Generation and Evaluation of Concept Embeddings Via Fine-Tuning Using Automatically Tagged Corpus

Kanako Komiya Daiki Yaginuma Masayuki Asahara Hiroyuki Shinnou

Ibaraki University
4-12-1, Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan
{kanako.komiya.nlp, 18nm740n, hiroyuki.shinnou.0828}@vc.ibaraki.ac.jp

National Institute for Japanese Language and Linguistics
10-2 Midoricho, Tachikawa, Tokyo, Japan
masayu-a@ninjal.ac.jp

Abstract

Word embeddings are used in various fields of natural language processing. The use of word embeddings and concept or word sense embeddings demonstrated effectiveness in many tasks, such as machine translation and text summarization. However, it is difficult to obtain a sufficiently large concept-tagged corpus, as the annotation of concept-tags is time-consuming. Therefore, in this paper, we propose a method for generating concept embeddings of Word List by Semantic Principles, a Japanese thesaurus, using both a corpus tagged by an all-words word sense disambiguation (WSD) system and a manually tagged corpus. We generated concept embeddings via fine-tuning using both an automatically tagged corpus and a small manually tagged corpus. In this paper, we propose a novel method of evaluating concept embeddings using the tree structure of Word List by Semantic Principles. Experiments revealed the effectiveness of fine-tuning. The best performance was achieved when the concept embeddings were initially trained with a corpus tagged by an all-words WSD system and retrained with a manually tagged corpus.

1 Introduction

In this paper, we propose a technique for generating concept embeddings using fine-tuning and two types of corpora. In recent years, word embeddings, which are distributed representations of words with low-dimensional vectors, and concept [1] (or word sense) embeddings demonstrated their effectiveness in a number of tasks, such as machine translation and text summarization.

Word embeddings are usually generated using text corpora. It is possible to generate concept embeddings by the same method used to generate word embeddings if the word sequence (i.e., text corpus) is replaced with a concept sequence constructed from a concept-tagged corpus. However, it is difficult to obtain a sufficiently large concept-tagged corpus because the annotation of concept tags is time-consuming. There have been several studies that assigned word senses using the all-words word sense disambiguation (WSD) method (Edmonds and Cotton, 2001), (Snyder and Palmer, 2004), (Navigli et al., 2007), (Iacobacci et al., 2016), (Raganato et al., 2017a), (Raganato et al., 2017b), (Suzuki et al., 2018), (Shinnou et al., 2018). As a result, it is possible to create a concept-tagged corpus using the methods proposed in these studies. However, the results of all-words WSD systems are not always correct; therefore, an automatically tagged corpus created via all-words WSD may not be suitable for generating concept embeddings.

In this paper, we generate concept embeddings of Word List by Semantic Principles (WLSP) (National Institute for Japanese Language and Linguistics, 1964), a Japanese thesaurus, from manually and automatically tagged corpora. First, concept embeddings are generated from a concept-tagged corpus tagged by an all-words WSD system and are fine-tuned using a small, highly accurate corpus in which the concept tags are manually annotated. For comparison, we also generate the following concept em-

[1] Concept refers to a meaning unit of Word List by Semantic Principles.

beddings: (1) concept embeddings generated from only a small, highly accurate corpus in which the concept tags are manually annotated, (2) concept embeddings generated from only a concept-tagged corpus tagged by an all-words WSD system, and (3) concept embeddings initially trained with a small, highly accurate corpus in which the concept tags are manually annotated and fine-tuned using a concept-tagged corpus tagged by an all-words WSD system. The obtained concept embeddings are evaluated by rankings measured by the distances between the concept embeddings based on the tree structure of WLSP, which is a proposed evaluation method in this paper.

2 Related Work

In recent years, word embeddings have been widely used in various fields of natural language processing. In addition, there have been a number of studies on the generation of concept (or word sense) embeddings.

For example, a study by Ouchi et al. (2016), to construct distributed representations of word senses, the authors utilized the distributed representations of synonyms of each word sense. In addition, Yamaki et al. (2017) proposed a method for constructing sense embeddings using training data with sense tags and the multi-sense skip-gram (MSSG) model, which considers the frequency of each word sense. However, these studies did not use a sense-tagged corpus, but rather, a regular text corpus and word embeddings.

Word embeddings are usually generated using a text corpus that is a word sequence. Concept or word sense embeddings can be generated using the same tools as for a sense-tagged corpus, that is, a word sense sequence or concept sequence instead of a text corpus. However, it is generally difficult to obtain a sufficiently large sense-tagged corpus, as only several are available and most are small.

If there are insufficient tagged corpora, automatic generation of tagged corpora may be helpful. A concept-tagged corpus can be automatically created with the all-words WSD system. There are several studies on all-words WSD systems. For example, in studies by Raganato et al. (2017a) and Shinnou et al. (2018), all-words WSD is considered a labeling problem in which every word is assigned a concept tag. Using an automatic tagger, it is possible to create a concept-tagged corpus. However, an automatic tagger does not always produce correct results. For example, there may be cases in which concept tags are not assigned to new words. In these cases, the concept-tagged corpus would not be suitable for generating concept embeddings.

Therefore, in this study, we generate concept embeddings of WLSP using two types of corpora: a large corpus in which the concept tags are assigned using the all-words WSD method and a manually tagged corpus.

3 Generation of Concept Embeddings

We generated four types of vectors using two corpora tagged with concepts from WLSP.

3.1 WLSP

WLSP is a Japanese thesaurus in which a word is classified and ordered according to its meaning. A WLSP record is composed of the record ID number, lemma number, record type, class, division, section, article, concept number, paragraph number, small paragraph number, word number, lemma with explanatory note, lemma without explanatory note, reading and reverse reading. The concept number consists of a category, medium item, and classification item. In WLSP, some words are polysemous; for example,“子供 (child or children)” is a polyseme, and two concepts are registered in WLSP: 1.2050 and 1.2130 (Table 1).

The tree structure of WLSP is illustrated in Figure 1.

3.2 Corpora

In this study, we used two concept-tagged corpora based on the Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa et al., 2014). The first corpus is a large corpus in which concept tags were automatically assigned using the all-words WSD method. We used an all-words WSD tagger proposed by (Shinnou et al., 2018). Hereinafter, this corpus is referred to as the all-words WSD corpus. The second corpus is a small corpus in which concept tags were manually assigned. We used the annotation data of WLSP by the National Institute of Japanese Language and Linguistics (Kato et al.,

Concept number	Class	Division	Section	Article
1.2050	Nominal words	Agent	Human	Young or old
1.2130	Nominal words	Agent	Family	Child or descendant

Table 1: Concept tags and their corresponding class, division, section, and article of "子供 (child or children)" from Word List by Semantic Principles

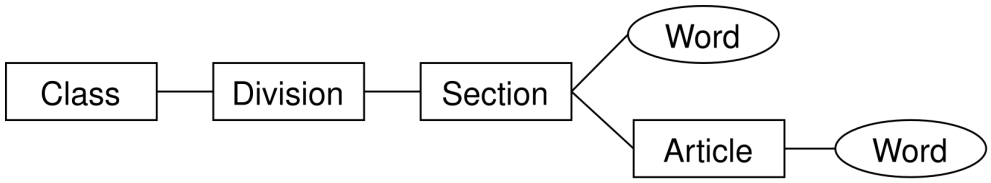

Figure 1: Tree structure of Word List by Semantic Principles

2018). This corpus is in its infancy. Hereinafter, this corpus is referred to as the manual corpus. There are two types of BCCWJ: the core and non-core data. For the core data, the word tokenization is manually conducted, but for the non-core data, word tokenizer, MeCab with Unidic dictionary is used for the word tokenization. The core data includes approximately 1,300,000 words and the non-core data includes approximately 25,800,000,000 words. The core data is included in the non-core data. We used the non-core data including the core data for the all-words WSD corpus, with the concept tag annotation via the all-words WSD system. The manual corpus is the part of the core data with manual annotation of the concept tags, which includes approximately 340,000 words.

Examples of the text corpus and a generated concept sequence are presented in Table 2. In the table, an original Japanese text, its English translation and concept sequence are shown. The concepts of "なく" and "ない" are both 3.1200 because they are the same words after lemmatization. Table 3 presents the number of words, vocabulary, and concepts in each corpus.

3.3 Vectors

In this study, word2vec[2] (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c) was used to generate concept embeddings. Then, fine-tuning was performed. Fine-tuning is a method in which generated distributed representations are given as initial values and retrained with a new corpus. The following four types of concept embeddings were created:

- All-words WSD vector: concept embeddings were trained with the all-words WSD corpus.
- All-words WSD-fine vector: concept embeddings were trained with the all-words WSD corpus and retrained with a manual corpus.
- Manual vector: concept embeddings were trained with a manual corpus.
- Manual-fine vector: concept embeddings were trained with a manual corpus and retrained with the all-words WSD corpus.

When fine-tuning the embeddings, vectors of the new words in the new corpus were generated if the number of occurrences of the new words exceeded the threshold value.

4 Evaluation of Concept Embeddings

We evaluated the concept embeddings using WLSP. Because WLSP has a tree structure, we assume that concepts that belong to the same node are similar to each other. Figure 2 presents an example of leaves of WLSP. In this figure, we assume that the concept of *wolf* is closer to that of *hyena* than that of *cat* or *dog*. Based on this assumption, evaluation of the generated concept embeddings was performed.

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

Text	モノでなく心ではないのか
English translation	It is not a thing but a heart, isn't it?
Concept sequence	1.4000 で 3.1200 1.3000 では 3.1200 の か

Table 2: Example of concept-tagged corpus

Concept Embeddings	Words	Vocabulary	Concepts
All-words WSD corpus	23,968,826	75,028	851
Manual corpus	347,094	3,164	916

Table 3: Number of words, vocabulary, and concepts in each corpus

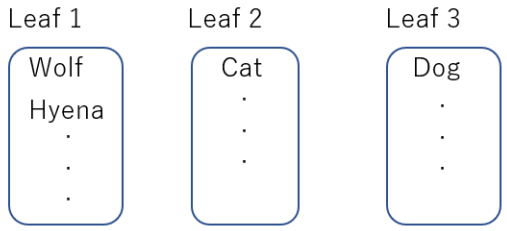

Figure 2: Example of leaves of Word List by Semantic Principles

4.1 Evaluation Procedure

The evaluation procedures were as follows.

1. For each concept c of the concept embeddings e, identify a corresponding leaf node n in WLSP.

   For example, if c is the concept of *wolf*, the corresponding node n includes concepts such as *hyena*. In Figure 2, n is Leaf 1. In this method, we assume that every concept has at least two words so that the distance between them can be calculated.

2. Obtain a sibling leaf node set N of n.

   A sibling leaf node set N includes a node that contains a concept such as *cat* and another node that contains a concept such as *dog*. In Figure 2, N includes Leaves 2 and 3.

3. Calculate d_c, the average distance between e and the concept embeddings of all concepts in n except for c.

   For this step, we calculated d_c, the average distance between the concept embeddings of *wolf* and the concept embeddings of *hyena* and other concepts in n (Leaf 1). We used the arithmetic mean to average the distance.

4. Calculate the average distances $d_1 ... d_{|N|}$ between e and the concept embeddings of all concepts in each leaf node in N.

   We calculated the average distance between concept embeddings of *wolf* and the concept embeddings of all concepts from the node containing *cat*, and obtained d_1. Likewise, we calculated the average distance between the concept embeddings of *wolf* and the concept embeddings of all concepts from the node containing *dog*, and obtained d_2. Following this step, we obtained the averaged distances $d_1 ... d_{|N|}$.

5. Obtain the ranking of n compared with all nodes in N based on the average distance from e_i.

   We compared $d_1 ... d_{|N|}$ and d_c, and obtained the ranking of d_c. For example, if d_c was the second shortest in $d_1 ... d_{|N|}$ and d_c, n was in second place.

6. Obtain the closest distance d_{close} and the closest leaf node to e based on the average distance.

   We obtained the closet leaf node to e. For example, if the closest leaf node to the concept *wolf* was the node that contained the concept *dog*, d_2 would be the shortest, which signifies that $d_{close} = d_2$.

7. Obtain $d_c - d_{close}$.

We calculated $d_c - d_{close}$, which is the difference between the average distances from the concept in first place. In other words, we calculated the difference between the average distance between *wolf* and concepts such as *hyena*, which is the node that *wolf* belongs to in WLSP, and the average distance between *wolf* and concepts such as *dog*, which was in first place. If all rankings of n were first place, the difference would be zero.

In this manner, we evaluated the concept embeddings that were generated using ranking and $d_c - d_{close}$.

4.2 Experimental Settings

For the parameters used in the calculation of word2vec, we used 200 dimensions, 5 window sizes, 1,000 batch sizes, and 5 iterations. We used CBOW as the algorithm. The training parameters used for fine-tuning were identical to the ones used when the original concept embeddings were generated in advance. Cosine similarity was used to compare the distances between the generated concept embeddings.

4.3 Results

Table 4 presents the average ranking of the correct nodes, which are the nodes that each concept whose embeddings were generated by this method belonged to. Table 4 also displays the average difference between the closest leaf node and the correct nodes, and the average number of leaf nodes. This table indicates that the poorest ranking of each concept embeddings was 6.868 for the manual vector. Because the average number of leaf nodes was 42, the average ranking of a random selected node was approximately 21. This suggests that, even when concept embeddings were generated using the worst method, the ranking of the nodes produced better results than when the random baseline was used.

5 Discussion

Table 4 indicates that the average ranking and difference of the all-words WSD-fine vector were smaller than those of the all-words WSD vector. In addition, the average ranking and difference of manual-fine vector were smaller than those of the manual vector. Smaller values of the average ranking and difference indicate better performance; therefore, these results demonstrate that fine-tuning improved the vectors. Table 4 also indicates that the results of the all-words WSD vector were superior to those of the manual vector, while the all-words WSD-fine vector was superior to the manual-fine vector. The poorest results were associated with the manual vector. These results suggest that the all-words WSD corpus is effective for generating concept embeddings without fine-tuning or for initial training of fine-tuning. We believe that a large corpus is necessary for generating improved word embeddings. We used the same parameters of word2vec for all vectors, which were tuned so that the results of the manual vector, the method with the poorest performance, could achieve the best performance. The other three vectors (i.e., all-words WSD vector, all-words WSD-fine vector, manual-fine vector) could be improved if the parameters were tuned for each method. This is because the results of vectors often improve when the parameters are tuned depending on the size and characteristics of the corpora. Table 5 presents the evaluation results of the all-words WSD vector and manual vector generated with 10 iterations. Other parameters are identical to those used in the experiments presented in Table 4. The results in Table 5 are inferior to those in Table 4; therefore, extensive experiments are necessary to tune the parameters suitable for each corpus.

The number of words in the all-words WSD corpus was approximately 69 times larger than the number of words in the manual corpus (see Table 3). In addition, according to Shinnou et al. (2018), the accuracy of the WSD system was approximately 80% for all words and approximately 70% for all ambiguous words in the test corpus (the annotation data of WLSP). In our experiments, the test corpus would be identical to the manual corpus and sub-corpus of the all-words WSD corpus if concept tags were removed and manually tagged. Therefore, we assume that the accuracy of the all-words WSD corpus would be approximately 70% or 80%. The results of the concept embeddings trained with the all-words WSD corpus were superior to the results of the concept embeddings trained with the manual corpus regardless of whether fine-tuning was used. This demonstrates that the all-words WSD corpus was superior

Concept Embeddings	Avg. Ranking	Avg. Difference from First Place	Number of Leaf Nodes
All-words WSD vector	2.945	0.059	42
All-words WSD-fine vector	2.644	0.046	42
Manual vector	6.868	0.102	42
Manual-fine vector	3.143	0.049	42

Table 4: Evaluation by ranking measured by distance

Concept Embeddings	Avg. Ranking	Avg. Difference from First Place	Number of Leaf Nodes
All-words WSD vector	3.217	0.043	42
Manual vector	7.52	0.105	42

Table 5: Evaluation by ranking using distance with 10 iterations

to the manual corpus in generating concept embeddings. In other words, our experiments revealed that the corpus that was concept-tagged with 70% or 80% accuracy and whose size was approximately 69 times larger, was more suitable for generating concept. However, it cannot be claimed that when generating concept embeddings, the corpus size is more important than the accuracy of the concept tags of the corpus. Therefore, we conducted additional experiments to investigate the effect of the size of the all-words WSD corpus. Table 6 presents the average ranking of correct nodes, average difference from the concept in first place, and the number of leaf nodes according to the size of the all-words WSD corpus. We tested 10% to 100% of the size of the entire corpus in increments of 10%. This figure indicates that the average ranking monotonically improved from 10% to 60%, worsened at 70%, 80% and 90%, and achieved the best value when the entire corpus was used.

Finally, according to Table 4, we can observe the effect of order of the data used for training and retraining of word-embeddings. All-words WSD-fine vector and manual-fine vector use both the manual corpus and the all-words WSD corpus. The difference of two method is order of the data. It indicates that not only the size of the data but also the order of the data used for training and fine-tuning is important to improve the quality of word embeddings.

However, additional experiments are necessary to investigate the relationship between accuracy and corpus size.

For future work, other algorithm for word2vec, skip-gram can be tried instead of CBOW algorithm. Also, other word embeddings such as GloVe or fastText could be other options.

6 Conclusion

In this study, we generated concept embeddings using a concept-tagged corpus that was tagged by an all-words WSD system, and using fine-tuning. In addition, we evaluated the concept embeddings using rankings measured by the distances between the concept embeddings based on the tree structure of WLSP. We compared four concept embeddings: 1) concept embeddings that were trained with a concept-tagged corpus tagged by an all-words WSD system, 2) concept embeddings that were trained with a small and manually tagged corpus, 3) concept embeddings of 1) that were fine-tuned with a small and manually tagged corpus, and 4) concept embeddings of 2) that were fine-tuned with a cocept-tagged corpus tagged by an all-words WSD system. Experiments revealed that fine-tuning was effective in generating better concept embeddings when we utilized a small, manually tagged corpus and a corpus that was concept-tagged by an all-words WSD system. The all-words WSD-fine vector, which represented the concept embeddings initially trained with a large corpus automatically tagged by an all-words WSD system and fine-tuned with a small, manually tagged corpus, was superior when the concept embeddings were evaluated using the tree structure of WLSP.

Percentage of corpus used	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
All-words WSD vector	5.458	4.531	4.156	4.055	3.843	3.707	3.848	3.705	3.770	3.455
All-words WSD-fine vector	4.689	4.004	3.750	3.663	3.449	3.474	3.470	3.447	3.613	3.087
manual-fine vector	5.184	4.694	4.331	4.205	3.896	3.888	4.054	3.917	4.017	3.619

Table 6: Evaluation by ranking using distance according to the size of the all-words word sense disambiguation (WSD) corpus

Acknowledgments

This work was supported by JSPS KAKENHI Grants Number 18K11421, 17H00917, and a project of the Center for Corpus Development, NINJAL.

References

Philip Edmonds and Scott Cotton. 2001. Senseval-2: Overview. In *Proceedings of *SEMEVAL 2001*, pages 1—5.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of ACL 2016*, pages 897—-907.

Sachi Kato, Masayuki Asahara, and Makoto Yamazaki. 2018. Annotation of 'word list by semantic principles' labels for the balanced corpus of contemporary written Japanese. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong, 1–3 December. Association for Computational Linguistics.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written Japanese. *Language resources and evaluation*, 48(2):345–371.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of ICLR Workshop 2013*, pages 1–12.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS 2013*, pages 1–9.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL 2013*, pages 746–751.

National Institute for Japanese Language and Linguistics. 1964. *Word List by Semantic Principles*. Shuuei Shuppan, In Japanese.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of *SEMEVAL 2007*, pages 30–35.

Katsuyuki Ouchi, Hiroyuki Shinnou, Kanako Komiya, and Minoru Sasaki. 2016. Construction of word sense embeddings from word embeddings using synonyms. *Proceedings of NLP 2016 (in Japanese)*, pages 99–102.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017a. Neural sequence learning models for word sense disambiguation. In *Proceedings of EMNLP 2017*, pages 1156–1167.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017b. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of EACL 2017*, pages 99–110.

Hiroyuki Shinnou, Rui Suzuki, and Kanako Komiya. 2018. All-words wsd with wlsp number as s sense label using a bidirectional lstm. *Proceedings of the Language Resources Workshop 2018 (in Japanese)*, pages 2–4.

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In *Proceedings of *SEMEVAL 2004*, pages 41–43.

Rui Suzuki, Kanako Komiya, Masayuki Asahara, Minoru Sasaki, and Hiroyuki Shinnou. 2018. All-words word sense disambiguation using concept embeddings. *Proceedings of LREC 2018*, pages 1006–1011.

Shoma Yamaki, Hiroyuki Shinnou, Kanako Komiya, and Minoru Sasaki. 2017. Construction of word sense embeddings using training data. *Proceedings of NLP 2017 (in Japanese)*, pages 78–81.