $w_1$ and $w_2$ as in Eq. 1.

$$\text{cosine_similarity}(w_1, w_2) = \frac{w_1^T \cdot w_2}{||w_1|| \times ||w_2||}$$

3.2. Iterative Thresholded Subgraph Reconstruction

To determine the correct sense of ambiguous words in a graph-based WSD approach, establishing an efficient strategy that builds subgraph and ranks candidate senses of the target word plays an important role. In our research,
we introduce a new graph-based WSD algorithm, iterative thresholded subgraph reconstruction, fabricating a set of contextual words for the target word by thresholding words in a document based on the similarity to the ambiguous word.

There are two steps for our WSD algorithm of iterative thresholded subgraph reconstruction. In Algorithm 3.2, the preprocessing step is line 1 to 3 and the answer selection step is line 4 to 14.

**Preprocessing:** In preprocessing step, as line 1, we first tokenize and lemmatize the words in the input document doc. Subsequently, ambiguous words W are searched from LKB K (line 2). Lastly, as the words with only one sense defined in LKB are not required to be analyzed, we initialize Answer, an answer set, with the ambiguous words that are composed of only one sense.

**Answer selection:** In the answer selection step, we disambiguate each word sequentially. Specifically, we analyze the ambiguous words left to right order according to a sequence of texts (line 4 to 12). Firstly, contextual words Ci of the ith ambiguous wi are selected by thresholding with word similarity and, the words that have not been analyzed yet are Ci, and the words already analyzed are Ai (line 5 and 6). After then, we get the candidate senses, Swi, from the ambiguous word wi, and senses of the contextual words Sc_i in line 7 and 8. Next, we extend entire senses Swi, Sc_i and Ai with 2 level depth building subgraph Gi that represents current state (line 9). In the subgraph, the Swi is a set the senses we want to disambiguate, the Sc_i is a set of senses that express context and the Ai is a set of senses that fixed in the previous state 0 to i − 1. To determine what sense is the most appropriate one among the Swi considering current state, we adopt the modified PPR of Eq. 2 introduced in (Agirre et al., 2018) (line 10).

\[
\text{Pr} = c M \text{Pr} + (1 - c) v
\]  

\[\text{(2)}\]

In the equation, Pr indicates a page rank vector over the nodes v1, ..., vN of Gi, it embodies the importance of each node and is calculated by using the following steps. At first, Pr is initialized uniformly over the nodes. In the first term of right-hand side of the equation, c is a damping factor and M is an N × N transition matrix, where if ith and jth nodes are interconnected and the out-degree of ith is di then M_{ij} = 1/di. The second term is a smoothing term that ensures page rank converge. Here, v is a personalized normalizing vector that is uniformly assigned as 1/N in the original page rank but we non-uniformly assign a value to the nodes. Instead, we only provide values to the senses of words that can be directly found in the context, and the value of the other senses are set to zero. This makes the contextual words more influential while analyzing the target word. Pr is iteratively updated and stably converged.

Using PPR, we can estimate the mass over the candidate senses and identify an optimal sense \(\hat{s}^*\) that has the highest mass among the candidates (line 11). The \(\hat{s}^*\) is added into the Answer (line 12) and (line 4) is executed as the next step \(i + 1\). A series of operations iterate until the last ambiguous word is analyzed, and finally, the answer Answer is returned.

**Algorithm 1** Pseudo code for the ITSR

**Require:** An input document (Doc), knowledge-graph (K) and word vector representation (E)

1: \(L \leftarrow \text{Lemmatizer(Doc)}\)
2: \(W \leftarrow \text{GetAmbiguousWords}(L, K)\)
3: \(\text{Answer} \leftarrow \text{InitializeAnswerSet}(W, K)\)
4: for \(w_i\) in \(W\) do
5: \(C_i \leftarrow \text{SelectContext}(w_i, W - \text{Answer}, E)\)
6: \(A_i \leftarrow \text{SelectContext}(w_i, \text{Answer}, E)\)
7: \(S_{w_i} \leftarrow \text{GetSenses}(w_i)\)
8: \(S_{C_i} \leftarrow \text{GetSenses}(C_i)\)
9: \(G_i \leftarrow \text{ConstructSubgraph}(S_{w_i}, S_{C_i}, A_i)\)
10: \(\text{Pr} \leftarrow \text{PPR}(G_i)\)
11: \(\hat{s}^* \leftarrow \max_{s_j \in S_{w_i}} \text{Pr}(s_j)\)
12: \(\text{Answer} \leftarrow \text{Answer} \cup \hat{s}^*\)
13: end for
14: \(\text{Return} \text{Answer}\)
Table 1
of each part-of-speeches. In this table, N, V, ADJ and ADV indicates Nouns, Verbs, Adjectives, and Adverbs, respectively.

| Corpus     | LKBs     | N   | V   | ADJ | ADV |
|------------|----------|-----|-----|-----|-----|
| Senseval-02| WordNet  | 1,066| 517 | 445 | 254 |
| Senseval-03| WordNet  | 900 | 588 | 350 | 12  |
| SemEval-07 | WordNet  | 159 | 296 | -   | -   |
| SemEval-13 | WordNet  | 1,644| -   | -   | -   |
|            | BabelNet | 1,931| -   | -   | -   |
| SemEval-15 | WordNet  | 531 | 251 | 160 | 80  |
| Overall    | WordNet  | 3,769| 1,652| 955 | 346 |

4. Experiments
4.1. Experimental Corpora
In order to evaluate our WSD system, we prepared five different benchmark English all-word WSD corpora as presented in Table 1. In addition, to verify that the proposed WSD system can be exploited in various knowledge-bases, we used SemEval-13 corpus tagged data with both WordNet and BabelNet inventories. Finally, in our experiments, following Raganato et al. (2017)’s settings, we used the SemEval-07 corpus as a development set. The following descriptions are the details of the corpora we used.

- **Senseval-02**: Senseval-02 is composed of 2,282 annotated senses coming from WordNet 1.7 and the corpus includes nouns, verbs, adverbs and adjectives.
- **Senseval-03**: This corpus consists of three documents from three different domains. It annotated with WordNet 1.7.1 and has total 1,850 annotated senses.
- **SemEval-07**: SemEval-07 consist of three documents with 455 senses including only nouns and verbs. The senses in this corpus comes from WordNet 2.1 sense inventory.
- **SemEval-13**: This dataset includes 13 news articles having both of WordNet and BabelNet dataset. The WordNet annotated data contains 1,644 nouns and the BabelNet annotated data has 1,931 nouns. The senses on this corpus comes from the WordNet 3.0 sense inventory and the BabelNet 1.1.1 sense inventory.
- **SemEval-15**: This corpus has four documents from 3 different domains. The corpus also has data annotated with both the WordNet 3.0 inventory and it contains 1,022 ambiguous words.

As mentioned previously in the Section 2, as WordNet is a subset of BabelNet, the difference between the two datasets mainly arises due to the Wikipedia.

4.2. Experimental Settings
For all the experiments, we applied the following setting to train the word vector representation introduced in the Section 3.1. We trained our word vector representation based on BabelNet 2.5 as it covers all the WSD corpora exploited in the experiments. In addition, we adopted gensim Doc2Vec library\(^2\). Window size and learning rate were set to 3 and 0.025, respectively. Additionally, the vector dimension was assigned to 200, and all the other hyper-parameters were set to default. Finally, the word similarity threshold of Section 3.2 was set to 0.45.

For assessment, we utilized $F_1$-score criteria of an Eq 5. The $F_1$-score is a harmonic mean of the recall of an Eq 4 and precision of an Eq 3. We used an official code for the performance estimation of our proposed methods. \(^3\)

\[
Precision = \frac{\text{# of true positive ambiguous words}}{\text{# of outcome positive ambiguous words}}
\]  

---

\(^2\)https://radimrehurek.com/gensim/models/doc2vec.html  
\(^3\)http://nlp.lsi.upc.edu/tools/download-map.php
Table 2
Experimental results of the effect of the proposed method on the WordNet annotated WSD corpora. The terms ‘Dev.,” ‘Prec.,” and ‘Rec.” indicate development, precision, and recall, respectively.

| Models        | SemEval-07 | Senseval-02 | Senseval-03 | SemEval-13 | SemEval-15 |
|---------------|------------|-------------|-------------|------------|------------|
|               | Prec. | Rec. | \(F_1\) | Prec. | Rec. | \(F_1\) | Prec. | Rec. | \(F_1\) | Prec. | Rec. | \(F_1\) |
| ISR           | 53.7 | 53.6 | 53.7 | 70.4 | 68.1 | 69.2 | 64.8 | 64.8 | 64.8 | 66.5 | 66.5 | 66.5 |
| ITSR+Word2Vec | 48.5 | 48.4 | 48.4 | 68.3 | 37.4 | 48.3 | 69.1 | 69.1 | 69.1 | 68.8 | 68.8 | 68.8 |
| ITSR+SRP<sub>DFS</sub> | 56.0 | 56.0 | 56.0 | 67.9 | 55.4 | 61.0 | 70.9 | 70.9 | 70.9 | 72.5 | 72.5 | 72.5 |
| ITSR+SRP<sub>BFS</sub> | 57.6 | 57.6 | 57.6 | 79.0 | 66.9 | 72.4 | 77.1 | 66.3 | 71.3 | 73.6 | 73.6 | 73.6 |

Recall = \(\frac{\# of\ true\ positive\ ambiguous\ words}{\# of\ true\ ambiguous\ words}\) \hspace{1cm} (4)

\(F_1\) score = \(2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\) \hspace{1cm} (5)

We also performed a student \(t\)-test (Manning, Raghavan and Schütze, 2010) to verify that the results are significantly different.

4.3. Experimental Results
4.3.1. Impacts of Proposed Methods
To verify the effects of the proposed methods, we performed the experiments in the following four environments.

- **ISR**: It is a baseline iterative subgraph reconstruction (ISR) based WSD (Manion and Sainudiin, 2014). Notably, because we have reimplemented the ISR WSD system, the results may deviate from those of the original paper.

- **ITSR+Word2Vec**: In this environment, the iterative thresholded subgraph reconstruction (ITSR) method of section 3.2 is used. In addition, an official version of Word2Vec\(^4\), known as the most suitable for WSD among existing word vector representations (Iacobacci et al., 2016), is adopted for the word similarity measurement.

- **ITSR+SRP<sub>DFS</sub>**: In this environment, the ITSR method is used. For the word similarity measurement, we adopted SRP2Vec of section 3.1 and searched the subgraph of SRP with the DFS algorithm.

- **ITSR+SRP<sub>BFS</sub>**: In this environment, the ITSR method is used. For the word similarity measurement, we adopted SRP2Vec of section 3.1 and searched the subgraph of SRP with the BFS algorithm.

Table 2 indicates performance of the ISR, the baseline model, and the proposed methods in WordNet environment. Experimental results showed that the performances of our final model, ITSR+SRP<sub>BFS</sub>, was significantly superior to ISR \((p < 0.01)\) for the entire corpora. This implies that by limiting the contextual words based on word similarity,

\(^4\)https://github.com/mmihaltz/word2vec-GoogleNews-vectors
we can eliminate the unnecessary information from a document. Furthermore, in the case of the ITSR+Word2Vec environment, its performance was seen to improve in Senseval-03, SemEval-13, and SemEval-15 corpora, while in the Senseval-02 corpus, the performance was degraded. In contrast, ITSR+SRP$_{BFS}$ and ITSR+SRP$_{DFS}$ demonstrated superior performances than ITSR+Word2Vec in all the corpora ($p < 0.05$). From these results, we can notice that when limiting the context words with word similarity it is better choice to consider only semantic information. Finally, the SRP vector representation was seen to be affected by the graph search algorithm, and the BFS algorithm was found to be able to produce higher quality vector representation than the DFS algorithm.

Additional experiments on the BabelNet annotated corpus are presented in Table 3. The ITSR model achieved a similar outcome with the WordNet annotated corpora showing a significantly enhanced performance than the ISR model ($p < 0.01$). From these results, we can see that our ITSR can effectively be applied not only to the WordNet knowledge graph but also to more complex knowledge graphs. Particularly, we can observe that the results of the ITSR models showed higher performance in the BabelNet annotated corpus, while ISR did not show any significant difference in the results between the WordNet and BabelNet annotated corpora. As a difference between these two corpora is originated from use of the Wikipedia sense repository, it can conclude that our model is more powerful in the case of proper nouns such as named entities.

4.3.2. Results of Comparison Tests

In order to attest the competitiveness of our model, we have compared the performance with the state-of-the-art knowledge-based and supervised WSD models.

First, we prepared the following knowledge-based WSD models. The idiosyncrasies of each model are briefly described as below.

- **Banerjee 03**: This a model suggested by Banerjee and Pedersen (2003) using a Lesk algorithm based WSD approach. Furthermore, it also regards conventional term frequency and inverse document frequency term weighting scheme (Salton and McGill, 1983).

- **Basile 14**: This is an enhanced model of Lesk algorithm where word vector representation is additionally exploited to calculate similarity between target word and definition of a dictionary (Basile et al., 2014).

- **Moro 14**: This model is a unified approach for WSD and entity linking (Moro et al., 2014). Notably, the model creates the semantic signature of ambiguous words utilizing whole text and subgraph structure calculating node importance by random walk and restart (Tong, Faloutsos and Pan, 2006).

- **Tripodi 17**: Tripodi and Pelillo (2017) adopted evolutionary game theory to calculate node importance. Especially, they design WSD as a constraint satisfaction problem deriving it exploiting game theorem tools.

- **Chaplot 18**: This model is a divergent of the Latent Dirichlet Allocation (LDA) (Blei et al., 2003) but instead of demonstrating document by its topics the model manifesting document probabilities of synsets (Chaplot and Salakhutdinov, 2018).

- **Agirre 18**: The model is a graph-based WSD model (Agirre et al., 2018). It defines the context of a word with a certain window size, i.e. 20 and calculates the graph connectivity with PPR.

In addition, we compared the performance with following supervised models.

- **Zhong 10**: This model was suggested in (Zhong and Ng, 2010). In feature extraction, it uses several text analyzers to prudently extract hand-crafted linguistic features that include surrounding POS tags, words, and local collocations. The extracted features are trained with a linear classifier (Fan, Chang, Hsieh, Wang and Lin, 2008).

- **Weissenborn 15**: This model jointly optimized both WSD and entity linking (Weissenborn, Hennig, Xu and Uszkoreit, 2015). In addition, during the disambiguation process, it determines the answer senses of the ambiguous words by calculating the optimal set over the candidate senses.

- **Iacobacci 16**: This model was suggested in (Iacobacci et al., 2016). In this study, the additional word vector representation features was used on the Zhong 10. The authors evaluated several word vector representation models and the model with Word2Vec showed the best result.
Table 4

$F_1$-score comparison of our model and state-of-the-art WSD systems on the WordNet annotated corpora. Note that, the SemEval-07 corpus was used for validation set in our paper and it is hard to directly compare with the results of the other systems.

| Systems   | Senseval-02 | Senseval-03 | SemEval-07* | SemEval-13 | SemEval-15 |
|-----------|-------------|-------------|-------------|-------------|-------------|
| Supervised|             |             |             |             |             |
| Zhong 10  | 72.8        | 69.2        | 60.0        | 65.0        | 69.3        |
| Iacobacci 16 | 72.2    | 70.4        | 62.6        | 65.9        | 71.5        |
| Raganato 17 | 72.0    | 69.1        | **64.8**    | 66.9        | 71.5        |
| Peters 18  | 71.6        | 69.6        | 62.2        | 66.2        | 71.3        |
| Luo 18     | 72.2        | 70.5        | -           | 67.2        | 72.6        |
| Knowledge-based |       |             |             |             |             |
| Banerjee 03 | 50.6    | 44.5        | 32.0        | 53.6        | 51.0        |
| Basile 14  | 63.0        | 63.7        | 56.7        | 66.2        | 64.6        |
| Moro 14    | 67.0        | 63.5        | 51.6        | 66.4        | 70.3        |
| Chaplot 18 | 69.0        | 66.9        | 55.6        | 65.3        | 69.6        |
| Agirre 18  | 68.8        | 66.1        | 53.0        | 68.8        | 70.3        |
| ITSR+SRP$_{BFS}$ | 72.4 | **71.3** | *57.6* | **73.6** | **73.5** |

Table 5
Experimental results on comparison test between our model and state-of-the-art WSD systems on the SemEval-13 BabelNet annotated corpus.

| Systems   | SemEval-13 |
|-----------|-------------|
| Supervised|             |
| Zhong 10  | 66.3        |
| Weissonborn 15 | 71.5 |
| Knowledge-based |       |             |
| Moro 14    | 69.2        |
| Tripodi 17 | 70.8        |
| ITSR+SRP$_{BFS}$ | **78.5** |

- **Raganato 17**: Raganato et al. (2017) suggested long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) encoder-decoder based WSD model. Particularly, the model jointly optimizes the main task (WSD) and auxiliary tasks (POS tags and semantic labels) to leverage an advantage of multi-task learning (Alonso and Plank, 2016).

- **Peters 18**: This model is a semi-supervised model (Peters, Neumann, Iyyer, Gardner, Clark, Lee and Zettlemoyer, 2018), comprising of two steps. First, a bi-directional LSTM based neural language model was trained on an unlabeled corpus. Next, a contextual vector was learned for WSD task with a supervised approach.

- **Luo 18**: This model is comprised of the LSTM based contextual WSD classifier module and the neural glossary memory mechanism based WSD classifier module (Luo et al., 2018). Each module respectively calculates the probability distribution of the candidate senses and these distributions are then linearly combined by deriving the final result for WSD.

Table 4 lists the experimental results of the comparison tests on the WordNet annotated corpora. The results show that our WSD model ITSR+SRP$_{BFS}$ outperformed the existing knowledge-based WSD models in all of the test set corpora with a huge margin (more than 3%p). Particularly, in Senseval-03, SemEval-13 and SemEval-15, ITSR+SRP$_{BFS}$ surpassed the state-of-the-art supervised WSD models showing the best performance. Besides, our model showed comparable performance to the existing supervised models in Senseval-02. Finally, from the results on Table 5, we can also see that ITSR+SRP$_{BFS}$ accomplished the best performance on the SemEval-13 BabelNet annotated corpus.

4.4. Discussion

Here, we will analyze the results of each part of speech (POS) tag and the error case from the model. In addition, we provide an error analysis of our model.
Table 6

$F_1$-score comparison on the different POS tags in the WordNet annotated corpora. In this table, N, V, ADJ, ADV, and All indicates Nouns, Verbs, Adjectives, Adverbs, and entire ambiguous words, respectively.

| Systems            | N   | V   | ADJ  | ADV  | All  |
|--------------------|-----|-----|------|------|------|
| Supervised         |     |     |      |      |      |
| Zhong 10           | 71.0| 53.3| 77.1 | 82.7 | 68.3 |
| Iacobacci 16       | 71.9| 56.6| 75.9 | 84.7 | 70.1 |
| Raganato 17        | 71.5| 57.5| 75.0 | 83.8 | 69.9 |
| Luo 18             | 72.2| 57.7| 76.6 | 85.0 | 70.6 |
| Knowledge-based    |     |     |      |      |      |
| Banerjee 03        | 54.1| 27.9| 54.6 | 60.3 | 48.7 |
| Basile 14          | 69.8| 51.2| 51.7 | 80.6 | 63.7 |
| Moro 14            | 68.6| 49.9| 73.2 | 79.8 | 65.5 |
| Chaplot 18         | 69.7| 51.2| 76.0 | 80.9 | 66.9 |
| ITSR+SRP$_{BFS}$   | 76.7| 62.8| 60.7 | 69.8 | 71.6 |

Figure 3: An example sentence for an error propagation case. The ambiguous words are represented with special characters ‘[ ]’ and the answer senses are denoted with red color.

4.4.1. Experimental Results on Each POS Tag

Table 6 indicates the experimental results of each POS on the WordNet annotated corpora (SenseEval-02, SenseEval-03, SemEval-07, SemEval-13 and SemEval-15). From these results, it is observed that our model, ITSR+SRP$_{DFS}$, demonstrated the best result in the case of nouns, the performance of our model in the case of verbs superior to that of the state-of-the-art comparison models, and, however, it was inferior as compared to the baseline models. In addition, our model also showed inferior performance than the comparative models in the case of adjectives and adverbs. However, the overall results show that our model achieved the best score.

From our experiment, the following conclusions can be inferred. First, our model showed successful performance on the noun ambiguous words that account for about 56% of all the ambiguous words in the WordNet annotated corpora. Contrarily, it can be seen that syntactic information needs to be considered for adjectives and adverbs, which are used as modifiers in text. Thus, integrating a statistical language model (Bellegarda, 2004) that probabilistic distribution on the co-occurrence information over sequences of words to our WSD system can be a feasible solution.

4.4.2. Error Analysis

Although the effectiveness of our proposed system was authenticated, there still exists an intrinsic weakness that arises from greedy algorithmic peculiarity of the iterative subgraph reconstruction mechanism. In other words, erroneously disambiguated words can have detrimental effect on the words that are yet to be analyzed.

Fig. 3 represents an example for the aforementioned issue. In this example, there are six ambiguous words, (‘Alimta,’ ‘powder,’ ‘made_up,’ ‘solution,’ ‘infusion,’ and ‘vein’), and our model performs disambiguation in an order starting from ‘Alimta’ to ‘vein’. The first incorrectly analyzed word was ‘made_up’ and it was analyzed as ‘Apply
make-up or cosmetics to one’s face to appear prettier.” that was prompted by the sense of the word ‘powder’. Moreover, the selected answers of ‘made_up’ and ‘solution’ lead to the selection of ‘infusion’ to misinterpret the meaning of ‘A solution obtained by steeping or soaking a substance’. If the word ‘vein’ would have been considered, it would have been possible to accurately analyze the words.

The error propagation is a typical common issue in the greedy search. Beam search (Socher, Lin, Manning and Ng, 2011) that finds N best candidates for each time step is a widely used solution to alleviate the error propagation and it also can be a feasible solution for our WSD system.

5. Conclusion

This paper introduced an iterative thresholded subgraph reconstruction strategy for application in knowledge-based WSD systems. In this strategy, the contextual words for an ambiguous word are selected using word similarity. Moreover, to further strengthen our strategy, we suggested a novel way to calculate the word similarity by training the knowledge graph structure. Experiments were conducted on a publicly accessible English WSD corpora and our model demonstrated a significant improvement in performance against the baseline model of (Manion and Sainudiin, 2014). Moreover, our model outperformed the existing knowledge-based WSD systems in the test sets and demonstrated results comparable to the state-of-the-art supervised WSD systems. In particular, our model showed great performance when the target words were nouns and verbs.

Nonetheless, there were several limitations that were not solved yet. Firstly, the performance of adverb and adjectives still leaves room for improvement. Since a word occurs in different contexts of each sentence, we could achieve improvement by utilizing co-occurrence information in texts. In addition, our model suffered from error propagation because it has greedy algorithmic characteristics. A beam search algorithm could be a solution although it has increased search space.

In the future, we plan to integrate both supervised and knowledge-based approaches. More specifically, instead of heuristic graph search rules exploited in the knowledge-based WSD, we will attempt to train these rules with supervised approaches such like a REINFORCE learning algorithm (Li, 2017).

6. Acknowledgement

This work was supported by Institute for Information communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) [No. 2017-0-00550, Development of Automatic Text Data Labeling and Verification Tools for Machine Learning].

References

Agirre, E., de Lacalle, O.L., Soroa, A., 2018. The risk of sub-optimal use of open source nlp software: Ukb is inadvertently state-of-the-art in knowledge-based wsd, in: Proceedings of Workshop for NLP Open Source Software (NLP-OSS), pp. 29–33.
Alonso, H.M., Plank, B., 2016. When is multitask learning effective? semantic sequence prediction under varying data conditions. arXiv preprint arXiv:1612.02251 .
Banerjee, S., Pedersen, T., 2003. Extended gloss overlaps as a measure of semantic relatedness, in: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 805–810.
Basile, P., Caputo, A., Semeraro, G., 2014. An enhanced lesk word sense disambiguation algorithm through a distributional semantic model, in: Proceedings of the International Conference on Computational Linguistics: Technical Papers, pp. 1591–1600.
Bellegarda, J.R., 2004. Statistical language model adaptation: review and perspectives. Speech communication 42, 93–108.
Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. Journal of Machine Learning research 3, 993–1022.
Chaplot, D.S., Bhattacharyya, P., Paranjape, A., 2015. Unsupervised word sense disambiguation using markov random field and dependency parser, in: Proceedings of the Association for the Advancement of Artificial Intelligence.
Chaplot, D.S., Salakhutdinov, R., 2018. Knowledge-based word sense disambiguation using topic models, in: Proceedings of the Association for the Advancement of Artificial Intelligence.
Dictionaries, O., 2019-04. Oxford english dictionary. URL: https://en.oxforddictionaries.com/definition/star.
Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J., 2008. Liblinear: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874.
Gonzales, A.R., Mascarell, L., Sennrich, R., 2017. Improving word sense disambiguation in neural machine translation with sense embeddings, in: Proceedings of the Second Conference on Machine Translation, pp. 11–19.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735–1780.
Iacobacci, I., Pilehvar, M.T., Navigli, R., 2016. Embeddings for word sense disambiguation: An evaluation study, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 897–907.
Word Sense Disambiguation using Knowledge-based Word Similarity

Kang, I.S., Na, S.H., Lee, J.H., 2004. Influence of wsd on cross-language information retrieval, in: Proceedings of the International Conference on Natural Language Processing, Springer. pp. 358–366.

Kitagiriff, A., Rosenzweig, J., 2000. Framework and results for english senseval. Computers and the Humanities 34, 15–48.

Kozen, D.C., 1992. Depth-first and breadth-first search, in: The Design and Analysis of Algorithms. Springer, pp. 19–24.

Lahtatani, A.R., Permanasari, A.E., Setiawan, N.A., 2016. Cosine similarity to determine similarity measure: Study case in online essay assessment, in: 2016 4th International Conference on Cyber and IT Service Management, IEEE. pp. 1–6.

Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents, in: Proceedings of the International Conference on Machine Learning, pp. 1188–1196.

Lesk, M., 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, in: Proceedings of the Annual International Conference on Systems Documentation, pp. 24–26.

Li, Y., 2017. Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274.

Luo, F., Liu, T., Xia, Q., Chang, B., Sui, Z., 2018. Incorporating glosses into neural word sense disambiguation, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 2473–2482.

Manion, S.L., Saimudin, R., 2014. An iterative sudoku style approach to subgraph-based word sense disambiguation, in: Proceedings of the Joint Conference on Lexical and Computational Semantics, pp. 40–50.

Manning, C., Raghavan, P., Schütze, H., 2010. Introduction to information retrieval. Natural Language Engineering 16, 100–103.

Mihalcea, R., 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling, in: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 411–418.

Mihalcea, R., 2007. Knowledge-based methods for wsd, in: Word sense disambiguation. Springer, pp. 107–131.

Miller, G.A., 1995. Wordnet: A lexical database for english. Communications Of The ACM 38, 39.

Moro, A., Navigli, R., 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking, in: Proceedings of the international workshop on semantic evaluation (Semeval 2015), pp. 288–297.

Moro, A., Raganato, A., Navigli, R., 2014. Entity linking meets word sense disambiguation: a unified approach. Transactions of the Association for Computational Linguistics 2, 231–244.

Navigli, R., 2009. Word sense disambiguation: A survey. ACM computing surveys 41, 10.

Navigli, R., Jurgens, D., Vannella, D., 2013. Semeval-2013 task 12: Multilingual word sense disambiguation, in: Proceedings of the Seventh International Workshop on Semantic Evaluation, pp. 222–231.

Navigli, R., Lapata, M., 2007. Graph connectivity measures for unsupervised word sense disambiguation, in: Proceedings of the International Joint Conference on Artificial intelligence, Morgan Kaufmann Publishers Inc., pp. 1683–1688.

Navigli, R., Lapata, M., 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 678–692.

Navigli, R., Poncetta, S.P., 2012. Babelfnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence 193, 217–250.

O, D., Kwon, S., Kim, K., Ko, Y., 2018. Word sense disambiguation based on word similarity calculation using word vector representation from a knowledge-based graph, in: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2704–2714.

Oele, D., van Noord, G., 2018. Simple embedding-based word sense disambiguation, in: Proceedings of the Global WordNet Conference, p. 262.

Palmer, M., Fellbaum, C., Cotton, S., Delsi, L., Dang, H.T., 2001. English tasks: All-words and verb lexical sample, in: Proceedings of the International Workshop on Evaluating Disambiguation Systems, pp. 21–24.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations, in: Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237.

Pradhan, S., Loper, E., Dligach, D., Palmer, M., 2007. Semeval-2007 task-17: English lexical sample, srl and all words, in: Proceedings of the International Workshop on Semantic Evaluation, pp. 87–92.

Raganato, A., Camacho-Collados, J., Navigli, R., 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison, in: Proceedings of the the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 99–110.

Salton, G., McGill, M.J., 1983. Introduction to modern information retrieval.

Snyder, B., Palmer, M., 2004. The english all-words task, in: Proceedings of the International Workshop on Semantic Evaluations, pp. 222–231.

Socher, R., Lin, C.C., Manning, C., Ng, A.Y., 2011. Parsing natural scenes and natural language with recursive neural networks, in: Proceedings of the International Conference on Machine Learning, pp. 129–136.

Tong, H., Faloutsos, C., Pan, J.Y., 2006. Fast random walk with restart and its applications, in: Proceedings of the International Conference on Data Mining, IEEE. pp. 613–622.

Trippodi, R., Pelillo, M., 2017. A game-theoretic approach to word sense disambiguation. Computational Linguistics 43, 31–70.

Vickrey, D., Biewald, L., Teyssier, M., Koller, D., 2005. Word-sense disambiguation for machine translation, in: Proceedings of the Human Language Technology and Empirical Methods in Natural Language Processing, pp. 771–778.

Weissenborn, D., Hennig, L., Xu, F., Uszkoreit, H., 2015. Multi-objective optimization for the joint disambiguation of nouns and named entities, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 596–605.

Zhong, Z., Ng, H.T., 2010. It makes sense: A wide-coverage word sense disambiguation system for free text, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics System Demonstrations, pp. 78–83.

Zhong, Z., Ng, H.T., 2012. Word sense disambiguation improves information retrieval, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pp. 273–282.