Semi-supervised Word Sense Disambiguation Using Example Similarity Graph

Rie Yatabe, Minoru Sasaki
Ibaraki University
{19nm732r, minoru.sasaki.01}@vc.ibaraki.ac.jp

Abstract

Word Sense Disambiguation (WSD) is a well-known problem in the natural language processing. In recent years, there has been increasing interest in applying neural networks and machine learning techniques to solve WSD problems. However, these previous supervised approaches often suffer from the lack of manually sense-tagged examples. In this paper, to solve these problems, we propose a semi-supervised WSD method using graph embeddings based learning method in order to make effective use of labeled and unlabeled examples. The results of the experiments show that the proposed method performs better than the previous semi-supervised WSD method. Moreover, the graph structure between examples is effective for WSD and it is effective to utilize a graph structure obtained by fine-tuning BERT in the proposed method.

1 Introduction

In human languages, many words have multiple meanings, depending on the context in which they are used. Identifying the sense of a polysemous word within a given context is a fundamental problem in natural language processing. For example, the English word "bank" has different meanings as "a commercial bank" or "a land along the edge of a river," etc. Word sense disambiguation (WSD) is the task of deciding the appropriate meaning of a target ambiguous word in its context (Navigli, 2009).

Among various approaches to the WSD task used over the past two decades, a supervised learning approach has been the most successful. However, it is quite expensive in both time and cost to annotate a large amount of reliable training data because supervised WSD requires a large amount of manually labeled training examples to achieve good performance. Unsupervised learning approach does not need labeled examples and uses large amount of unlabeled examples to find word clusters which discriminates the senses of the words in different clusters. Unsupervised learning algorithms are typically less accurate than supervised algorithms because examples may not be assigned the correct sense. For these reasons, we focus on a semi-supervised learning method that uses both sense-labeled and unlabeled examples in different proportions.

In this paper, we propose a semi-supervised WSD method using graph embeddings based learning method. In this method, we extract features from the context around the target word for each labelled and unlabeled example and construct a graph structure between labelled and unlabeled examples to obtain classifiers for every polysemous word. Then, we construct classifiers for each polysemous word using Planetoid (Yang et al., 2016), which is a multi-task framework for graph-based semi-supervised learning. By using the proposed method, it is possible to incorporate information obtained from unlabeled examples without assigning a sense label to unlabeled examples. Moreover, by learning graph embeddings, it is possible to distinguish between two similar examples with different sense labels to construct a better classifier for WSD. To evaluate the efficiency of the proposed WSD method, we design some experiments using the Semeval-2010 Japanese WSD task data set and the Senseval-2 English lexical sample task data set.

The three contributions of this work can be summarized as follows:

1. We employ a graph embeddings based learning method for a semi-supervised WSD system to incorporate information obtained from unlabeled examples.

Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs), pages 51–59
Barcelona, Spain (Online), December 13, 2020

2. We show that the graph structure between examples, which is constructed by the relation between the training data and the unlabeled data, is effective for WSD. The graph constructed by determining if the example sentences are the same usage with BERT is more effective than the graph constructed by two major similarities, cosine similarity and Jaccard coefficient.

3. We show that the proposed method performs better than the previous semi-supervised WSD method on the Semeval-2010 Japanese WSD task and the SENSEVAL-2 English lexical sample task.

The rest of this paper is organized as follows. Section 2 is devoted to presenting related works in the literature. Section 3 describes the proposed semi-supervised WSD method. In Section 4, we describe an outline of our experiments. In Section 5, we present experimental results. Finally, we conclude the paper in Section 6.

2 Related Works

This section is a literature review of previous work on semi-supervised WSD and various related methods using a neural network.

In recent years, there has been increasing interest in applying neural networks and machine learning techniques to solve WSD problems. (Kågebäck and Salomonsson, 2016) employed a Bidirectional Long Short-Term Memory (Bi-LSTM) to encode information of both preceding and succeeding words within the context of a target word. (Yuan et al., 2016) used an LSTM language model to obtain a context representation from a context layer for the whole sentence containing a target word. The context representations were compared to the possible sense embeddings for the target word. Then, the word sense whose embedding had maximal cosine similarity was assigned to classify a target word. (Raganato et al., 2017) considered WSD as a neural sequence labelling task and constructed a sequence learning model for all-words WSD. These approaches are characterized by their high performance, simplicity, and ability to extract a lot of information from raw text.

In recent years, semi-supervised learning has been used in WSD tasks. Semi-supervised learning is a technique that makes use of a small number of sense-labelled examples with a large amount of unlabeled examples. (Yarowsky, 1995) proposed a bootstrapping model that only has a small set of sense-labelled examples that gradually assigns appropriate senses to unlabeled examples. (Taghipour and Ng, 2015) and (Yuan et al., 2016) proposed a semi-supervised WSD method to use word embeddings of surrounding words of the target word and showed that the performance of WSD could be increased by taking advantage of word embeddings. (Fujita et al. 2011) proposed a semi-supervised WSD method that automatically obtains reliable sense labelled examples using example sentences from the Iwanami Japanese dictionary to expand the labelled training data. Then, this method employs a maximum entropy model to construct a WSD classifier for each target word using common morphological features (surrounding words and POS tags) and topic features. Finally, the classifier for each target word predicts the sense of the test examples. They showed that this method is effective for the SemEval-2010 Japanese WSD task. (Sousa et al., 2020) proposed a graph-based semi-supervised WSD method using word embeddings and distance measures. Although this method fixes the graph structure and trains only the classification model, our method trains the classification model and the graph jointly.

Some research in the field of WSD has taken advantage of graph-based approaches. (Niu et al., 2005) proposed a label propagation-based semi-supervised learning algorithm for WSD, which combines labelled and unlabelled examples in the learning process. (Yuan et al., 2016) also introduced a label propagation (LP) for semi-supervised classification and LSTM language model. An LP graph consists of vertices of examples and edges that represent semantic similarity. In this graph, label propagation algorithms can be efficiently used to apply sense labels to examples based on the annotation of their neighbours. In this paper, we use a semi-supervised learning method that incorporates knowledge from unlabeled examples by using graph embeddings based learning method.

Besides the semi-supervised learning, several recent methods have shown that combining supervised neural WSD systems with external knowledge base information improves the WSD performance. GlossBERT (Huang et al., 2019) and BEM (Blevins and Zettlemoyer, 2020) combine supervised learning with knowledge from gloss information to make better performance. EWISE (Kumar et al., 2019) incorporates both gloss embeddings and Knowledge Graph Embeddings. EWISER (Bevilacqua and Navigli, 2020) incorporates both synset embeddings and WordNet relations instead of leveraging sense glosses.

These methods represent the relationships between words as a graph structure, which means that words are represented as nodes and the relationships between words are represented as edges. In this study, we use a graph structure where the nodes are examples and the edges are semantic similarity between examples, which is different from the structure of these previous methods.

3 WSD Method Using Graph-based Semi-supervised Learning

In this section, we describe the details of the proposed semi-supervised WSD method using graph embeddings based learning method.

3.1 Overview of the Proposed Method

Our WSD method is used to select the appropriate sense for a target polysemous word in context. WSD can be viewed as a classification task in which each target word should be classified into one of the predefined existing senses. Word senses were annotated in a corpus in accordance with "Iwanami's Japanese Dictionary (The Iwanami Kokugo Jiten)" (Nishio et al. 1994). It has three levels for sense Ids, and the middle-level sense is used in this task.

The proposed semi-supervised WSD method requires a corpus of manually labelled training data to construct classifiers for every polysemous word and a graph between labelled and unlabeled examples. For each labelled and unlabeled example, features are extracted from a context around the target word, and the feature vector is constructed. Given a graph structure and feature vectors obtained from training data, we learn an embedding space, for example, set to jointly predict the sense label and neighborhood similarity in the graph using Planetoid (Yang et al., 2016). When the WSD classifier is obtained, we input test examples only to the WSD model and predict one sense for each test example.

3.2 Features

3.2.1 Lexical and Syntactic Features

To implement the proposed WSD system, we extracted features from training data and test data of a target word, unlabeled examples from the Balanced Corpus of Contemporary Written Japanese (BCCWJ) corpus (Maekawa, 2014), and example sentences extracted from Iwanami Japanese Dictionary. To segment a sentence into words, we use popular Japanese morphological analyzer MeCab with the morphological dictionary UniDic.

In this paper, we extract the target word and the two words on either side of the target word and then use the following twenty features (BF) for the target word w_i, which is the *i*-th word in the example sentence.

e1: the word w_{i-2}
e2: part-of-speech of the word w_{i-2}
e3: subcategory of the e2
e4: the word w_{i-1}
e5: part-of-speech of the word w_{i-1}
e6: subcategory of the e5
e7: the word w_i
e8: part-of-speech of the word w_i
e9: subcategory of the e8
e10: the word w_{i+1}
e11: part-of-speech of the word w_{i+1}
e12: subcategory of the e11
e13: the word w_{i+2}
e14: part-of-speech of the word w_{i+2}
e15: subcategory of the e14
e16: word that contains dependency relation with the w_i
e17: thesaurus ID number of the word w_{i-2}
e18: thesaurus ID number of the word w_{i-1}

e19: thesaurus ID number of the word w_{i+1}
e20: thesaurus ID number of the word w_{i+2}

To obtain the thesaurus ID number of each word, we use five-digit semantic classes obtained from a Japanese thesaurus “Bunrui Goi Hyo” (NIJL, 2004). When a word has multiple thesaurus IDs, e17, e18, e19, and e20 contain multiple thesaurus IDs for each context word. As additional local collocation (LC) features, we use bi-gram, tri-gram, and skip-bigram patterns in the three words on either side of the target word like IMS (Zhong and Ng., 2010). Skip-bigram is any pair of words in an example order with arbitrary gaps. Then, we can represent a context of word w_i as a vector of these features, where the value of each feature indicates the number of times the feature occurs.

3.2.2 Contextual Word Embeddings Using ELMo

Next, considering the context of the input sentence, we apply contextual word representations derived from pre-trained bi-directional language models (biLMs) as features. We use a pre-trained ELMo model to obtain a word embeddings for input sentences containing the target word. This pre-trained ELMo model is a two-layer bidirectional LSTM structure to learn the contextual information from text. We use only the output of the last layer of the pre-trained model. The target word and the two words on either side of the target word are extracted and the word embeddings of these five words is obtained. Then, we generate a 5120 dimensional vector that concatenates the word embeddings of these five words. Word embeddings are generated for each word in an input sentence depending on the context so that different embeddings tend to be generated for each word sense. We consider that the word embeddings of these five words allows us to capture the context of the target word.

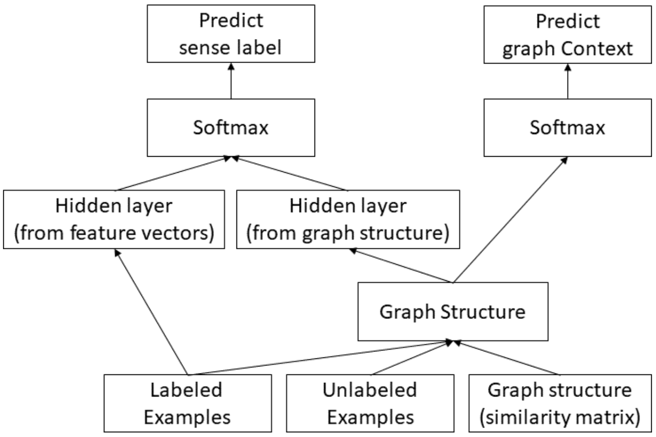
Predict sense label
Predict graph Context
Softmax
Softmax
Hidden layer (from feature vectors)
Hidden layer (from graph structure)
Graph Structure
Labeled Examples
Unlabeled Examples
Graph structure (similarity matrix)

Figure 1. WSD model using graph embeddings based learning

3.3 Graph-based Semi-supervised Learning

We employ Planetoid, which is a graph-based semi-supervised learning framework, for the WSD model and predicts the sense of target word. The planetoid has two versions, transductive and inductive models. In this study, we use the inductive model which can predict labels for instances that are not observed in the graph. In the inductive model, as shown in Figure 1, we use a set of training examples, unlabeled examples and a graph structure representing the relationship between examples as input and learn a WSD classifier and graph context simultaneously. The classifier predicts the sense of the target word for unknown example.

The training examples and unlabeled examples are represented by feature vectors. The graph structure is constructed from the similarity between the obtained vectors. We learn a WSD model from the training data vector and the graph structure. Finally, we predict the appropriate sense label of the target word for the unknown examples using the optimized WSD model.

3.4 Input Graph Structure

3.4.1 Graph Using Jaccard Coefficient and Cosine Similarity

The input graph structure is constructed by the relation between the training data and the unlabeled data. In the graph structure, each node is an example and an edge is the similarity between nodes. The similarity between nodes is calculated by using the following calculation method between two vectors of examples. In the proposed method, nodes with the highest similarity and nodes that have a similarity not less than the threshold value 0.9 are connected by edge. Figure 2 shows how the nodes are connected.

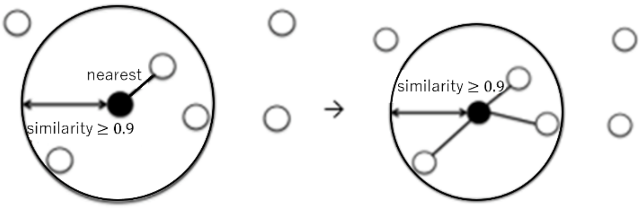

Figure 2. Edge connection between examples

The similarity calculation method between nodes uses Jaccard similarity J or cosine similarity. Jaccard similarity J is the ratio of the number of words in common between the two sets. Given a set of word vectors A and B, the similarity J is represented as follows:

$$J(A,B) = |A \cap B|/|A \cup B|, \qquad (0 \leq J(A,B) \leq 1)$$

Moreover, we use a mutual k-nearest neighbour graph to construct a graph structure. The mutual k-nearest neighbour graph is defined as a graph that connects edge between two nodes if each of the nodes belongs to the k-nearest neighbours of the other. In this method, the edges with the highest similarity between nodes are also added to the graph structure obtained by the mutual k-nearest neighbour graph. In our experiments, we use k=3 for the number of neighbours that have been provided by the user.

3.4.2 Graph Using Sentence Pair Classification by Fine-tuning BERT

To create a suitable graph structure, we use BERT to identify if two examples of the target word are used in the same sense. BERT is a ground-breaking natural language processing model for pre-training language representations. The pre-trained BERT model is built on a huge text data sets and is fine-tuned with just one additional output layer to create state-of-the-art models for various tasks. In our experiment, we use NWJC-BERT as a Japanese BERT pre-trained model published by the National Institute for Japanese Language and Linguistics (NINJAL) to compute BERT features. This model is trained on NINJAL Web Japanese Corpus (NWJC) pre-tokenized by MeCab with UniDic. Also, we use BERT-large-cased as an English pre-trained model and freeze the weights of parameters of BERT layers. We feed the output of these pre-training models into a three fully connected layer with ReLU activation function and softmax outputs to identify whether two examples are used in the same sense or not.

For each target word, we train a model to determine if usage is the same by entering example sentence pairs $s1$ and $s2$ in the training data. For all sentence pairs in the training data, pairs of seven words consisting of the target word and three words on either side of the target word $(w_{s1}^1, w_{s1}^2, \cdots, w_{s1}^7)$ and $(w_{s2}^1, w_{s2}^2, \cdots, w_{s2}^7)$ are extracted by performing morphological analysis. Then, we concatenate these sequences and feed the pair to BERT to get their contextualized representation:

$$\{'[\mathrm{CLS}]', w_{s1}^1, w_{s1}^2, \cdots, w_{s1}^7, '[\mathrm{SEP}]', w_{s2}^1, w_{s2}^2, \cdots, w_{s2}^7, '[\mathrm{SEP}]'\},$$

where a special token '[CLS]' denotes the beginning of the input sequence and a token '[SEP]' is used for separating examples or denoting the end of the sequence. We add an untrained layer with two hidden layers of 64 units on the end to the pre-trained BERT model and train the model using all the example pairs obtained from the training data for 20 epochs. This model is applied to every unlabeled sentence pair to obtain semantically equivalent sentence pairs. The obtained equivalent sentence pairs are used to generate a graph structure.

4 Experiments

To evaluate the efficiency of the proposed WSD method using graph embeddings based learning, we conducted some experiments to compare the results to the baseline system. In this section, we describe an outline of the experiments.

4.1 Data Set

For our experiments, we relied on the lexical-sample datasets of both SemEval-2010 Japanese WSD (Okumura, 2010) and Senseval-2 Lexical Sample (Kilgarriff, 2001).

The SemEval-2010 Japanese WSD task data set includes 50 target words comprising 22 nouns, 23 verbs, and 5 adjectives. In this data set, there are 50 training and 50 test instances for each target word (there is no development set). As unlabeled example data for the construction of a graph structure, we used the BCCWJ developed by the National Institute for Japanese Language and Linguistics. The BCCWJ corpus comprises 104.3 million words covering various genres.

The Senseval-2 English Lexical Sample data consists of 8,611 training and 4,328 test examples of 73 words. These examples were extracted from the British National Corpus. This task uses WordNet 1.7 sense inventory as gold standard, which has fine-grained sense distinctions. As unlabeled example data for the construction of a graph structure, we used the SemCor corpus. The SemCor corpus is a subset of the Brown corpus that contains 362 texts comprising about 234,000 words.

4.2 Settings

In our experiments, to construct a graph for all examples, two nodes that represent two examples are linked if they are nearest and if their similarity (based on the Jaccard coefficient) is not less than a specified threshold value of 0.9. The basic idea behind this is that two nodes tend to have a high similarity if the corresponding contexts of the target word are similar.

For learning the graph-based neural network, we use default parameter settings from the existing implementation[1]. For the initialized embeddings, optimization of the loss function of class label prediction is repeated for 10,000 iterations, and optimization of the loss function of graph context prediction is repeated for 1,000 iterations. Then, Joint training of classification and graph context prediction is repeated for 1,000 iterations. The obtained model is used to classify new examples of the target word into semantic classes.

5 Experimental Results

Table 1 shows the results of the experiments using lexical features with various graph structures on the SemEval-2010 Japanese WSD task. The reported WSD task results in precision are averaged over three runs. As shown in Table 1, the average precision of the proposed method is 78.11%. Statistical significance is obtained using the Student's t-test, †P<0.05 in comparison to the WSD model without the graph embeddings. The proposed method obtains higher precision than the WSD model without the graph embeddings, with statistical significance. Therefore, experimental results show that the graph structure between examples is effective for WSD. However, the model with randomly connected graph obtains a lower precision than the model without graph embeddings. This shows that it is not effective for WSD to use the graph structure simply in the model, but that it is more effective to use a graph structures that takes into account the similarity between examples.

Among the WSD models using the graph structure based on the similarity between examples, the model using sentence pair classification by fine-tuning BERT obtains the highest precision. Therefore, it is effective for WSD to utilize a graph structure obtained by fine-tuning BERT in the proposed method.

In the Table 2, we show the experimental results using word embeddings with various graph structures. Statistical significance is obtained using the Student's t-test, ††P<0.01 in comparison to the WSD model using ELMo without the graph embeddings. In these experiments, we use the WSD models using a graph structure obtained by fine-tuning BERT. By using contextual word embeddings using ELMo as features, the average precision of the proposed WSD model is 80.93%. As a result, the proposed method outperformed the existing method in the semi-supervised learning method on the SemEval-2010 Japanese

[1] https://github.com/kimiyoung/planetoid

WSD task data. The average precision of the proposed method was improved by 2.7% in comparison with the model without graph embeddings so that we show that the graph structure is effective for WSD even when word embeddings are used as features.

	Precision (%)
No Graph (MLP only)	76.92
Random Connection Graph	76.72
Cosine Similarity	77.24
Jaccard Coefficient	77.76†
Pair Classification with BERT (no fine-tuning)	77.40
Pair Classification with BERT (proposed)	**78.11**†

Table 1: Experimental results using lexical features with various graph structures

	Precision (%)
(Fujita et al., 2011) (baseline)	79.20
No Graph (word2vec, MLP only)	70.65
No Graph (ELMo, MLP only)	77.23
Pair Classification with BERT (word2vec)	79.29
Cosine Similarity	80.64††
Jaccard Coefficient	80.90††
Pair Classification with BERT (ELMo) (proposed)	**80.93**††

Table 2: Experimental results using word embeddings with various graph structures

To demonstrate the effectiveness of the proposed method, we compared it with a WSD model using word2vec embeddings as features. We obtain word embeddings of word2vec using nwjc2vec (Asahara, 2018) and perform experiments using the WSD model with 1000 dimensional vectors concatenated with 200-dimensional word embeddings for 5 words. Experimental results showed that the average precision of the WSD model using word2vec is 79.6%. These results show that the proposed WSD model using contextual word embeddings with ELMo is more effective than the model using word2vec. In the WSD model with word2vec, the average precision of the model with graph is 7.36% higher than that without graph. This indicates that the graph structure is also effective for the WSD method using word2vec.

Table 3 shows the results of the experiments of the proposed method on the SENSEVAL-2 English lexical sample task. As shown in Table 3, the average precision of the proposed method is 73.09%. The proposed method achieves higher precision than the existing WSD model such as (Taghipour and Ng, 2015), (Kågebäck and Salomonsson, 2016). However, the average precision of the proposed method is slightly better than the model without a graph. One of the reasons for this is that the number of SemCor documents used as unlabeled examples is quite small. Therefore, it is necessary to extract enough examples of the target words by using a large corpus such as OMSTI in addition to SemCor.

	Precision (%)
(Taghipour and Ng, 2015) (baseline)	66.2
(Kågebäck and Salomonsson, 2016) (baseline)	66.90
No Graph (ELMo, MLP only)	73.01
Pair Classification with BERT (ELMo) (proposed)	**73.09**

Table 3: Experimental results using contextual word embeddings on SENSEVAL-2 lexical sample task

6 Conclusion

In this paper, we proposed a semi-supervised graph embeddings based learning method for the WSD task. The efficiency of the proposed method was evaluated on the Semeval-2010 Japanese WSD task and the SENSEVAL-2 English lexical sample task. Experimental results show that the proposed method performs better than the previous semi-supervised WSD method. Moreover, the graph structure between examples is effective for WSD and it is effective to utilize a graph structure obtained by fine-tuning BERT in the proposed method.

In the future, we would like to explore methods to construct an effective graph structure by using paraphrase information, and the dependency analysis technique, the effective filtering method for unlabeled data. In addition, we would like to develop a method to use the examples of the Iwanami's Japanese dictionary effectively.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 18K11422.

Reference

Asahara, M. (2018). NWJC2Vec: Word embedding dataset from 'NINJAL Web Japanese Corpus'. Terminology, 24, pp. 7-22.

Bevilacqua M. and Navigli R. (2020). Breaking Through the 80% Glass Ceiling: Raising the State of the Art in Word Sense Disambiguation by Incorporating Knowledge Graph Information. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL2020), pp. 2854-2864.

Blevins, T. and Zettlemoyer, L. (2020). Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders. Proceedings of the 58th Association for Computational Linguistics (ACL2020), pp. 1006-1017.

Fujita, S., and Fujino, A. (2011). Word Sense Disambiguation by Combining Labeled Data Expansion and Semi-Supervised Learning Method. Proceedings of 5th International Joint Conference on Natural Language Processing, pp. 676-685.

Huang, L., Sun, C., Qiu, X. and Huang, X. (2019). GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3509-3514.

Kågebäck, M., and Salomonsson, H. (2016). Word Sense Disambiguation using a Bidirectional LSTM. Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V), pp.51-56.

Kumar, S., Jat, S., Saxena, K. and Talukdar, P. (2019). Zero-shot Word Sense Disambiguation using Sense Definition Embeddings. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL2019), pp. 5670-5681.

Maekawa, K., Yamazaki, M., Ogiso, T., Maruyama, T., Ogura, H., Kashino, W., Koiso, H., Yamaguchi, M., Tanaka, M., and Den, Y. (2014). Balanced Corpus of Contemporary Written Japanese. Language Resources and Evaluation (LREC2014), pp. 345-371.

National Institute for Japanese Language. (2004). Bunrui Goi Hyo (enlarged and revised version). Dainippon Tosho.

Navigli, R. (2009). Word sense disambiguation: A survey, ACM Computing Surveys, vol. 41, no. 2, pp. 10:1-10:69.

Nishio, M., Iwabuchi, E., and Mizutani, S. (1994). Iwanami Kokugo Jiten Dai Go Han, Iwanami Publisher (in Japanese).

Niu, Z., Ji, D., and Tan, C.L. (2005). Word Sense Disambiguation Using Label Propagation Based Semi-Supervised Learning. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 395-402.

Okumura, M., Shirai, K., Komiya, K., and Yokono, H. (2010). Semeval-2010 task: Japanese WSD., Proceedings of the SemEval-2010, ACL 2010, pp. 69-74.

Raganato, A., Bovi, C.D., and Navigli, R. (2017). Neural Sequence Learning Models for Word Sense Disambiguation. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP2017), pp. 1156-1167.

Shinnou, H., Murata, M., Shirai, K., Fukumoto, F., Fujita, S., Sasaki, M., Komiya, K. and Inui, T. (2015) Classification of Word Sense Disambiguation Errors Using a Clustering Method, Journal of Natural Language Processing vol. 22, no. 5,pp. 319-362.

Sousa, S., Milios, E. and Berton, L. (2020) Word sense disambiguation: an evaluation study of semi-supervised approaches with word embeddings. Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-8

Taghipour, K., and Ng, H.T. (2015). Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL2015), pp. 314-323.

Yang, Z., Cohen, W.W., and Salakhutdinov, R. (2016). Revisiting Semi-Supervised Learning with Graph Embeddings. Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16), pp. 40-48.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics pp. 189-196.

Yuan, D., Richardson, J., Doherty, R., Evans, C., and Altendorf, E. (2016). Semi-supervised Word Sense Disambiguation with Neural Models. Proceedings of the 26th International Conference on Computational Linguistics (COLING2016), pp. 1374-1385.

Zhong, Z., and Ng, H.T. (2010). It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. Proceedings of the ACL 2010 System Demonstrations, pp.78-83.