Composing Word Vectors for Japanese Compound Words Using Bilingual Word Embeddings

メタデータ	言語: eng 出版者: 公開日: 2021-08-31 キーワード (Ja): キーワード (En): 作成者: メールアドレス: 所属:
URL	https://doi.org/10.15084/00003449

Composing Word Vectors for Japanese Compound Words Using Bilingual Word Embeddings

Teruo Hirabayashi Kanako Komiya Masayuki Asahara Hiroyuki Shinnou

Ibaraki University
4-12-1, Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan
{20nd303t, kanako.komiya.nlp, 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

This study conducted an experiment to compare the word embeddings of a compound word and a word in Japanese on the same vector space using bilingual word embeddings. Because Japanese does not have word delimiters between words; thus various word definitions exist according to dictionaries and corpora. We divided one corpus into words on the basis of two definitions, namely, shorter and ordinary words and longer compound words, and regarded two word-sequences as a parallel corpus of different languages. We then generated word embeddings from the corpora of these languages and mapped the vectors into the common space using monolingual mapping methods, a linear transformation matrix, and VecMap. We evaluated our methods by synonym ranking using a thesaurus. Furthermore, we conducted experiments of two comparative methods: (1) a method where the compound words were divided into words and the word embeddings were averaged and (2) a method where the word embeddings of the latter words are regarded as those of the compound words. The VecMap results with the supervised option outperformed that with the identical option, linear transformation matrix, and the latter word method, but could not beat the average method.

1 Introduction

Japanese words have many definitions because Japanese does not have word delimiters between words, and word boundaries are unspecific. Therefore, the Japanese dictionary defines words individually. Japanese has different word definitions according to each corpus and dictionary. The long unit for compound words and the short unit for words in UniDic[1] (Maekawa et al., 2010) developed by the National Institute for Japanese Language and Linguistics (NINJAL) are some of them. For example, "いちご狩り, ichigo-gari, strawberry picking" is defined as one word (short unit), whereas "ぶどう狩り, budou-gari, grape picking" is defined as a compound word (long unit) with two words (short unit)[2] in UniDic. Due to the limit of the dictionary's coverage, a morphological analyzer using UniDic treats " いちご狩り, ichigo-gari, strawberry picking" as one word and "ぶどう狩り, budou-gari, grape picking" as two words, making it impossible to directly compare the word meanings of these two words via word embeddings.

Therefore, to address the word unit discrepancy issue, this study proposes the usage of bilingual word embeddings (BWEs), which is usually used for mapping the word embeddings of two different languages into the same vector space, to map the word embeddings of long and short units into a common vector space. Using the BWE makes it easy to compare the word embeddings of "いちご狩り, ichigo-gari, strawberry picking" and "ぶどう狩り, budou-gari, grape picking" because both are on the same vector space. This situation is more convenient for many application systems like an information retrieval system.

[1] https://unidic.ninjal.ac.jp/ (In Japanese)

[2] いちご means strawberries; ぶどう means grapes; and 狩り means picking or hunting in Japanese.

2 Related Work

According to a survey of cross-lingual word embedding models[3], the BWE is classified into four groups according to how cross-lingual word embeddings are made.

The first approach is monolingual mapping. This approach initially trains monolingual word embeddings and learns a transformation matrix that maps representations in one language to those of the other language. Mikolov et al. (2013) showed that vector spaces can encode meaningful relations between words and that the geometric relations that hold between words are similar across languages. They did not assume the use of specific language; thus their method can be used to extend and refine dictionaries for any language pairs.

The second approach is pseudo-cross-lingual. This approach creates a pseudo-cross-lingual corpus by mixing contexts of different languages. Xiao and Guo (2014) proposed the first pseudo-cross-lingual method that utilized translation pairs. They first translated all words that appeared in the source language corpus into the target language using Wiktionary. They then filtered out the noises of these pairs and trained the model with this corpus, in which the pairs were replaced with placeholders to ensure that the translations of the same word have the same vector representation.

The third approach is cross-lingual training. This approach trains their embeddings on a parallel corpus and optimizes a cross-lingual constraint between the embeddings of different languages that encourages embeddings of similar words to be close to each other in a shared vector space. Hermann and Blunsom (2014) trained two models to output sentence embeddings for input sentences in two different languages. They retrained these models with sentence embeddings using a least squares method.

The final approach is joint optimization, which not only considers a cross-lingual constraint but also jointly optimizes monolingual and cross-lingual objectives. Klementiev et al. (2012) performed the first research using joint optimization. Zou et al. (2013) used a matrix factorization approach to learn cross-lingual word representations for English and Chinese and utilized the representations for a machine translation task. In this study, we used the first approach, monolingual mapping.

The nearest works to this research are those of Komiya et al. (2019) and Kouno and Komiya (2020). Komiya et al. (2019) composed word embeddings for long units from the two word embeddings of short units using a feed-forward neural network system. They classified the dependency relations of two short units into 13 groups and trained a composition model for each dependency relation. Meanwhile, Kouno and Komiya (2020) performed the multitask learning of the composition of word embeddings and the classification of dependency relations.

We utilized the BWE herein for the same purpose. To the best of our knowledge, our study is the first to use the BWE to map the word embeddings of different word delimitation definitions.

3 Methods

The BWE is usually used for cross-lingual applications (e.g., machine translation).

In this study, we mapped the word embeddings of short and long units into the common vector space for a comparison. short units are language units defined from the perspective of morphology (Ogura et al., 2007), whereas long units are those defined based on a Japanese base phrase unit, bunsetsu (Fujiike et al., 2008). A long unit consists of one or more short units. For the BWE, we utilized the linear transformation matrix and the VecMap [4].

3.1 Bilingual Word Embeddings

We used monolingual mapping comprising two steps. First, monolingual word embeddings were trained for each language. We regarded the corpora of different term units as the corpora of two different languages and mapped them to a common vector space such that the word embeddings of the words whose meanings were similar to each other in two languages can be brought closer. The geometrical relations that hold between words are similar across languages; thus a vector space of a language can be transformed into that of another language using a linear projection. We adapted hereikn two methods of the BWE, namely, linear transformation matrix and VecMap. A linear projection matrix W was

[3] https://ruder.io/cross-lingual-embeddings/

[4] https://github.com/artetxem/vecmap#publications

learned when we used a linear transformation matrix. VecMap is an implementation of a framework of Artetxe et al. (2017) to learn cross-lingual word embedding mappings (Artetxe et al., 2018a)(Artetxe et al., 2018b).

3.1.1 Linear Transformation Matrix

We conducted the following experiments when a linear transformation matrix was learned:

1. Generate short and long unit corpora and learn short or long unit embeddings for each corpus from them using word2vec (cf. Figure 1).

2. Learn a linear projection matrix W from the vector space of the short units to that of the long units using pairs of embeddings for common words generated in the last step.

3. Apply matrix W to the short unit embeddings and obtain the projected long unit embeddings for them.

3.1.2 VecMap

VecMap was used as another method of the BWE. We projected the vector space of the short units into that of the long units when we used the linear transformation matrix. However, VecMap projected both the vector spaces of the short and long units into a new vector space. The two options (i.e., supervised and identical) were compared. The supervised VecMap uses the specified words, whereas the identical VecMap uses identical words in two languages as the projection seeds. Therefore, the seed words of the supervised VecMap were the same as the linear transformation matrix but those of the identical VecMap were different.

4 Experiments

We used NWJC2vec (Shinnou et al., 2017) for the word embeddings of the short units and the Balanced Corpus of Contemporary Japanese (BCCWJ) (Maekawa et al., 2014) for the word embeddings of the long units using word2vec.

4.1 Word Embeddings

NWJC2vec is a set of word embeddings generated from the 25 billion word scale NWJC-2014-4Q dataset (Asahara et al., 2014), which is an enormous Japanese corpus, NINJAL Web Japanese Corpus (NWJC), developed using the word2vec tool. The summary statistics for the NWJC-2014-4Q data and the parameters used to generate the word embeddings are respectively presented in Tables 1 and 2. We used continuous bag-of-words (CBOW) as a model architecture to produce the word embeddings.

BCCWJ is the 100 million word scale balanced corpus that contains texts from multiple domains constructed by NINJAL. Each text in this corpus has short ande long unit versions. The summary statistics for BCCWJ are listed in Table 3. The word2vec settings for training the word embeddings with BCCWJ are summarized in Table 4.

NWJC2vec contains morphological information, but the word embeddings generated for the long units using BCCWJ do not contain them. Therefore, the word embeddings for the short units can be differentiated from the words with the same spellings but are different parts of speech, whereas those for the long units cannot. Consequently, for some words, the word embeddings for the short units of some words had multiple vectors, but we still directly used them.

4.2 Bilingual Word Embeddings

The learning parameters of the linear transformation matrix are shown in Table 5. We used a 200-by-200 dimensional linear transformation matrix. We used Adam as the optimizer of loss function and iterated the training for 1,164 epochs. We decided on the number of epochs according to the preliminary experiments using 55,630 words randomly extracted from the training data. We averaged the best number of five trials. The vocabulary size of the word embeddings for BCCWJ and NWJC and the seed words we used for the linear transformation matrix is shown in Table 6.

We used the default settings for the VecMap tool for each option. The default settings of the parameters of each specific option and their general default settings are listed in Table 7. The vocabulary size of the word embeddings for BCCWJ and NWJC and the seed words used for VecMap is presented in Table 8.

The number of long units decreased for VecMap compared with the linear transformation matrix be-

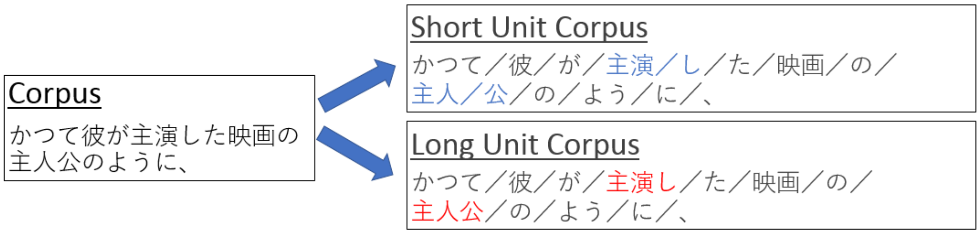

Figure 1: Short and long unit corpora

Number of URLs collected	83,992,556
Number of sentence	1,463,142,939
Number of words (tokens)	25,836,947,421

Table 1: Summary statistics for the NWJC-2014-4Q dataset

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

Table 2: Parameters used to generate NWJC2vec

Number of text samples	172,675
Number of short units (tokens)	104,911,464
Number of long units (tokens)	83,585,665

Table 3: Summary statistics for the Balanced Corpus of Contemporary Japanese (BCCWJ)

Parameters	Options	Settings
CBOW or skip-gram	-cbow	1
Dimensionality	-size	200
Window size	-window	5
Number of iterations	-iter	5
Batch size	-batch$_{words}$	1,000
Minimum count of words	-min-count	1

Table 4: Settings of word2vec

Parameters	Settings
Dimensionality	200 × 200
Optimization algorithm	Adam
Number of epochs	1,164

Table 5: Learning parameters of the linear transformation matrix

cause of the limitation of the machine power. We used 278,143 seed words and 11,662 compound words annotated with a concept number for the evaluation, which resulted to a total of 289,805 words.

5 Evaluation

We evaluated our methods by the ranking of synonyms using a thesaurus. Using a thesaurus, we can evaluate the similarity of concepts referring knowledge of people. However, if we directly use cosine similarity between concepts, the thresholds are difficult to decide. Therefore, we used the ranking among the nodes of the thesaurus. We used "Word List by Semantic Principles" (WLSP) (National Institute for Japanese Language and Linguistics, 1964) [5] as a thesaurus. The WLSP is a Japanese thesaurus that classifies and orders a word according to its meaning. One record is composed of the following elements: record ID number, lemma number, type of record, 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, a medium item, and a classification item. The tree structure of the WLSP is shown in Figure 2.

The WLSP has a tree structure; thus, we assumed that the concepts belonging to the same node or synonyms were similar to each other.

[5] https://pj.ninjal.ac.jp/corpus_center/goihyo.html

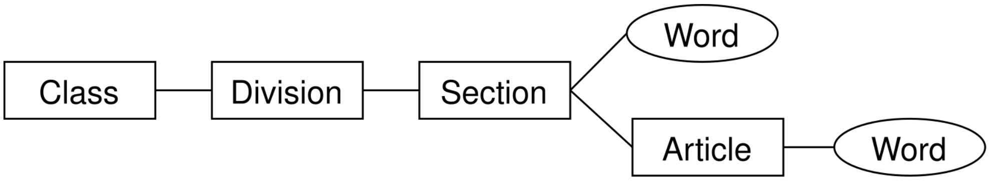

Figure 2: Tree structure of the Word List by Semantic Principles (WLSP)

Corpus	Vocabulary size (Number of word tokens)
BCCWJ (long unit)	2,745,657
NWJC2vec (short unit)	1,534,957
Seed words	278,143

Table 6: Vocabulary size (number of word tokens) of the word embeddings for BCCWJ and NWJC and seed words for the linear transformation matrix

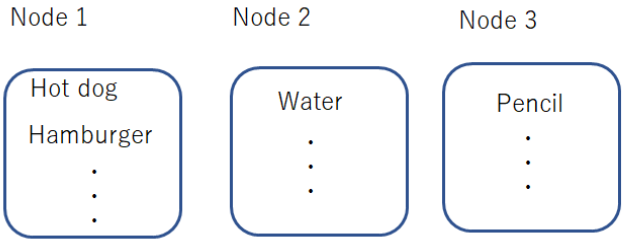

Figure 3: Example of the nodes of the WLSP

An example of the WLSP nodes is presented in Figure 3. In this figure, we assumed that *hot dog* was closer to *hamburger* than *water* or *pencil*. We used *hot dog* instead of long term like “葡萄狩り, grape picking” and *hamburger* instead of short term like “いちご狩り, strawberry picking” for example. We used *water* and *pencil* as short terms in this example.

We evaluated the mapped word embeddings on the basis of this assumption and subsequently defined “long term” and “short term.” A compound word that is a long unit and consists of two short units is referred to as “long term,” whereas a word that is a short unit with no long unit is referred to as “short term.”

5.1 Evaluation Procedure

All the NWJC or BCCWJ words were not listed on the WLSP; thus we had two compound word conditions for evaluation: (1) the compound word should be a long term listed on the WLSP, and (2) its constituents of it should be short terms listed on the WLSP. Hereinafter, wl_i denotes the compound word, and ws_{i_1} and ws_{i_2} denote the constituents. The evaluation procedures are as follows:

1. For each long term wl_i, identify a node $N_i(0)$ to which the long term belongs in WLSP.

   $N_i(0)$ includes synonyms of wl_i and both long and short terms. We assumed that every node has at least two words such that the similarity between them can be calculated. For example, if wl_i is the word *hot dog*, the corresponding node $N_i(0)$ includes synonyms such as *hamburger*. In Figure 3, $N_i(0)$ is Node 1.

2. Calculate $s_i(0)$, which is the average similarity between the word embeddings of wl_i and all the short terms in $N_i(0)$, using the mapped word embeddings.

   For this step, we calculated $s_i(0)$, which is the average similarity between the word embeddings of *hot dog* and those of *hamburger* and other concepts in $N_i(0)$ (Node 1). We used the cosine similarity for the similarity and the arithmetic mean to average the similarity.

3. Obtain sibling nodes $N_i(1)...N_i(n)$ of $N_i(0)$.

   A sibling nodes $N_i(1)...N_i(n)$ include a node that contains a word, such as *water*, and another node that contains a word such as *pencil*. In Figure 3, $N_i(1)...N_i(n)$ includes Nodes 2 and 3.

4. Similarly, calculate $s_i(k)$, which is the average similarity between the word embeddings of wl_i and those of all the short terms in node $N_i(k)$

5. Obtain the ranking of $s_i(0)$ in $s_i(0)...s_i(n)$.

We used 11,459 long terms for the evaluation because 11,662 long terms and their constituent short

Option	Parameter	Default setting of specific option	General sefault setting
Supervised	Batch size	1,000	10,000
Identical	Self-learning	TRUE	FALSE
Identical	Vocabulary_cutoff	200,000	0
Identical	csls_neibourhood	10	0

Table 7: Parameters of VecMap

Corpus	Vocabulary size
BCCWJ (long unit)	289,805
NWJC2vec (short unit)	1,534,957
Seed words	278,143

Table 8: Vocabulary size of word embeddings for BCCWJ and NWJC and seed words for VecMap

terms were annotated with a concept number, but 203 of them had un-annotated synonyms in the node to which the word belongs ($N_i(0)$). The number of nodes we used was 881 after excluding 14 nodes that included a word with no word embeddings.

We performed two comparative methods, namely, average and latter word methods. For the average method, the word embeddings of a long term were calculated as the average of its constituent short terms, that is, the average of the word embeddings of ws_{i_1} and ws_{i_2} was used. For the latter word method, the word embeddings of the latter short term were regarded as the word embeddings of the long term, that is, the word embeddings of ws_{i_2} were used.

5.2 Results and Discussion

The average rankings of the correct node according to each method are shown in Table 9.

Method	Ranking
Linear transformation matrix	187.50
VecMap (supervised)	131.98
VecMap (identical)	330.40
Average	80.41
Latter word	143.16

Table 9: Average rankings of the correct node according to method

Table 9 shows that the best method among the three proposed methods is VecMap with the supervised option. The ranking of the correct node when the method was used was 131.98th. The number of nodes we used was 881; thus, if the node is randomly selected, the ranking would be 440th or 441st. Therefore, VecMap outperformed the random baseline and the latter word method (Table 9). However, the average method known as the strong comparative method was the best among all the methods tested. BWEs could not beat it. This result indicates that the additive compositionality holds for many long units. For future work, Skipgram can be tried instead of CBOW algorithm. Also, other word embeddings such as Glove could be another option. Theoretically, we believe that our methods can be applied even if the dimensionalities of two embeddings are different,but should be tested to know the real results.

6 Conclusion

In this study, we mapped word the embeddings of a compound word and word in Japanese into the same vector space using the BWE. We used the linear transformation matrix and VecMap as the BWE methods. VecMap with the supervised option outperformed one baseline, which was the method where the word embeddings of the latter constituent word are regarded as the word embeddings of the compound word but could not beat another baseline, which was the method where the average of the word embeddings of the constituents was used for the word embeddings of the compound word.

Acknowledgments

This work was supported by JSPS KAKENHI Grants Number 18K11421, 17KK0002, and a project of the Younger Researchers Grants from Ibaraki University.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798.

Masayuki Asahara, Kikuo Maekawa, Mizuho Imada, Sachi Kato, and Hikari Konishi. 2014. Archiving and analysing techniques of the ultra-large-scale web-based corpus project of ninjal, japan. *Alexandria*, 25(1-2):129–148.

Yumi Fujiike, Hideki Ogura, Toshinobu Ogiso, Hanae Koiso, Kiyotaka Uchimoto, Satsuki Soma, and Takenori Nakamura. 2008. Short-term unit alanysis of balanced corpus of contemporary japanese. In *Proceedings of the NLP2008, (In Japanese)*, pages 931–934.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. *arXiv preprint arXiv:1404.4641*.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. *Proceedings of COLING 2012*, pages 1459–1474.

Kanako Komiya, Takumi Seitou, Minoru Sasaki, and Hiroyuki Shinnou. 2019. Composing word vectors for japanese compound words using dependency relations. *CICLING*. no 229.

Shinji Kouno and Kanako Komiya. 2020. Composition of word representation of long-term units from word representations of short-term units using multitask learning. In *Proceedings of the NLP2020, (In Japanese)*, pages 209–212.

Kikuo Maekawa, Makoto Yamazaki, Takehiko Maruyama, Masaya Yamaguchi, Hideki Ogura, Wakako Kashino, Toshinobu Ogiso, Hanae Koiso, and Yasuharu Den. 2010. Design, Compilation, and Preliminary Analyses of Balanced Corpus of Contemporary Written Japanese. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pages 1483–1486.

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, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

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

Hideki Ogura, Toshinobu Ogiso, Hanae Koiso, Yumi Fujiike, and Satsuki Soma. 2007. Short-term unit alanysis of balanced corpus of contemporary japanese. In *Proceedings of the NLP2007, (In Japanese)*, pages 720–723.

Hiroyuki Shinnou, Masayuki Asahara, Kanako Komiya, and Minoru Sasaki. 2017. nwjc2vec: Nwjc2vec: Word embedding data constructed from ninjal web japanese corpus. *Journal of Natural Language Processing (In Japanese)*, 24(5):705–720.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 119–129.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398.