Supervised Word Sense Disambiguation with Sentences Similarities from Context Word Embeddings

Shoma Yamaki, Hiroyuki Shinnou, Kanako Komiya, Minoru Sasaki
Ibaraki University, Department of Computer and Information Sciences
4-12-1 Nakanarusawa, Hitachi, Ibaraki JAPAN 316-8511
16nm724r@vc.ibaraki.ac.jp
{hiroyuki.shinnou.0828, kanako.komiya.nlp, minoru.sasaki.01}
@vc.ibaraki.ac.jp

Abstract

In this paper, we propose a method that employs sentences similarities from context word embeddings for supervised word sense disambiguation. In particular, if N example sentences exist in training data, an N-dimensional vector with N similarities between each pair of example sentences is added to a basic feature vector. This new feature vector is used to train a classifier and identification. We evaluated the proposed method using the feature vectors based on Bag-of-Words, SemEval-2 baseline as basic feature vectors and SemEval-2 Japanese task. The experimental results suggest that the method is more effective than the method with only basic vectors.

1 Introduction

Conventionally, the meaning of a word has been represented using a high-dimensional sparse Bag-of-Words (BoW) vector. Recently, there has been considerable interest in word embeddings, where words meanings are represented by low-dimensional and dense vectors using deep learning. With word embeddings, the distance between words can be measured more precisely than that provided by a vector based on the BoW model. Therefore, word embeddings has been used effectively for various natural language processing tasks. With regard to word sense disambiguation (WSD) tasks, some studies have considered that the word embeddings comprise embeddings of word senses(Chen et al., 2014)(Neelakantan et al., 2014)(Sakaizawa and Komachi, 2015)(Bhingardive et al., 2015);however, these studies only consider unsupervised WSD. To the best of our knowledge, the only study that addresses supervised WSD with word embeddings is by Sugawara(Sugawara et al., 2015). In Sugawara's method, one BoW-based vector and one vector based on context word embeddings (CWE) are merged, and they are used for training a classifier and identification. The method proposed by Sugawara is more effective than the method that only uses a vector based on the BoW model. However, we have found two problems with this method. First, it restricts the position of the word in the context. Second, it includes function words. In this paper, we propose a method that addresses both problems. Specifically, if N example sentences exist in training data, an N-dimensional vector that consists of the similarities between each pair of example sentences is added to a basic feature vector. This new feature vector is used for training a classifier and identification. The similarity between sentences is calculated using CWE. This solves the first problem. In addition, the proposed method only uses content words to calculate similarities between example sentences, which solves the second problem. We used SemEval-2 Japanese task to compare Sugawara's method and the proposed method. We found that the proposed method demonstrated higher precision. Furthermore, we performed experiments with basic features used in SemEval-2 baseline system and determined that the proposed method gave better results.

2 Word Embedding for WSD

Feature vectors can be created using the words around a target word in a sentence. This method can present a context of the target word with the vector in a binary representation. Therefore, unknown words cannot be handled.

To address this problem, superordinate concepts in a thesaurus are used because it provides the similarities between different words.

Thus, using a thesaurus is effective for WSD. In this paper, we propose to increase the accuracy of WSD using word embedding as a thesaurus.

3 Sentences Similarities

Sugawara's supervised WSD method represents features using one vector based on the BoW model and another vector that consists of CWE (the context is five words before and after a target word). For example, when the five words before a target word are $(w_{-5}, w_{-4}, w_{-3}, w_{-2}, w_{-1})$ and the five words after the target word are $(w_1, w_2, w_3, w_4, w_5)$, the features vector comprises a binary vector based on the BoW model $(1, 0, 0, 1, 0, \cdots, 1)$ and a vector with word embeddings $(v_{w-1}, v_{w-2}, \cdots, v_{w4}, v_{w5})$ as shown in Figure 1. Sugawara's experimental results suggested that word embeddings useful for WSD. However, we found following 2 problems in his method;

1. It restricts a position of a word in the context.
2. It includes function words.

Therefore, we propose a method that uses the similarities between example sentences from word embeddings to address these problems.

The similarities between two sentences are defined as the average of the cosine of each word embedding in sentences, then i-th sentence (V_i) and j-th sentence (V_j) in training data, and the similarities between V_i and V_j are expressed as follows:

$$\boldsymbol{V}_i = (\boldsymbol{v}_{wi-1}, \boldsymbol{v}_{wi-2}, ..., \boldsymbol{v}_{wi4}, \boldsymbol{v}_{wi5})$$

$$\boldsymbol{V}_j = (\boldsymbol{v}_{wj-1}, \boldsymbol{v}_{wj-2}, ..., \boldsymbol{v}_{wj4}, \boldsymbol{v}_{wj5})$$

$$sim(i,j) = \frac{\sum_{\boldsymbol{v}_{iw}}^{\boldsymbol{V}_i} \sum_{\boldsymbol{v}_{jw}}^{\boldsymbol{V}_j} cos(\boldsymbol{v}_{iw}, \boldsymbol{v}_{jw})}{|\boldsymbol{V}_i| \cdot |\boldsymbol{V}_j|}$$

When only content words are used to calculate similarities, all function words are removed from V_i, V_j.

4 Proposed Method

The proposed method employs a new features vector comprising the basic vector and a vector using the similarities between example sentences with word embedding. As mentioned previously, Sugawara's method employs a features vector comprising a vector based on the BoW model and a vector comprising CWE. However, the proposed method employs a new features vector comprising a vector based on the BoW model and a vector comprising the similarities between sentences from CWE (Figure 2).

In our experiments, we denote the method that includes content words and function words in features words to calculate similarities as "Proposed Method (1)" and the method that does not include function words as "Proposed Method (2)."

5 Features with Thesaurus

The grain size of thesaurus is the important problem in WSD (Shinnou et al., 2015). On the other hand, concepts of words are continuance because distance between words can be calculated with word embeddings. Therefore, it is assumed that using word embedding instead of thesaurus can increase accuracy of WSD.

We implement the SemEval-2 baseline system as a general method using thesaurus. The training algorithm is linear SVM (Support Vector Machine) and features are following twenty things (PoS; Part of Speech, w_i; a word positioned in context)

```
e1=2 previous word, e2=the PoS,
e3=the sub PoS,
e4=1 previous word, e5=the PoS,
e6=the sub PoS,
e7=target word, e8=the PoS,
e9=the sub PoS,
e10=1 following word, e11=the PoS,
e12=the sub PoS,
e13=2 following word, e14=the PoS,
e15=the sub PoS,
e16=relation,
e17=ID of 2 previous word
```

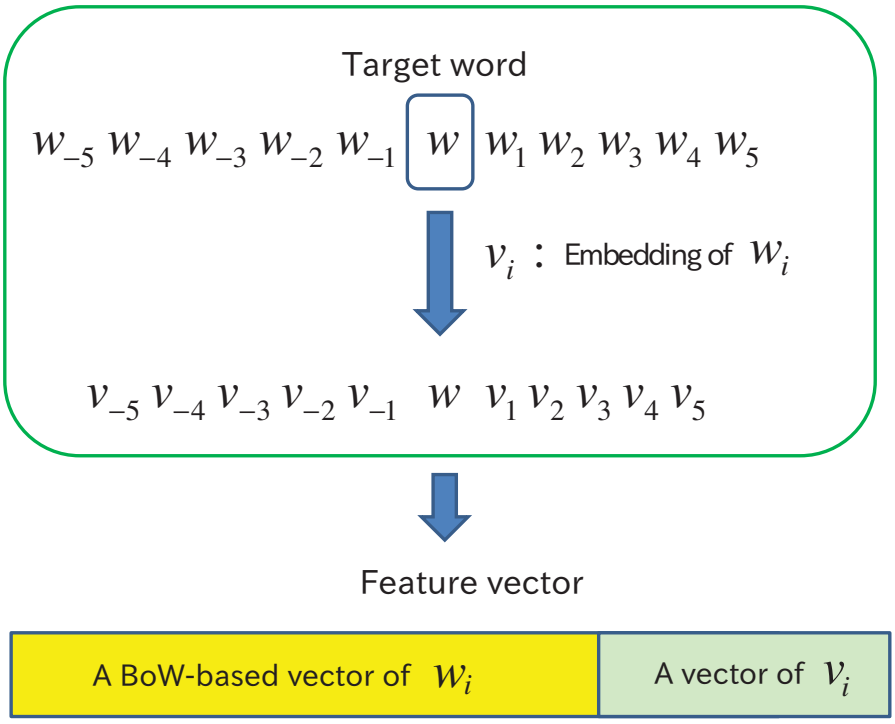

Figure 1: Feature vector in Sugawara's method

```
    in thesaurus,
e18=ID of 1 previous word
    in thesaurus,
e19=ID of 1 following word
    in thesaurus,
e20=ID of 2 following word
    in thesaurus
```

We use only the five character ID in thesaurus although both of the four and five character ID are used in the conventional baseline system. Moreover, the features vector for `e17, e18, e19, e20` are multiple because there are several ID for one word.

This features can be divided into two features; non-thesaurus features from `e1` to `e16` (std-0) and thesaurus features from `e1` to `e20` (std-1). We use two vectors based on std-0 and std-1 as the basic vectors to create the new features vector that the each of basic vectors and the similarities vector are merged. The new features vector are used in the experiments to confirm whether it can increase accuracy of WSD using word embeddings instead of thesaurus.

6 Experiments

6.1 Set-up

We used the SemEval-2 Japanese task in the experiments. This data consists of fifty multivocals. Fifty training data and fifty test data are for each multivocals. Both of training data and test data are adopted morpheme analysis and saved as XML format.

Word embeddings are 200-dimensional vectors calculated by word2vec [1] with Japanese articles in wikipedia.

We used the linearSVC of scikit-learn[2] to make the classifier and set its normalize parameter C to 1.0 .

In addition, we defined content words to the words whose the part of speech is noun, verb, adjective or adverb.

6.2 Results

First, we performed an experiment to confirm that Sugawara's method is to determine whether it is valid for the SemEval-2 Japanese task. The accu-

[1]https://code.google.com/p/word2vec/
[2]http://scikit-learn.org/stable/index.html

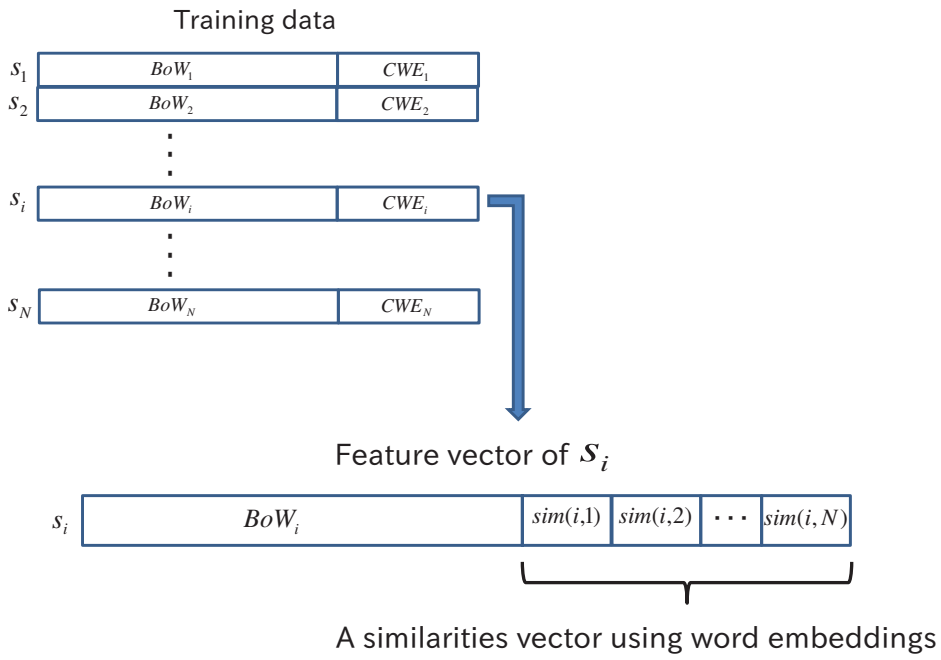

Figure 2: Features vector of training data in the proposal method

racy of the BoW features and the BoW+CWE features are shown in Table 1.

features	accuracy
BoW	0.716
BoW + CWE	0.745

Table 1: Result of the BoW and the BoW+CWE

The result suggested that the method can obtain higher accuracy than the BoW.

Second, we performed an experiment to compare the method using the BoW+CWE and our proposed method. The result is shown in Table 2.

features	accuracy
BoW + CWE	0.745
Proposal method (1)	0.753
Proposal method (2)	0.754

Table 2: Result of the BoW+CWE and the proposed method

The result suggested that the proposed method can obtain better accuracy than the BoW+CWE method. It was found that the proposed method (2) has obtained higher accuracy than proposed method (1). The accuracy for each of the target words is summarized in Table 4. The numbers in bold represents the maximum values for each of the target words, and the underlined numbers represents the number of the strictly larger by comparing the proposed method and the BoW+CWE.

Likewise, the experimental result using std-0 and std-1 as the basic vectors are shown in Table 3

features	accuracy
std-0	0.757
std-1	0.769
std-0 + similarities	0.761
std-1 + similarities	0.771

Table 3: Accuracy of std-0, std-1 and similarities

The result suggested that using the vectors comprising the each of basic vectors and the similarities vector can be obtained the higher accuracy than only using the basic vector. The accuracy for each of the target words is summarized in Table 5.

7 Discussions

We performed the experiment using the vectors based on the BoW, std-0 and std-1 as the basic vec-

tors, it was found that the vector merged the basic vectors and sentence similarities vectors can produce higher accuracy than only the basic vectors. By comparing the result of BoW+CWE and the proposed method for each of the target words, the proposed method got strictly higher accuracy than the BoW+CWE in sixteen words and got lower accuracy in twelve words. Furthermore, by comparing the result of the std-0 and the proposed method, the proposed method got strictly higher accuracy than std-0 in ten words and got lower accuracy in three words. Likewise, by comparing the result of std-1 and the proposed method, the proposed method got higher accuracy in five words and got lower accuracy in one word. Therefore, the proposed method is considered to be effective in improving accuracy of WSD.

By comparing the result of the proposed method (1) and (2) in Table 4, the proposed method (1) got higher accuracy than the proposed method (2) in three words and got lower accuracy in four words. The accuracy rate of the method (2) was higher than the method (1) by 0.001. Therefore, we found that the superiority of the proposed method (2) was very slight.

A purpose of this experiment is to confirm whether that using word embeddings instead of a thesaurus can improve the accuracy of WSD. According to the accuracy rate in Table 3, the accuracy of the std-1 (0.769) is lower than the accuracy of the std-0 + similarities (0.761). This result suggested that the method using thesaurus is more effective for WSD than the method using the similarities between example sentences. However, it is assumed that the method using word embeddings instead of a thesaurus can improve the accuracy of WSD because of following reasons; there are a lot of methods other our proposing, and the quality of word embeddings depend on quality and quantity of text corpora.

8 Conclusion

In this paper, we have proposed a method that uses sentences similarities from CWE for supervised WSD. Specifically, if N example sentences exist in training data, an N-dimensional vector with N similarities between each pair of example sentences is added to a basic feature vector. We performed experiments with basic features used in a SemEval-2 baseline system and determined that the proposed method gave more accurate results than a previous method with only the basic features vector. The results suggested that the proposed method improves the accuracy of WSD. In future, we plan to confirm whether the method can further improve WSD by using word embeddings trained from other text corpora.

References

Sudha Bhingardive, Dhirendra Singh, V Redkar Murthy, Hanumant Redkar, and Pushpak Bhattacharyya. 2015. Unsupervised Most Frequent Sense Detection using Word Embeddings. In *HLT-NAACL-2015*, pages 1238–1243.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A Unified Model for Word Sense Representation and Disambiguation. In *EMNLP-2014*, pages 1025–1035.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. In *EMNLP-2014*, pages 1059–1069.

Yuya Sakaizawa and Mamoru Komachi. 2015. Paragraph vector wo mochiita kyoushi nashi gogi aimaisei kaishou no kousatsu (in japanese). In *NLP 2015*, P1-29.

Hiroyuki Shinnou, Minoru Sasaki, and Kanako Komiya. 2015. Gogi aimaisei kaishou ni okeru thesaurus riyou no mondai bunseki (in japanese). In *NLP 2015*, P1-15.

Hiromu Sugawara, Hiroya Takamura, Ryohei Sasano, and Manabu Okumura. 2015. Context Representation with Word Embeddings for WSD. In *PACLING-2015*, pages 149–155.

Table 4: Accuracy of the each target words (1)

target words	BoW	BoW+CWE	proposed method (1)	proposed method(2)
相手 (aite)	**0.82**	**0.82**	**0.82**	**0.82**
会う (au)	0.60	0.70	0.70	0.70
上げる (ageru)	0.36	0.36	0.44	0.42
与える (ataeru)	0.64	0.64	0.66	0.68
生きる (ikiru)	**0.94**	**0.94**	**0.94**	**0.94**
意味 (imi)	0.38	0.52	0.64	**0.68**
入れる (ireru)	0.72	**0.74**	**0.74**	**0.74**
大きい (ookii)	**0.94**	**0.94**	**0.94**	**0.94**
教える (oshieru)	0.22	0.34	0.38	0.38
可能 (kanou)	0.68	**0.74**	0.62	0.60
考える (kangaeru)	**0.98**	**0.98**	**0.98**	**0.98**
関係 (kankei)	0.82	0.88	**0.96**	**0.96**
技術 (gijutsu)	0.84	0.84	**0.86**	**0.86**
経済 (keizai)	**0.98**	**0.98**	**0.98**	**0.98**
現場 (genba)	0.74	0.74	0.74	0.74
子供 (kodomo)	0.60	0.54	0.44	0.42
時間 (jikan)	0.86	0.84	**0.88**	**0.88**
市場 (shijou)	0.58	**0.64**	0.60	0.60
社会 (shakai)	**0.86**	**0.86**	**0.86**	**0.86**
情報 (jouhou)	0.70	0.76	0.82	0.82
進める (susumeru)	0.44	0.58	0.86	0.86
する (suru)	0.54	0.66	**0.72**	**0.72**
高い (takai)	0.86	0.86	0.86	0.86
出す (dasu)	0.40	0.46	0.40	0.40
立つ (tatsu)	0.46	0.50	0.58	**0.60**
強い (tsuyoi)	**0.92**	**0.92**	**0.92**	**0.92**
手 (te)	**0.78**	**0.78**	**0.78**	**0.78**
出る (deru)	0.62	**0.66**	0.58	0.58
電話 (denwa)	0.78	0.78	0.78	0.78
取る (toru)	0.24	0.26	**0.32**	**0.32**
乗る (noru)	0.56	0.58	0.60	0.60
場合 (baai)	0.86	**0.88**	0.84	0.84
入る (hairu)	**0.66**	**0.66**	**0.66**	**0.66**
はじめ (hajime)	0.90	**0.96**	**0.96**	**0.96**
始める (hajimeru)	0.78	0.80	0.78	0.78
場所 (basho)	0.94	**0.96**	**0.96**	**0.96**
早い (hayai)	0.58	0.66	0.62	0.62
一 (ichi)	**0.92**	**0.92**	**0.92**	**0.92**
開く (hiraku)	**0.90**	**0.90**	0.88	0.88
文化 (bunka)	**0.98**	**0.98**	**0.98**	**0.98**
他 (hoka)	**1.00**	**1.00**	**1.00**	**1.00**
前 (mae)	0.66	0.76	**0.78**	**0.78**
見える (mieru)	0.60	0.60	0.58	0.58
認める (mitomeru)	0.80	0.80	0.78	0.78
見る (miru)	**0.80**	**0.80**	**0.80**	**0.80**
持つ (motsu)	0.64	0.74	0.76	0.76
求める (motomeru)	**0.76**	0.74	0.74	**0.76**
もの (mono)	**0.88**	**0.88**	**0.88**	**0.88**
やる (yaru)	0.94	**0.96**	**0.96**	**0.96**
良い (yoi)	0.36	0.40	0.38	0.38
average	0.716	0.745	0.753	0.754

Table 5: Accuracy of each target words (2)

target words	std-0	std-1	std-0 + similarities	std-1 + similarities
相手 (aite)	0.78	0.80	0.78	0.80
会う (au)	0.88	0.92	0.90	0.92
上げる (ageru)	0.44	0.52	0.48	0.56
与える (ataeru)	0.76	0.70	0.74	0.70
生きる (ikiru)	0.94	0.94	0.94	0.94
意味 (imi)	0.48	0.44	0.46	0.46
入れる (ireru)	0.74	0.74	0.74	0.74
大きい (ookii)	0.94	0.94	0.94	0.94
教える (oshieru)	0.36	0.52	0.40	0.52
可能 (kanou)	0.68	0.64	0.68	0.64
考える (kangaeru)	0.98	0.98	0.98	0.98
関係 (kankei)	0.96	0.96	0.96	0.96
技術 (gijutsu)	0.84	0.82	0.84	0.82
経済 (keizai)	0.98	0.98	0.98	0.98
現場 (genba)	0.74	0.76	0.74	0.76
子供 (kodomo)	0.60	0.62	0.60	0.60
時間 (jikan)	0.86	0.84	0.86	0.86
市場 (shijou)	0.52	0.56	0.52	0.56
社会 (shakai)	0.86	0.86	0.86	0.86
情報 (jouhou)	0.86	0.84	0.86	0.84
進める (susumeru)	0.92	0.92	0.92	0.92
する (suru)	0.64	0.72	0.66	0.72
高い (takai)	0.86	0.88	0.86	0.88
出す (dasu)	0.40	0.50	0.42	0.50
立つ (tatsu)	0.52	0.50	0.52	0.52
強い (tsuyoi)	0.92	0.90	0.92	0.90
手 (te)	0.78	0.78	0.78	0.78
出る (deru)	0.52	0.52	0.52	0.52
電話 (denwa)	0.84	0.78	0.80	0.78
取る (toru)	0.26	0.28	0.26	0.28
乗る (noru)	0.78	0.78	0.78	0.78
場合 (baai)	0.84	0.84	0.84	0.84
入る (hairu)	0.54	0.56	0.54	0.56
はじめ (hajime)	0.88	0.88	0.88	0.88
始める (hajimeru)	0.88	0.86	0.88	0.86
場所 (basho)	0.90	0.96	0.92	0.96
早い (hayai)	0.70	0.70	0.72	0.72
一 (ichi)	0.92	0.90	0.92	0.90
開く (hiraku)	0.78	0.84	0.80	0.84
文化 (bunka)	0.98	0.98	0.98	0.98
他 (hoka)	1.00	1.00	1.00	1.00
前 (mae)	0.76	0.76	0.76	0.76
見える (mieru)	0.68	0.70	0.68	0.70
認める (mitomeru)	0.76	0.82	0.78	0.82
見る (miru)	0.78	0.78	0.78	0.78
持つ (motsu)	0.78	0.80	0.78	0.80
求める (motomeru)	0.64	0.76	0.68	0.76
もの (mono)	0.88	0.88	0.88	0.88
やる (yaru)	0.96	0.96	0.96	0.96
良い (yoi)	0.56	0.54	0.56	0.54
average	0.757	0.769	0.761	0.771