Unsupervised Word Sense Disambiguation based on Word Embedding and Collocation

Shangzhuang Han and Kiyoaki Shirai
School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology,
Asahidai 1-1, Nomi, Ishikawa, Japan

Keywords: Unsupervised Machine Learning, Word Sense Disambiguation, Word Embedding, Collocation.

Abstract: This paper proposes a novel unsupervised word sense disambiguation (WSD) method. It utilizes two useful features for WSD. One is contextual information of a target word. The similarity between words in a context and a sense of a target word is measured based on the pre-trained word embedding, then the most similar sense to the context is chosen. Furthermore, we introduce a procedure not to use irrelevant words in a context in a calculation of the similarity. The other is a collocation, which is an idiomatic phrase including a target word. High-precision rules to determine a sense by a collocation is automatically acquired from a raw corpus. Finally, the above two methods are integrated into our final WSD system. Results of the experiments using Senseval-3 English lexical sample task showed that our proposed method could improve the precision by 4.7 point against the baseline.

1 INTRODUCTION

Word Sense Disambiguation (WSD) is a fundamental task and long-standing challenge in Natural Language Processing (NLP), which aims to determine a sense of an ambiguous word in a particular context (Navigli, 2009). The WSD approaches can be grouped into two main categories: methods based on supervised machine learning (supervised methods) and knowledge-based methods (unsupervised methods).

Supervised methods often train a classifier or neural network model from a sense tagged corpus (e.g. SemCor corpus). In recent years, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) has often achieved the state-of-the-art performance in many NLP tasks. Since BERT can classify a pair of sentences into predefined categories, Huang et al. proposed the GlossBERT, which constructs context-gloss pairs from all possible senses of the target word in WordNet, then treats the WSD task as a sentence-pair classification problem (Huang et al., 2019). Although supervised methods tend to achieve good performance, sense tagged corpora are required for training. Obviously, they are hard to construct due to heavy manual annotation.

Knowledge-based WSD methods rely on lexical resources like a dictionary or WordNet (Miller, 1995) rather than sense tagged corpora. A gloss, which defines a meaning of a word, is first utilized in Lesk algorithm (Lesk, 1986). Given a word and its context, Lesk algorithm calculates a score of each sense by measuring the number of overlapped words in a gloss (definition) of a sense of a target word and that of words in a context. Then, the sense with the highest score is chosen. A lot of studies follow it and propose its extended models. In addition to methods using gloss sentences, a graph-based WSD method is also investigated (Navigli and Lapata, 2009). In this approach, graph nodes correspond to word senses, whereas edges represent dependencies between senses (e.g. synonymy and antonymy). Sense disambiguation process corresponds to find the most “important” node in the graph.

Recently, Basile et al. propose a new unsupervised WSD algorithm which extends the Lesk’s WSD method (Basile et al., 2014). They introduce a distributional semantic space for WSD, then the word similarity in the semantic space is regarded as gloss-context overlap. Although Basile’s method is promising, it only relies on words in a context of a target word. However, it is well-known that a collocation is another useful feature for WSD. Collocation is a series of words or terms that frequently co-occur more than expected by chance. Words in a collocation usually have a special and fixed meaning. For example, the collocation “hot spring” indicates that the sense of

Han, S. and Shirai, K.
Unsupervised Word Sense Disambiguation based on Word Embedding and Collocation.
DOI: 10.5220/0010380112181225
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 1218-1225
ISBN: 978-989-758-484-8

“spring” is FOUNTAIN, not SEASON.

This paper proposes a novel unsupervised WSD method that extends Basile’s method. While the Basile’s method only considers words in a context for WSD, our method also takes collocations into account to determine a sense of a given word. In addition to the ordinary collocation (adjacent words that often appear together), we also define a dependency collocation, which is a syntactic dependency relation between a target word and another word in a sentence.

We also propose to change the way how to make a context vector in the semantic space. In the original research, context embedding is computed by average of word embedding of all words in a context. However, not all words are related to a sense of a target word. Our method only considers words that are highly related to the sense when the context embedding is built.

The rest of the paper is organized as follows. Section 2 provides a brief introduction about related work. Section 3 describes the details of our proposed method. Section 4 reports several experiments to evaluate our method. Finally, Section 5 concludes the paper.

2 RELATED WORK

There are three commonly used features in WSD. The first one is words in the surroundings of the target word. Part-of-speech (POS) tags of the neighboring words are also widely used features. Local collocations represent another standard feature that captures the ordered sequence of words which tend to appear around the target word (Bazell, 1959).

Many unsupervised WSD methods are based on calculation of similarity between word sense and its context using some features. One of the most traditional methods for unsupervised WSD is Lesk algorithm (Lesk, 1986). It is based on the assumption that words in a given section of text will tend to share a same topic. As already explained, it computes the similarity between the sense definition of an ambiguous word and the terms appearing in its neighborhood. There are many measures to determine the similarity between a sense and a context. Torres and Gelbukh present a comparison of several similarity measures applied to WSD by the Lesk algorithm (Torres and Gelbukh, 2009). Since gloss sentences tend to be short, several methods use external resources to get additional information of the sense. Bhingardive et al. try to use broad information of lexical database related to the sense, such as hypernyms, hyponyms, synonyms, and even example sentences in the dictionary to construct vector representation of the sense in order to identify the most frequent sense (Bhingardive et al., 2015).

The most important paper related to this study is (Basile et al., 2014). It utilizes semantic space, which is geometrical space of words where vectors express concepts of words. The proximity in the space can measure semantic relatedness between words. Since the gloss (definition) and the context are composed by several terms, the vector of each set of terms is built by adding the vector of every single words in the set. Pre-trained word embedding is used to construct the gloss and context vectors. The cosine similarity between gloss and context is used to choose the appropriate sense of the target word.

As already discussed in Section 1, this paper extends the Basile’s method in two directions. One is to incorporate a mechanism to determine a sense using a collocation. Rules to determine a sense, which are based on collocations, are automatically acquired from a raw corpus, then these rules are integrated to the Basile’s WSD model. The other is to propose a better way to construct the context vector, since the performance of WSD heavily relies on the quality of it.

3 PROPOSED METHOD

Figure 1 shows an overview of the proposed system. It accepts a sentence including a target word as an input and proposes a sense for it as an output. Our system consists of two modules: one is a rule based WSD system, the other is a WSD system based on Highly Related Word Embedding (hereafter, the HRWE method in short). The first module uses the database of collocation WSD rules, which determine the sense by a collocation (word sequence). Briefly, these rules determine the sense by a collocation as *collocation* → *sense*. If a rule is hit for a collocation in a given sentence, the sense is chosen by the rule, otherwise the next module is applied. The second module is similar to (Basile et al., 2014). It measures the similarity between gloss sentences in a dictionary and a context of a target word in a given sentence, then chooses the sense whose gloss is the most similar to the context of the target word. Since the rule-based module is designed to achieve high precision in compensation for low recall, it is applied first.

In the following subsections, the HRWE method will be introduced first, since it is also used to construct the sets of the collocation WSD rules. Then, the rule based WSD system is described, especially how to acquire WSD rules automatically.

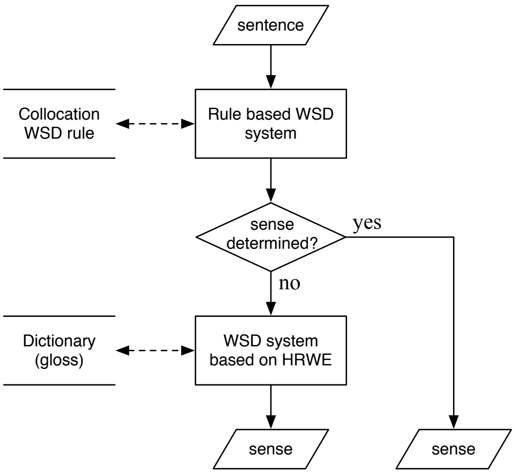

Figure 1: Overview of proposed WSD system.

3.1 Highly Related Word Embedding Method

In (Basile et al., 2014), three steps are required to determine a sense of a target word: (1) to construct a context vector $\vec{c}$, (2) to construct sense vectors $\vec{s_i}$, and (3) to calculate the cosine similarity of two vectors to choose the sense. The context vector is obtained by averaging the vectors of all context words in a context as Equation (1)

$$\vec{c} = \frac{1}{|W|} \sum_{w_k \in W} \vec{w_k} \tag{1}$$

, where W stands for a set of words in the context. Pretrained word embedding are used as word vectors $\vec{w_k}$. Similarly, the sense vector is constructed by averaging the word vectors in a gloss sentence. Finally, as show in Equation (2), the sense whose vector is the most similar to the context vector is chosen.

$$s = \underset{s_i}{\arg\max}\ cos(\vec{c}, \vec{s_i}) \tag{2}$$

An important parameter is the context window size, *CWS*. When constructing the context vector, the most nearest *CWS* words appearing before and after the target word are taken into account. Note that function words are ignored. That is, a content word is used to make a context vector if it is the *CWS*-th closest content word, even when the distance between it and a target word is greater than *CWS*. See also an example in the end of this subsection.

The gloss sentences in WordNet are used in the Basile's study, but they are rather concise. To enrich sense vectors, Basile expanded the gloss using an API provided by BabelNet, which could extract all the meanings related to a particular sense. In this study, we expand the sense information with the gloss of the hypernyms, hyponyms and synonyms, and empirically evaluate its effectiveness in the experiment.

In our HRWE method, not all words but only words highly related to senses are used to construct the context vector. For each sense s_i, a different context vector, denoted as $\vec{c^{(i)}}$, is made from contextual words relevant to s_i. First, for each word w_k in a context, the relevance score in terms of the i-th sense is defined as Equation (3), i.e. the maximum similarity between a word and sense vector.

$$RelevantScore(w_k^{(i)}) = max\ cos(\vec{w_k}, \vec{s_i}) \tag{3}$$

We assume that the word with high *RelevantScore* is strongly related to the particular sense, thus it is effective feature for WSD. The relevant word set, $WR^{(i)}$, is made by selecting the top T_r words with the highest *RelevantScore* for each sense s_i. Then the new sense-dependent context vector is made by averaging word vectors of words in $WR^{(i)}$ as in Equation (4).

$$\vec{c^{(i)}} = \frac{1}{|WR^{(i)}|} \sum_{w_k \in WR^{(i)}} \vec{w_k} \tag{4}$$

Finally, the sense with the highest $cos(\vec{c}^{(i)}, \vec{s_i})$ is chosen.

Figure 2 shows an example to obtain a relevant word set $WR^{(i)}$. In this example, the target word is "argument" that have four senses, and *CWS* and T_r are set to 5 and 3, respectively. The bottom table shows the cosine similarity between the word vector of each word in a context and the sense vector of each sense. The values in bold indicate the three highest *RelevantScore* for each sense, and these words are chosen as the relevant word set, shown in the bottom of Figure 2.

3.2 Collocation based WSD

Unlike the HRWE method, this method determines the sense by only looking at a collocation, i.e. idiomatic phrase including a target word.

3.2.1 Collocation WSD Rule

Collocation WSD rule is defined in the following form:

$$collocation \rightarrow \text{sense} = s_i \tag{5}$$

It means: when *collcocation* appears in an input sentence, s_i is chosen as the sense of the target word.

Two types of the collocation is considered in this study. The first one is a word collocation that is a sequence of words including the target word. Five types of word collocation are defined as in Figure 3.

Sentence:

> While using those methods, values passed to those variables are called **arguments**.

Sense of *argument*:

> s_1 a fact or assertion offered as evidence that something is true.
> s_2 a reference or value that is passed to a function, procedure, subroutine, command, or program.
> s_3 a summary of the subject or plot of a literary work or play or movie.
> s_4 a contentious speech act; a dispute where there is strong disagreement.

Relevance score:

	method	value	pass	variable	call
s_1	**0.7**	0.1	**0.5**	**0.6**	0.3
s_2	0.5	**0.5**	**0.8**	**0.9**	0.4
s_3	0.3	**0.5**	0.2	**0.4**	**0.4**
s_4	**0.5**	**0.8**	**0.6**	0.3	0.4

$WR^{(1)}$ = { method, pass, variable }
$WR^{(2)}$ = { value, pass, variable }
$WR^{(3)}$ = { value, variable, call }
$WR^{(4)}$ = { method, value, pass }

Figure 2: Example of relevant word set.

$w_{i-2}\ w_{i-1}\ \mathbf{w}$	→	sense=s_i
$w_{i-1}\ \mathbf{w}$	→	sense=s_i
$w_{i-1}\ \mathbf{w}\ w_{i+1}$	→	sense=s_i
$\mathbf{w}\ w_{i+1}$	→	sense=s_i
$\mathbf{w}\ w_{i+1}\ w_{i+2}$	→	sense=s_i

Figure 3: Template of word collocation WSD rule.

$\mathbf{w}$ - *rel* - w_c	→	sense=s_i
w_c - *rel* - $\mathbf{w}$	→	sense=s_i

Figure 4: Template of dependency collocation WSD rule.

w stands for the target word, while w_x stands for a word just before or after the target word. $x \in \{i-2, i-1, i+1, i+2\}$ denotes a relative position.

The second collocation is a dependency collocation, which is defined as a pair of words under a certain syntactic relation such as "subject", "object", "modifier" and so on. Figure 4 shows the precise definition of the rule. **w** is the target word, while w_c is a word in a context that is under the dependency relation *rel* with **w**.

3.2.2 Construction of Collocation WSD Rule

Collocation WSD rules are automatically acquired from a raw corpus. Figure 5 shows overall procedures.

First, for each sentence in an unlabeled corpus, the HRWE method determines a sense of a target word. If the chosen sense is reliable enough, the sentence is used to obtain candidates of collocation WSD rules. The reliability of the disambiguated sense s_i is defined as the cosine similarity between the context vector $\vec{c^{(i)}}$ and sense vector $\vec{s_i}$. If it is less than the threshold T_{wsd}, the sentence is just ignored.

Next, candidates of collocation WSD rules are generated by applying rule templates shown in Figure 3 and 4. For example, from the sentence "they were always getting into arguments about politics", where the HRWE determines the sense of "argument" as s_1, the following rules are obtained:

getting into **argument**	→	sense=s_1
into **argument**	→	sense=s_1
into **argument** about	→	sense=s_1
argument about	→	sense=s_1
argument about politics	→	sense=s_1
getting - *obj* - **arugment**	→	sense=s_1
argument - *case* - into	→	sense=s_1
argument - *nmod* - politics	→	sense=s_1

The first five rules are word collocation WSD rules, while the rest are dependency collocation WSD rules, which are derived from the dependency tree shown in Figure 6. Stanford Parser[1] is used to analyze dependency relations in this study.

3.2.3 Filtering Collocation WSD Rule

After obtaining the candidates of the collocation WSD rules, inaccurate ones are filtered out. We apply the following three filtering procedures.

- **Stop Word.**
  The collocation consisting of only the target word and function words may not strongly associate with any senses. Therefore, rules including such collocations are discarded. Here are examples of the removed rules.

play a	→	sense=s_2
the **argument**	→	sense=s_1

  We have prepared 29 function words for this filtering.

- **Infrequent Collocation.**
  If the frequency of a collocation in a corpus is

[1] https://nlp.stanford.edu/software/lex-parser.shtml

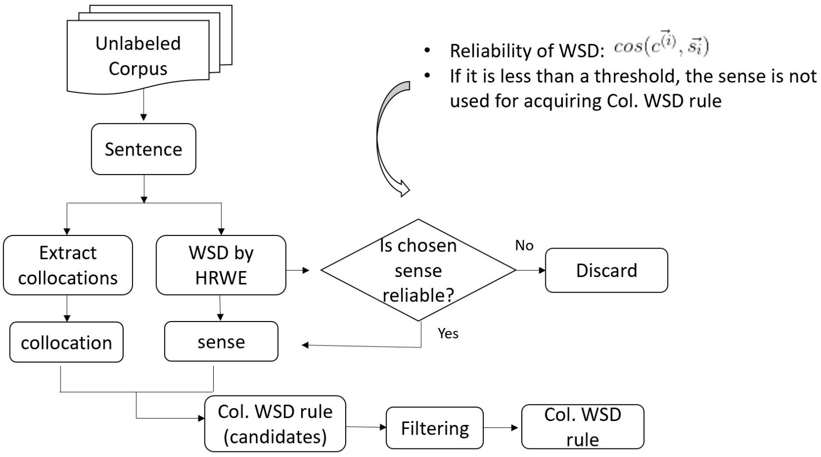
Unlabeled Corpus
Sentence
Extract collocations
WSD by HRWE
collocation
sense
Is chosen sense reliable?
No
Discard
Yes
Col. WSD rule (candidates)
Filtering
Col. WSD rule
• Reliability of WSD: $cos(\vec{c^{(i)}}, \vec{s_i})$
• If it is less than a threshold, the sense is not used for acquiring Col. WSD rule

Figure 5: Flowchart of acquisition of collocation WSD rule.

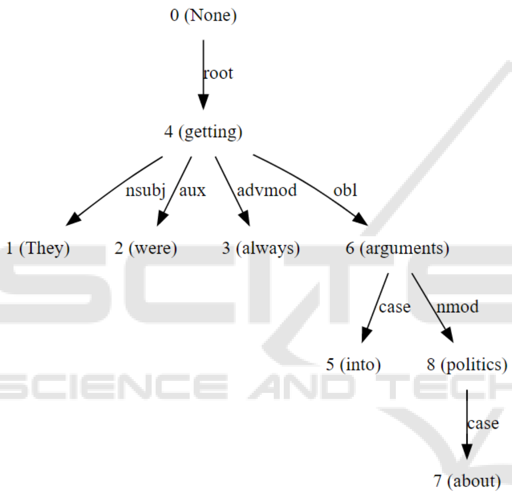

Figure 6: Dependency tree of example sentence.

small, a rule might be unreliable. Therefore, rules are removed if the number of the collocation is less than the threshold T_{fre}.

- **Reliability.**
  Obviously, not all rules are effective to choose a correct sense. Several rules are even inconsistent when the same collocation determines different senses such as "$col \rightarrow sense = s_1$" and "$col \rightarrow sense = s_2$". Therefore, the reliability score of the rule is defined as

  $$score(col \rightarrow s_i) = \frac{f(col, s_i)}{\sum_i f(col, s_i)} \quad (6)$$

  , where $f(col, s_i)$ is the frequency of sentences including the collocation col and the sense s_i. Basically, this score means the precision of WSD when the sense is determined by the rule. If $score(col \rightarrow s_i)$ is less than T_{sco}, the rules are removed.

After applying these three filtering modules, the final set of collocation WSD rules is obtained.

4 EXPERIMENTS

4.1 Experimental Setting

The dataset of Senseval-3 English lexical sample task is used to evaluate the performance of WSD of the proposed systems. It consists of instances (sentences or paragraphs including the target word) annotated with gold senses for several target verbs, nouns, and adjectives. The statistics of the dataset is shown in Table 1.

Table 1: Dataset of Senseval-3 English lexical sample task.

POS	# of words	ave.# of instances
Verb	32	53.1
Noun	20	78.5
Adjective	5	28.2
Total	57	59.8

In the Senseval-3 data, the senses are defined by WordNet (Miller, 1995). As for the sense inventory, glosses in WordNet 1.7.1 are used for nouns and adjectives, while definition sentences in Wordsmyth[2] are used for verbs. Since the senses in WordNet are fine-grained and differences of some senses are too subtle, we define a set of coarse-grained senses by manually merging similar senses. The average num-

[2] http://www.wordsmyth.net/

bers of the senses per word in the original WordNet and our coarse sense set are shown in Table 2.

Table 2: Average number of senses.

POS	WordNet	Our coarse sense
Verb	6.31	4.07
Noun	5.8	3.58
Adjective	10.2	2.75

A large unlabeled corpus is required to mine the collocation WSD rules. In this experiment, 200,000 English sentences from the Leipzig corpus[3] are used.

To construct the context and sense vectors, three pre-trained word embedding are used: word embedding pre-trained by the Skip-gram model from Google News corpus[4], Glove[5](Pennington et al., 2014), and BERT[6](Devlin et al., 2018). Since word embedding in BERT is dynamic, i.e. sentence-dependent, we expect that it is good to produce abstract vector representation of a context and sense. A context sentence or a gloss sentence is given to the pre-trained language model of BERT, then the average vector of every token in the last layer of the BERT forms the context or sense vector.

Preliminarily, these three word embeddings are compared. Table 3 shows the average precision for disambiguation of verbs in the test data using the Basile's method with different word embedding models. Here the context window size *CWS* is set to 10. It is found that the performance of the BERT is rather poor. It indicates that pre-trained BERT model may not be appropriate for WSD. Since the result of this experiment indicates that the Skip-gram model is the best, only the Skip-gram model is used in our experiments.

Table 3: Comparison of word embedding.

Type	Precision
Skip-gram	0.544
Glove	0.529
BERT	0.424

4.2 Results

First, a preliminary experiment is carried out to confirm the effectiveness of the word expansion. As explained in Subsection 3.1, in Basile's method, not only gloss sentences but also glosses of its related words (hypernym, hyponym, and synonym) are used to make a sense vector. Table 4 compares the precision of the original method with and without the gloss expansion. Although (Basile et al., 2014) reported that gloss expansion was effective, it is not true in our experiment using Senseval-3 dataset. We are still uncertain why it happens. Careful investigation on impact of the gloss expansion in unsupervised WSD is worth being carried out in future. Anyway, the gloss expansion is not performed in the rest of experiments.

Table 4: Evaluation of gloss expansion in Basile's method.

Model	Precision		
	Verb	Noun	Adj
w/o expansion	0.542	0.505	0.560
with expansion	0.517	0.457	0.447

Table 5 reveals the precision of WSD for verbs, nouns, adjectives, and all POSs. The third row is the baseline that is equivalent to (Basile et al., 2014), while the fourth row is the WSD system using our proposed HRWE method only. The HRWE outperforms the baseline for nouns and verbs, but not for adjectives. However, the precision is improved by 3.2 point for all POSs. It indicates that our idea to select contextual words strongly associated with senses for the context embedding is effective.

The fifth row shows the precision by using collocation WSD rules only. The applicability, proportion of the number of disambiguated instances by the rules to the total number of instances in the test data, is shown in the sixth row. The applicability of the rules is low, i.e. senses in many sentences cannot be determined. However, the rules tend to achieve the higher precision than the previous two systems, especially for nouns and adjectives. It is confirmed that we can obtain the disambiguation rules whose recall is low but precision is high as we aimed. Note that the applicability of all other WSD systems in Table 5 is 1, that is, senses of all target instances are determined.

The 7-th to 10-th rows show the performance of the systems integrating the baseline or HRWE with the word or dependency collocation WSD rules. The use of two different WSD systems can increase the precision. Therefore, it is confirmed that both words in a context (considered in the baseline or HRWE) and collocations (considered in the rules) can contribute to choose the appropriate sense. Comparing 7-th and 8-th or 9-th and 10-th rows, the contribution of two types of collocation WSD rules (word vs. dependency) are almost equivalent.

Finally, the last row shows the precision of the WSD system with the HRWE and both word and dependency collocation WSD rules. It achieves the best

[3] https://wortschatz.uni-leipzig.de/en/download/english

[4] https://code.google.com/archive/p/word2vec/

[5] https://nlp.stanford.edu/projects/glove/

[6] https://github.com/google-research/bert

Table 5: Comparison of WSD methods.

Method	Precision			
	Verb	Noun	Adj	All
Baseline	0.542	0.506	**0.560**	0.525
HRWE only	0.583	0.534	0.511	0.557
collocation WSD rule only	0.573	0.631	0.625	0.591
(applicability)	(36.4%)	(17.8%)	(11.3%)	(26.8%)
Baseline + word collocation	0.553	0.516	0.553	0.536
Baseline + dependency collocation	0.547	0.510	0.546	0.530
HRWE + word collocation	0.588	0.545	0.511	0.565
HRWE + dependency collocation	0.589	0.540	0.525	0.564
HRWE + word & dependency collocation	**0.594**	**0.552**	0.525	**0.572**

performance for nouns,verbs and all POSs as indicated in bold. Its precision is 0.572, which is 4.7 point better than the baseline.

It is found that our HRWE and collocation WSD rules poorly perform for the disambiguation of adjectives. However, the number of target adjectives in the test data is rather small, i.e. only 5. We will evaluate our proposed method for more adjectives and investigate how our system can improve sense disambiguation of adjectives in future.

4.3 Evaluation of Collocation WSD Rules

The details of the acquisition of the collocation WSD rules are reported in this subsection.

Recall that there are three thresholds for rule acquisition: T_{wsd} (the reliability of WSD), T_{fre} (the frequency of the collocation), and T_{sco} (the score of the rule). These parameters are empirically determined for individual POSs as in Table 6. They are optimized so that the precision on the test data becomes the highest, although they should be normally optimized on a development data.

Table 6: Parameters for acquisition of collocation WSD rule.

	T_{wsd}	T_{fre}	T_{sco}
Verb	0.75	4	0.7
Noun	0.7	5	0.7
Adjective	0.7	4	0.7

Table 7 shows the number of candidates of rules and rules after the filtering. Around five hundred word collocation WSD rules and nine hundred dependency rules are finally obtained. It is found that most of the candidates are inaccurate and discarded by our filtering methods.

Table 7: Number of rules mined from raw corpus.

	candidates	after filtering
word col. rule	132,300	528
depen. col. rule	120,170	379

We could find many correct rules. Figure 7 shows the examples of acquired rules. For example, the last rule indicates that when “argument” is an object of the verb “refute”, its meaning is s_1 (assertion).[7]

earth’s **atmosphere** → sense=s_1(air)
bank robber → sense=s_3(financial institute)
running **arguments** → sense=s_2(parameter)
talk - advmod - speechify → sense=s_1(speech)
refute - obj - **argument** → sense=s_1(assertion)

Figure 7: Example of acquired col. WSD rule.

4.4 Discussion about Context Window Size

Next, influence of the context window size *CWS* on the WSD performance is investigated. *CWS* is changed to 5, 8, and 10 in the baseline and HRWE method, then the WSD precision of these models are compared. Note that collocation WSD rules are not used in this experiment. Figure 8 (a) and (b) show the results for verb and noun, respectively.[8] The precision of our HRWE method is improved when *CWS* is increased, while that of the baseline is declined for both verbs and nouns. In the baseline method, when more context words are added to the context vector, words irrelevant to the correct sense are also added in great numbers. It results in spoiling the quality of the context vector. On the other hand, in the HRWE, not all but fixed number of highly related words are used to make the context vector. When the context window size is increased, words that are far from a target word but effective for WSD can be taken into account.

[7] See also the sense definition in Figure 2.

[8] A result of adjective is omitted since the number of target words in the test data is small.

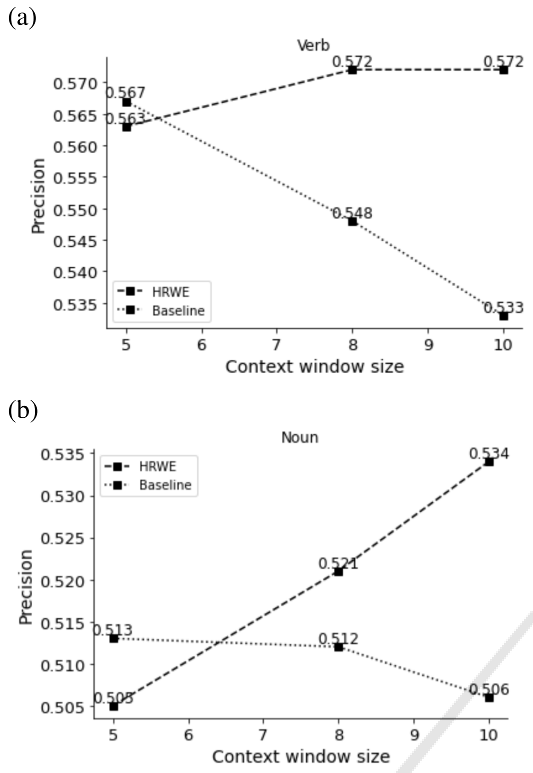

Figure 8: Precision of models with different context window sizes.

5 CONCLUSION

This paper proposed the novel unsupervised WSD system consisting of two methods. One was the method to determine the sense by looking up the collocation that strongly indicated the sense of the target word. Two types of the collocation WSD rules were acquired from a raw corpus. The other was the HRWE method that measured the similarity between the context and the gloss sentences, where noisy words were ignored in the construction of the context vector. The experimental results on Senseval-3 English lexical sample task dataset showed that our proposed method outperformed the previous work (Basile et al., 2014) by 4.7 points.

The contribution of the paper was summarized as follows. First, the collocation was newly integrated as another useful feature into the existing word embedding based method, which only considered words in the context. Ensemble of collocation based and word embedding based methods was effective to improve the precision of WSD. Another contribution was to refine how to make the context vector, where only highly related words were chosen to get better representation of the context.

In future, more sophisticated methods to make the context and sense vectors will be explored. For example, it is worth investigating a method to use Sentence BERT (Reimers and Gurevych, 2019) to obtain the vector representation of the sentences. Another important line is to combine other unsupervised methods such as graph based ones with our HRWE method and collocation WSD rules.

REFERENCES

Basile, P., Caputo, A., and Semeraro, G. (2014). An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600.

Bazell, C. (1959). Studies in linguistic analysis. special volume of the philological society, vii, 205 pp., 5 plates. oxford: Basil blackwell, 1957. 70s. *Bulletin of the School of Oriental and African Studies*, 22(1):182–184.

Bhingardive, S., Singh, D., Rudramurthy, V., Redkar, H., and Bhattacharyya, P. (2015). Unsupervised most frequent sense detection using word embeddings. In *Proceedings of the 2015 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies*, pages 1238–1243.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Huang, L., Sun, C., Qiu, X., and Huang, X. (2019). GlossBERT: BERT for word sense disambiguation with gloss knowledge. *arXiv preprint arXiv:1908.07245*.

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 5th annual international conference on Systems documentation*, pages 24–26.

Miller, G. A. (1995). WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Navigli, R. and Lapata, M. (2009). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):678–692.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv preprint arXiv:1908.10084*.

Torres, S. and Gelbukh, A. (2009). Comparing similarity measures for original WSD Lesk algorithm. *Research in Computing Science*, 43:155–166.