

All-words Word Sense Disambiguation Using Concept Embeddings

Rui Suzuki, Kanako Komiya, Masayuki Asahara, Minoru Sasaki, Hiroyuki Shinnou

Ibaraki University, 4-12-1 Nakanarusawa, Hitachi, Ibaraki JAPAN,

National Institute for Japanese Language and Linguistics, 10-2 Midoricho, Tachikawa, Tokyo, JAPAN,

{17nm709g@vc.ibaraki.ac.jp, kanako.komiya.nlp, minoru.sasaki.01, hiroyuki.shinnou.0828}

@vc.ibaraki.ac.jp, masayu-a@ninjal.ac.jp

Abstract

All-words word sense disambiguation (all-words WSD) is the task of identifying the senses of all words in a document. Since the sense of a word depends on the context, such as the surrounding words, similar words are believed to have similar sets of surrounding words. We therefore predict the target word senses by calculating the distances between the surrounding word vectors of the target words and their synonyms using word embeddings. In addition, we introduce the new idea of concept embeddings, constructed from concept tag sequences created from the results of previous prediction steps. We predict the target word senses using the distances between surrounding word vectors constructed from word and concept embeddings, via a bootstrapped iterative process. Experimental results show that these concept embeddings were able to improve the performance of Japanese all-words WSD.

Keywords: word sense disambiguation, all-words, unsupervised

1. Introduction

Word sense disambiguation (WSD) involves identifying the senses of words in documents. In particular, the WSD task where the senses of all the words in a document are disambiguated is referred to as all-words WSD. Much research has been carried out, not only on English WSD but also on Japanese WSD, for many years. However, there has been little research on Japanese all-words WSD, possibly because no tagged corpus has been available that was large enough for the task. Usually, the Japanese sense dataset Balanced Corpus of Contemporary Written Japanese (BC-CWJ) (Maekawa et al., 2014), tagged with sense IDs from Iwanami Kokugo Jiten (Nishio et al., 1994), is used for supervised WSD. However, unsupervised approaches to all-words WSD often require synonym information, which sense datasets cannot provide. This paper reports research on Japanese all-words WSD that uses a corpus that is in its infancy, namely BCCWJ annotated with concept tags or article numbers from the Word List by Semantic Principles (WLSP) (Kokuritsukokugokenkyusho, 1964), which is a Japanese thesaurus. In the WLSP, the article numbers indicate shared synonyms.

In the WLSP thesaurus, words are classified and organized by their meanings¹ and each WLSP record contains the following fields: record ID number; lemma number; record type; class; division; section; article; article number; paragraph number; small paragraph number; word number; lemma (with explanatory note); lemma (without explanatory note); reading; and reverse reading. Each record has an article number, which represents four fields: class; division; section; and article. For example, the word “犬”(inu, meaning spy or dog) has two records in the WLSP, and therefore has two article numbers, 1.2410 and 1.5501, indicating that the word is polysemous. In addition, there are 240 semantic breaks in the WLSP, which allow words to be classified in more detail than with the article numbers alone. Note that the article numbers are used as concept tags, because

many words have the same article numbers. Several words can have the same precise article number, even when the semantic breaks are considered.

2. Related Work

WSD methods can broadly be divided into two categories: supervised and unsupervised approaches. Generally, WSD using supervised learning can achieve high accuracy rates, but requires substantial manual effort due to the need for a sufficient amount of manually-annotated training data. On the other hand, unsupervised learning does not need such manual input, but it is difficult to obtain as high an accuracy rate as with the supervised learning.

Many WSD methods have been proposed. WLSP article numbers or hypernyms of target words obtained from the WLSP are often used as supervised learning features. Vu and Parker (2016) proposed the idea of K-embeddings for learning concept embeddings. Komiya et al. (2015) proposed a surrounding word sense model for Japanese all-words WSD using unsupervised learning which assumes that the sense distribution of surrounding words changes depending on the sense in which a polysemous word is used. Shinnou et al. (2017b) proposed a WSD system capable of performing Japanese WSD easily using a supervised approach.

3. WSD Using Synonym Information from the WLSP

We propose three WSD methods that use synonym information from the WLSP: 1) a method using only the word embeddings of synonyms, 2) a method using both the word and concept embeddings of synonyms, and 3) a method using only the concept embeddings of synonyms.

3.1. WSD Using the Word Embeddings of Surrounding Words

Since the senses of words are determined by context, such as the surrounding words, similar words are believed to

¹<https://www.ninjal.ac.jp/english/publication/catalogue/goihyo/>

have similar sets of surrounding words. The proposed method is based on this idea and consists of three steps. First, we generate word embeddings from an untagged corpus and concatenate the embeddings of the words surrounding each target word, creating what we will refer to as the surrounding word vectors. For example, a word “犬” (*inu*, meaning spy or dog) has two article numbers, 1.2410 and 1.5501, in WLSP. Therefore, this word is polysemous. If there was a sentence including the word “犬” below, the surrounding word vector for the word “犬” is the concatenated vector of the word embeddings or the words “警察”, “の”, “だ”, and “.”.

彼は 警察 の 犬 だ .
 he (topic marker) police of spy is .
 ‘He is a spy of police.’

Second, we make synonym lists for the senses of each target word using the WLSP, and create surrounding word vectors for each synonym appearing in the corpus. Note that each surrounding word vector is labeled according to the sense of the target word, which is equivalent to the sense of the synonym. For example, if the target word was “犬”, its synonyms are the words that have the article number, 1.2410 or 1.5501. They are agent, ninja, and so on when the number was 1.2410, and wolf, fox, and so on when it was 1.5501. The surrounding word vectors for these synonyms are created from each synonym appearing in the corpus and are labeled as 1.2410 or 1.5501. These labels can be obtained in an unsupervised manner, that is, it does not make use of sense-tagged data, and this method is knowledge-based because the WLSP is a thesaurus.

Finally, we predict the target word senses using the K-nearest neighbors (KNN) algorithm, based on the distances between the surrounding word vectors for the target words and their synonyms. In other words, we calculate the distances between the surrounding word vectors for the target word “犬” and its synonyms labeled as 1.2410 or 1.5501 and determine the word sense of the target word via KNN algorithm. That is, if the synonym with label 1.2410 was nearer by the algorithm, the word sense of the target word will be 1.2410 and vice versa.

3.2. WSD Using the Word and Concept Embeddings of Surrounding Words

For this method, we repeat the target word sense prediction process (based on the one described in Section 3.1.), with the prediction steps at the n_{th} iteration being as follows.

1. Replace the word tokens in the corpus with their concept tags, using the results from the $n - 1_{st}$ prediction step, and create concept embeddings using the conceptually-tagged corpus.(cf. Figures 1 and 2).

根本 要 が ビデオ ジョッキー を 務め、1960 年代 後半 から 80 年代 前半 の
 ロック を 届ける。
 自民 党 の 「 政策 新 人類 」 渡辺 喜美 衆院 議員 は 提案 する 「 経済 非常 事態 」
 を 宣言 し 産業 再生 委 の 創設 を !
 :

Figure 1: Word Tokens

根本 要 が 1.4620 1.2410 を 2.3320、1.1960 1.1960 1.1960 1.1630 1.1623
 1.1650 から 1.1960 1.1960 1.1630 1.1623 1.1650 の ロック を 2.1521。
 1.2760 党 の 「 1.3084 3.1660 1.5501 」 渡辺 喜美 1.2730 1.2400 は 1.1210 2.1211
 「 1.3710 1.1331 1.1000 」 を 1.3100 2.1211 1.3801 1.1211 委 の 1.1220 を !
 :

Figure 2: Conceptually-tagged Corpus

For example, a text shown in Figure 1 were converted to a concept-text in Figure 2. Here, the word “届ける” (*todokeru*, meaning deliver) is polysemous that has three article numbers, 2.1521, 2.3141, and 2.3830. This word is replaced with 2.1521 in Figure 2 according to the result of the $n - 1_{st}$ prediction step. In addition, the words that have no record in WLSP are not replaced by the article numbers. The words “根本” (*Nemoto*, person name) and “要” (*Kaname*, person name) in Figure 2 are those examples.

2. Generate surrounding word vectors for the target words and their synonyms, namely vectors where the word and concept embeddings have been concatenated. For the first prediction step, use the prediction results for the word embeddings only, as described in Section 3.1.
3. Predict the target word senses using the KNN algorithm, as described in Section 3.1.

For the method using only the concept embeddings of synonyms, we concatenated only the concept embeddings instead of the word and concept embeddings in the second step. We investigate the optimal number of iterations experimentally in Section 4.

3.3. Word List by Semantic Principles

Table 1 shows the structure of WLSP. We extracted メートル (meter) and its synonyms for example. In the WLSP thesaurus, words are classified according to an article number. The article number represents four fields: class, division, section, and article. The class classifies the words into four groups according to a part of speech, and the division, section, and article further classify them by according to a word’s meaning. In addition, there are 240 semantic breaks (“*” in Table 1) in WLSP, which allow words to be classified in more detail than with the article numbers alone. For example, there are approximately 500 counter suffixes (meter, yard, liter, gallon, etc.) that have an article number, 1.1962, in WLSP. Since these words have same article number, they can be deemed as words that have the same meanings. However, if the semantic breaks are took into consideration, they are deemed as two word groups, “meter yard” and “liter gallon.”

3.4. Selecting Synonyms Using the WLSP

First, we find the WLSP article numbers for all words in the corpus. The synonyms used for our methods of Section 3.1. are as follows.

- Words with the same article numbers as the target words. Here semantic breaks are also considered if available.
- Words are excluded if they are synonyms for more than one sense of a given target word.

article number	paragraph number	small paragraph number	word number	lemma
1.1962	15	1	1	メートル (meter)
			2	キロメートル (kilometer)
			3	キロ (kilo)
			⋮	⋮
	18	2	1	ヤード (yard)
	⋮	⋮	⋮	⋮
	23	99	99	*
	24	1	1	リットル (liter)
	⋮	⋮	⋮	⋮
	26	1	1	ガロン (gallon)
⋮	⋮	⋮	⋮	

Table 1: WLSP

For example, imagine that a target word, X , has two senses: Sense 1 and Sense 2. If the synonyms for Sense 1 are A, B, and C, and the synonyms for Sense 2 are C, D, and E, we exclude C from the synonym sets for both Sense 1 and Sense 2.

However, the above conditions can not take into consideration the ambiguity of the synonyms. For example, if a word A, the synonym of X for Sense 1, is polysemous, the surrounding word vectors of A are not necessarily vectors meaning only Sense 1. Therefore, we take into consideration the ambiguity of the synonyms for method in Section 3.2..

The synonyms used at the n_{th} iteration of our method are as follows.

- Polysemous words with the same sense as the target word. For the KNN algorithm at the n_{th} step, we only use surrounding word vectors whose predicted sense was the same as that of the target word at the $n - 1_{st}$ prediction step.
- Monosemous words with the same sense as the target word.
- Words are excluded if they are synonyms for more than one sense of a given target word.

In other words, the surrounding word vectors of A are created not from all word tokens of A, but from only word tokens of A predicted as Sense 1 at the $n-1$ prediction.

4. Experiment

We used the BCCWJ for our experiments, and used the annotation by (Kato et al., 2017) to add word sense annotations. This corpus includes 3,790 word types and 22,568 word tokens, including 1,096 word types and 4,760 word tokens for polysemous words. The polysemous words have an average of 3.16 word senses per word token and an average of 2.59 word senses per word type, so the accuracy rate of a random baseline method would be 31.65% (the inverse of the average number of senses for the polysemous words). The accuracy of the most frequent sense baseline is 91.7%.

Number of URLs collected	83,992,556
Number of sentences (tokens)	3,885,889,575
Number of sentence (types)	1,463,142,939
Number of words (tokens)	25,836,947,421

Table 2: Statistics for the NWJC-2014-4Q Dataset

CBOW or skip-gram	-cbow	1
Dimensionality	-size	200
Number of surrounding words	-window	8
Number of negative samples	-negative	25
Hierarchical softmax	-hs	0
Minimum sample threshold	-sample	1e-4
Number of iterations	-iter	15

Table 3: Parameters Used to Generate NWJC2vec

Note that we cannot know what is the most frequent sense using an unsupervised approach. We used NWJC2vec² (Shinnou et al., 2017a) for the Japanese word embeddings. This is a set of word embeddings generated from the NWJC-2014-4Q dataset, which is an enormous Japanese corpus developed using the word2vec³ tool. Tables 2 and 3 present summary statistics for the NWJC-2014-4Q data and the parameters used to generate the word embeddings, respectively. We used word2vec (Mikolov et al., 2013c; Mikolov et al., 2013a; Mikolov et al., 2013b) to generate the concept embeddings, and the parameters used are summarized in Table 4.

The window size for the surrounding word vectors was set to two, meaning four words in total. When the number of surrounding words was smaller than the window size, we used a zero vector. Therefore, the dimensionality of the surrounding word vectors was 800 when they were created using only the word embeddings, and 1,000 when both the word and concept embeddings were used. It was 200 when only the concept embeddings are used.

We used KNeighborsClassifier from the scikit-learn⁴ library as the KNN algorithm. We tried K values of 1, 3, and 5, as well as uniform and distance-based weights. Default settings were used for all other parameters.

5. Results

Table 5 shows the WSD results when only the word embeddings of the surrounding words were used, and Table 6

CBOW or skip-gram	-cbow	1
Dimensionality	-size	50
Number of surrounding words	-window	5
Number of negative samples	-negative	5
Hierarchical softmax	-hs	0
Minimum sample threshold	-sample	1e-3
Number of iterations	-iter	5
Minimum frequency to consider	-min-count	1

Table 4: Parameters Used for the Concept Embeddings

²<http://nwjc-data.ninjal.ac.jp/>

³<https://code.google.com/archive/p/word2vec/>

⁴<http://scikit-learn.org/stable/>

Weights	K=1	K=3	K=5
Uniform	52.6	53.0	53.0
Distance-based	52.6	52.7	52.7

Table 5: WSD Accuracy Rate Using Word Embeddings Only

K	weight	1	2	3	4	5	6
1	Uniform	57.0	56.2	56.4	56.3	56.5	56.3
1	Distance	57.0	56.2	56.4	56.4	56.5	56.4
3	Uniform	57.1	56.5	56.6	56.5	56.6	56.5
3	Distance	57.1	56.3	56.5	56.4	56.6	56.4
5	Uniform	57.3	56.6	56.8	56.7	56.8	56.7
5	Distance	57.2	56.3	56.6	56.5	56.7	56.5

Table 6: WSD Accuracy Rate Using Both Word and Concept Embeddings, for between One and Six Iterations

shows the results when both the word and concept embeddings were used. Table 7 shows the WSD results when only the concept embeddings of the surrounding words were used. The numbers in the column headings give the numbers of iterations.

Table 5 shows that uniform weights always gave better results than distance-based weights. The best results in Table 5 occurred for $K = 3$ or 5 and uniform weights, where we obtained an accuracy rate of 53.0%. In addition, Table 5 shows that our method significantly outperformed the random baseline, regardless of the K and weight settings used. The best results in Table 6 occurred at the 1_{st} step, for $K = 5$ and uniform weights and we obtained the best results in Table 7 at the 2_{nd} step, for $K = 5$ and distance weights. The results when using only the concept embeddings were best in this experiment.

6. Discussion

Considering our results, we can see that the accuracies in Table 6 are better than those in Table 5, indicating that starting from word-embedding-based predictions and then developing concept embeddings generated from a conceptually-tagged text corpus is effective for WSD.

In addition, Tables 5 and 6 show that the accuracy of WSD using word and concept embeddings is high only when that of WSD using only word embeddings is also high. However, Tables 5 and 7 show that the accuracy of WSD using only concept embeddings does not follow this trend. To study this further, we varied the conditions used to build the synonym lists for the initial predictions to investigate

K	weight	1	2	3	4	5	6
1	Uniform	54.7	58.5	57.2	57.7	58.4	57.8
1	Distance	54.7	58.0	56.9	56.9	58.2	56.6
3	Uniform	54.4	58.3	56.9	58.1	56.3	58.5
3	Distance	53.7	58.4	57.4	56.7	58.4	57.8
5	Uniform	55.2	58.0	57.4	58.2	58.0	57.2
5	Distance	55.8	58.8	57.4	58.6	57.5	57.3

Table 7: WSD Accuracy Rate Using Concept Embeddings Only

Weights	K=1	K=3	K=5
Uniform	51.3	53.3	53.4
Distance-based	51.3	51.7	51.7

Table 8: WSD Accuracy Rate Using Word Embeddings Only, under Condition 1

K	weight	1	2	3	4	5	6
1	Uniform	56.9	56.3	56.3	56.3	56.4	56.4
1	Distance	56.9	56.3	56.3	56.3	56.4	56.3
3	Uniform	54.7	56.1	56.9	56.1	56.8	56.1
3	Distance	56.9	56.4	56.4	56.4	56.5	56.4
5	Uniform	54.9	56.2	57.0	56.2	56.9	56.2
5	Distance	57.0	56.5	56.5	56.5	56.6	56.6

Table 9: WSD Accuracy Rate Using Both Word and Concept Embeddings, under Condition 1, for between One and Six Iterations

the effect of this on WSD performance. We originally excluded words that were synonyms for more than one sense of a given target word, because we believed that KNN classification accuracy would decrease when the same vectors were generated for duplicate synonyms. The condition variations that we now considered were as follows.

1. Ignore semantic breaks.
2. Substitute paragraph numbers for semantic breaks.
3. Use only monosemous words as synonyms.

Tables 8, 9, and 10 show the WSD results when only the word embeddings of the surrounding words were used, when both the word and concept embeddings were used, and when only the concept embeddings of the surrounding words were used under Condition 1, respectively. Tables 11, 12, and 13 are those results under Condition 2 and Tables 14, 15, and 16 are those results under Condition 3.⁵ Under Condition 1, the number of target word synonyms increased compared that for the original conditions, meaning that the synonyms included more words with different senses. Condition 1 caused the accuracy to increase, when using only word embeddings, but the accuracy decreased when using word and concept embeddings. It indicates that the quality of the concept embeddings does not always de-

K	weight	1	2	3	4	5	6
1	Uniform	53.3	58.7	57.4	56.5	57.7	57.1
1	Distance	53.5	58.0	56.5	57.0	56.6	56.7
3	Uniform	54.5	56.4	57.3	58.0	57.2	57.1
3	Distance	53.7	59.5	58.3	59.1	58.4	58.5
5	Uniform	54.1	57.5	58.5	59.1	58.9	57.0
5	Distance	54.0	56.9	59.2	59.2	58.9	58.8

Table 10: WSD Accuracy Rate Using Concept Embeddings Only, under Condition 1, for between One and Six Iterations

⁵We carried out nine iterations under all the conditions but omit the results from $6_{th} - 9_{th}$ iterations under Condition 1 and Condition 2 due to space limitation.

Weights	K=1	K=3	K=5
Uniform	51.5	51.5	51.5
Distance-based	51.5	51.5	51.5

Table 11: WSD Accuracy Rate Using Word Embeddings Only, under Condition 2

K	weight	1	2	3	4	5	6
1	Uniform	56	58	58	57.9	57.9	57.9
1	Distance	56	58	57.9	57.9	57.9	57.
3	Uniform	56.4	58.2	58.2	58.1	58.1	58.1
3	Distance	56.2	58.1	58	58	58	58
5	Uniform	56.4	58.4	58.3	58.2	58.3	58.2
5	Distance	56.2	58.2	58.2	58.1	58.1	58.1

Table 12: WSD Accuracy Rate Using Both Word and Concept Embeddings, under Condition 2, for between One and Six Iterations

pend on the prediction accuracy using only word embeddings. The accuracy increased when using only concept embeddings and we obtained the best results when using only concept embeddings.

Under Condition 2, the synonyms only included words whose senses were closer to those of the target words than they were under the original conditions, since the paragraph numbers allowed the words to be classified in more detail than the semantic breaks were able to achieve. Here, the accuracy decreased when using only word embeddings but the accuracies increased when using word and concept embeddings and when using only concept embeddings. We again obtained the best results when using only concept embeddings.

In contrast, all accuracies increased under Condition 3 (shown in Tables 14, 15 and 16). The best results in this research occurred at the 7th iteration for K = 3 and distance weights, where we obtained an accuracy rate of 59.8%. When polysemous words were used as synonyms, their senses were not necessarily the same as those of the target words, so the quality of the concept embeddings was improved by using only monosemous words as synonyms. In addition, about 70% of the words in the WLSP are monosemous, so the number of synonyms did not significantly decrease under Condition 3, and we believe this is why the accuracies improved. Only under Condition 3, many iterations caused the accuracy to increase. We believe that this is because the quality of the concept embeddings was better.

K	weight	1	2	3	4	5	6
1	Uniform	56.9	59.5	57.7	56.6	57.9	57
1	Distance	57.8	58.7	55.2	58.6	55.7	57.9
3	Uniform	57.9	57.6	54.9	57.9	54.9	55.6
3	Distance	56.6	56.1	58.3	57.1	57	59.1
5	Uniform	57.5	59.2	56.1	59.1	58.8	54.2
5	Distance	57.2	59.1	55.3	55.6	57.1	56.3

Table 13: WSD Accuracy Rate Using Concept Embeddings Only, under Condition 2, for between One and Six Iterations

Weights	K=1	K=3	K=5
Uniform	55.7	55.7	55.7
Distance-based	55.7	55.7	55.7

Table 14: WSD Accuracy Rate Using Word Embeddings Only, Using Condition 3

K	weight	1	2	3	4	5	6
1	Uniform	56.3	55.6	56.6	56.0	57.5	56.0
1	Distance	56.3	55.6	56.6	56.0	57.5	56.0
3	Uniform	56.6	55.7	56.6	56.1	57.5	56.2
3	Distance	56.6	55.7	56.7	56.0	57.6	56.1
5	Uniform	56.7	55.9	56.8	56.3	57.7	56.4
5	Distance	56.7	55.8	56.8	56.1	57.7	56.2

Table 15: WSD Accuracy Rate Using Both Word and Concept Embeddings, under Condition 3, for between One and Six Iterations

7. Conclusion

In this paper, we have proposed three methods for all-words WSD: 1) a method using only word embeddings of synonyms, 2) a method using both word and concept embeddings of synonyms, and 3) a method using only concept embeddings of synonyms. Experimental results for the proposed methods show that they all significantly outperformed a random baseline, indicating concept embedding was effective for WSD. The optimal conditions for selecting synonyms depend on both the corpus size and the target words. In the current study, the accuracies increased when only monosemous words were used as synonyms. We will use our method to annotate corpora.

Acknowledgment

This work includes the result of “Development of All-words WSD System for Creation of Correspondence Table of Word List of Semantic Principles and Iwanami Kokugo Jiten”, the joint research project of National Institute for Japanese Language and Linguistics. This work

K	weight	1	2	3	4	5	6
1	Uniform	58.5	57.4	58.4	57.8	58.3	56.4
1	Distance	59.0	55.8	57.9	56.6	58.1	58.5
3	Uniform	57.5	56.1	57.6	56.6	59.1	57.4
3	Distance	57.7	55.6	57.4	58.3	57.7	59.1
5	Uniform	56.7	58.0	59.0	57.8	59.3	59.2
5	Distance	57.5	56.2	56.6	58.0	59.2	58.0

K	weight	7	8	9
1	Uniform	57.7	57.7	57.7
1	Distance	58.7	56.2	58.5
3	Uniform	58.2	56.5	59.3
3	Distance	59.8	57.6	58.5
5	Uniform	56.7	57.4	58.1
5	Distance	57.1	58.0	59.2

Table 16: WSD Accuracy Rate Using Concept Embeddings Only, under Condition 3, for between One and Nine Iterations

was partially supported by JSPS KAKENHI Grant Number 15K16046 and research grant of Woman Empowerment Support System of Ibaraki University.

References

- Kato, S., Asahara, M., and Yamazaki, M. (2017). Annotation of 'Word List by Semantic Principles' Information on 'Balanced Corpus of Contemporary Written Japanese'. In *Processing of NLP 2017*, pages 306–309 (In Japanese).
- Kokuritsukokugokenkyusho. (1964). *Bunruigoihyo*. Shuei Shuppan, In Japanese.
- Komiya, K., Sasaki, Y., Miruta, H., Sasaki, M., Shinnou, H., and Kotani, Y. (2015). Surrounding Word Sense Model for Japanese All-words Word Sense Disambiguation. In *PACLIC 2015*, pages 35–43.
- 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. In *Language Resources and Evaluation*, volume 48, pages 345–371.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of ICLR Workshop 2013*, pages 1–12.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS 2013*, pages 1–9.
- Mikolov, T., tau Yih, W., and Zweig, G. (2013c). Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of NAACL 2013*, pages 746–751.
- Nishio, M., Iwabuchi, E., and Mizutani, S. (1994). *Iwanami Kokugo Jiten Dai Go Han*. Iwanami Publisher, In Japanese.
- Shinnou, H., Asahara, M., Komiya, K., and Sasaki, M. (2017a). Nwjc2vec: Word Embedding Data Constructed from NINJAL Web Japanese Corpus. *Journal of Natural Language Processing*, 24(4):705–720 (In Japanese).
- Shinnou, H., Komiya, K., Sasaki, M., and Mori, S. (2017b). Japanese all-words WSD system using the Kyoto Text Analysis ToolKit. In *PACLIC 2017*, page no 11.
- Vu, T. and Parker, D. S. (2016). K-Embeddings: Learning Conceptual Embeddings for Words using Context. In *Proceedings of NAACL-HLT 2016*, pages 1262–1267.