# Deep Case Identification Using Word Embedding

Takumi Kawasaki and Masaomi Kimura

***Abstract*—Computer-based text analyses have been widely applied to the Japanese language. However, the application is complicated because a word order is flexible in Japanese sentences; thus, the role of nouns cannot be determined only by sentence structure. Such roles are referred to deep cases that represent the essential meaning of nouns, which is important in the semantic analysis of sentences. Identifying deep cases of nouns can improve text analysis results. Since surrounding words are important to determine these roles, their identification should have high affinity to word embedding, which has gained popularity in natural language processing. In word embedding, similar role words have similar vectors with high cosine similarity. Therefore, in this paper, we propose a semantic analysis method using word embedding.**

***Index Terms*—Deep cases, semantic role labeling, semantic analysis.**

## I. Introduction

Many computer-based text mining methods have been proposed. However, the application of text analysis to the Japanese language is complicated because Japanese grammar is more variable and less restrictive relative to word order than English. Such variability can cause incorrect text analysis results. The largest problem of computer-based analysis applied to Japanese text data is that the roles of words cannot be determined only by a sentence structure (word position). Such roles are referred to as deep cases, which represent the essential meanings of nouns and play an important role in the semantic analysis of sentences. For example, in the sentence “私は学校へ行く” (I go to school), the deep cases for “私” (I) and “学校” (school) are agent and goal respectively. Deep case identification, which has been applied in machine translation and question answer systems, is important to understand the semantic information contained in sentences.

Previous studies have used syntactic and dependency features to identify deep cases. For example, Ide *et al.* [1] proposed a method to assign deep cases to nouns by identifying deep cases using noun and verb classes. In that method, in order to define noun and verb classes properly, they are added to the dictionary corresponding to each class prior to identification.

Recently, several studies have used word embedding and artificial neural networks (ANNs) to identify deep cases. Word embedding maps words to vectors using various models and algorithms such as Word2Vec [2], GloVe [3], and fastText [4]. The vectors are calculated to reflect the meanings of words, i.e., words with similar meanings have similar vectors. The vectors are learned using the words surrounding the target words. This suggests that word embedding is suitable for identifying deep cases because similar nouns tend to have the same deep case. Thus, in this study, we propose a deep case identification method using word embedding.

## II. Corresponding Techniques

In this section, we review the Japanese natural language processing techniques used in our study according to the example Japanese sentence, “専門家は危険に気付いており、それを回避すべき。” (The experts have noticed the danger, and they should avoid it.).

### A. Morphological Analysis

The Japanese language is agglutinative; thus, the words in a Japanese sentence are not separated by spaces. Therefore, morphological analysis is required to extract words from a sentence and predict a given word’s part of speech. Morphological analysis divides a sentence into morphemes, i.e., the smallest grammatical unit of meanings. In this study, we used the MeCab [5] Japanese morphological analysis tool. The MeCab output for the example sentence is shown in Table I.

Note that Japanese particles follow other words such as nouns, verbs, and adjectives, and in a given sentence, particles, which can be compared to English prepositions, indicate the relationship of a preceding noun to a verb (referred to as a surface case). A surface case corresponds to a particle in a sentence and specifies a surface semantic relationship.

TABLE I: Example of Morphological Analysis Outputs

| Word | Part of Speech | Meaning |
|---|---|---|
| 専門家 | Noun | experts |
| は | Particle | (N/A) |
| 危険 | Noun | danger |
| に | Particle | to |
| 気付い | Verb | noticed |
| て | Particle | (N/A) |
| おり | Verb | have |
| それ | Pronoun | it |
| を | Particle | (N/A) |

### B. Related Work

Many semantic role labeling methods have been proposed [1], [6]-[9]. Ide *et al.* [1] proposed a method to assign deep

Manuscript received October 5, 2018; revised; December 8, 2018.

Takumi Kawasaki is with the Electrical Engineering and Computer Science Department, Shibaura Institute of Technology, Tokyo Japan (e-mail: ma17031@shibaura-it.ac.jp).

Masaomi Kimura is with the Information Science and Engineering Department, Shibaura Institute of Technology, Tokyo Japan (e-mail: masaomi@shibaura-it.ac.jp).

DOI: 10.7763/IJCTE.2018.V10.1229

cases to nouns using an ANN whose inputs were vectors associated with a combination of a noun, a verb, and a particle. Initially, Ide *et al*. defined seven new deep cases. A deep case shows the meaning and position of a target noun relative to its modifying verb; however just superficial information, i.e., the collocation relationships of nouns and their corresponding particles, can give us only surface case assignment. When a combination of a noun, verb, and particle is given, their proposed method converts the noun and verb into pre-classified classes. A one-hot vector is then assigned to each class and each particle for nouns and verbs. Then, the method utilizes an ANN whose input is the concatenated one-hot vectors corresponding to the input combination. The ANN outputs the degree of noun applicability to each of the seven deep cases. The noun is then assigned to the case with the highest scale. However, with this method, no ANN input can be set if the nouns and verbs are not registered in the noun and verb classes.

Recently, several studies have employed word embedding and ANNs [10]-[13]. For example, Okamura *et al*. [10] developed deep case identification models that use ANNs whose input is a bag-of-words (BOW) and word embedding. In this study, we used word embedding rather than a BOW.

TABLE II: DEEP CASES DEFINED IN PREVIOUS STUDY

| No. | Deep Case Name | Feature |
|---|---|---|
| 1 | Subject | Entity that causes some behavior or state change |
| 2 | Before change | State before a certain operation or state change occurs |
| 3 | Object | Those affected by certain actions and state changes |
| 4 | Situation | Status when a certain action or state change occurs |
| 5 | Point of attachment | End point in a certain operation or state change |
| 6 | Destination | Influence on certain operations and state changes |
| 7 | Relation | Relationship with objects where certain actions or state changes occur |

## III. METHODOLOGY

### A. Learning Word Embedding

We performed learning to obtain word embedding using Word2Vec, which has two learning methods, i.e., Skip-Gram [14] and Continuous BOW (CBOW) [14]. Note that we used both methods in this study. We used a Japanese Wikipedia dump [15] as training data and the WikiExtractor tool [16] to extract plain text from the dump. In addition, we used MeCab [5] to divide sentences in the Wikipedia text data into words. The Word2Vec parameters are shown in Table III.

TABLE III: WORD2VEC PARAMETERS

| | | |
|---|---|---|
| -size | Dimensionality of the feature vectors | 250 |
| -window | Maximum distance between the current and predicted word within a sentence | 10 |
| -min_count | Ignores all words with a lower total frequency | 20 |
| -iter | Number of iterations (epochs) over the corpus | 5 |

### B. Creating a Model to Estimate Deep Cases

We created a model to estimate deep cases as shown in Fig. 1. Here, the inputs to the ANN were word embedding of nouns and verbs, as well as a one-hot vector corresponding to particles. We set the one-hot vectors of the deep cases shown in Table II as the output layer. The ANN parameters are shown in Table IV.

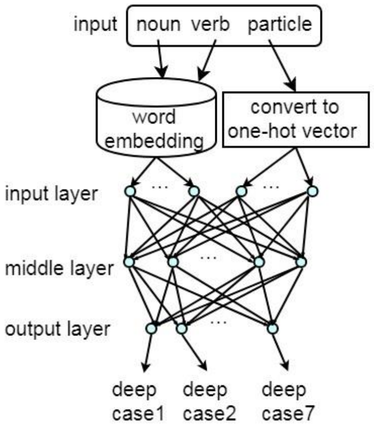
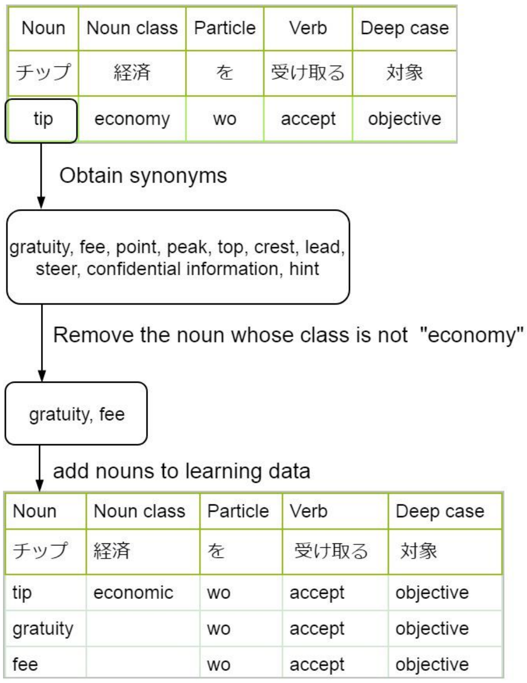


Fig. 1. Deep case estimation model.

TABLE IV: DEEP CASES ESTIMATION MODEL PARAMETERS

| | |
|---|---|
| Middle layer num | 350 |
| Activation function for middle layer | ReLU |
| Activation function for output layer | softmax |
| Loss function | Log loss |
| Optimization algorithm | SGD |
| Epoch | 100 |
| Batch size | 50 |

### C. Addition of Training Data



| Noun | Noun class | Particle | Verb | Deep case |
|---|---|---|---|---|
| チップ | 経済 | を | 受け取る | 対象 |
| tip | economy | wo | accept | objective |

Obtain synonyms
gratuity, fee, point, peak, top, crest, lead, steer, confidential information, hint
Remove the noun whose class is not "economy"
gratuity, fee
add nouns to learning data

| Noun | Noun class | Particle | Verb | Deep case |
|---|---|---|---|---|
| チップ | 経済 | を | 受け取る | 対象 |
| tip | economic | wo | accept | objective |
| gratuity | | wo | accept | objective |
| fee | | wo | accept | objective |

Fig. 2. The flow of training data addition.

We used a thesaurus [17] to create the model described in Section III-B. Note that the training data number is biased in each deep case. The accuracy of deep cases with less training data is low; thus, we added data as follows. Here, we used a classification vocabulary table [18] as noun classes. An example is shown in Fig. 2, where we translated Japanese words to English words.

1. Obtain synonyms of the nouns included in the training data using Japanese WordNet [19].
2. Remove nouns that are not of the same noun class as the nouns in the training data from the obtained synonyms.
3. Add synonyms that were not removed to training data.

In Japanese WordNet, some words are related to a number of synsets (concepts). The synsets can include nouns different meaning from the nouns in the training data; therefore, we included Step 2. Using this method to add training data, we obtained 104445 records in total from 10850 records. Here, we used 82170 records whose word embedding could be obtained to train the deep case identification models.

## IV. Experiments and results

We used the thesaurus [17] for our experiments. It includes combinations of a noun, a particle, a verb, and a deep case. We converted the deep cases into the ones defined by Ide *et al*. and used them to train models.

### A. Effect of Increasing Training Data

The results of the analysis obtained by the model discussed in Section III-A are shown in Table V. As can be seen, precision was improved by increasing the volume of training data except for the situation case. To clarify the cause of the low precision in the situation case, we counted the number of situation case data assigned to each of other cases. The results are shown in Table VI. Most of incorrect assignments of the situation case were assigned to the object case or the point of attachment case. In addition, the numbers of data assigned to the object and point of attachment cases were counted for each particle. The results are shown in Tables VII and VIII. The data whose particle is “が ” (ga) are most frequently assigned incorrectly to the object case. The data whose particle is “に ” (ni) are most frequently assigned incorrectly to the point of attachment case. The object case primarily comprises “が” (ga), and the point of attachment case is dominated by “に” (ni) . Moreover, the training data number of the situation case is less than that of the object and point of attachment cases. From this information, it is reasonable to consider that this was because weights to particles in the ANN increased. Moreover, the situation case tended to be judged as object case or point of attachment case because the situation case data was less than that of the object case or the point of attachment case.

Note that the Skip-Gram analysis demonstrated better results than the CBOW analysis. This suggests that Skip-Gram made the vectors of nouns in the same category closer than CBOW.

### B. Variations of ANN Inputs

We created models using different inputs to the ANN (Section IV-A). The different inputs are shown in Table IX. Note that, except for the number of middle layer dimensions, the parameters were unchanged. The dimension of the input layer was changed according to the number of features; thus, the number of middle layer dimensions was set to the sum of the number of input layers and output layers (number of input layers + number of output layers) ×2/3).

TABLE V: Results of Deep Case Identification Model 1

| | Subject | Before change | Object | Situation | Point of attachment | Destination | Relation | sum |
|---|---|---|---|---|---|---|---|---|
| Training data before addition | 3127 | 252 | 5440 | 212 | 1427 | 77 | 315 | 10850 |
| Test data number | 2521 | 268 | 4752 | 188 | 1249 | 86 | 243 | 9307 |
| Correct number (Skip-Gram) | 2389 | 159 | 4456 | 74 | 1023 | 46 | 134 | 8281 |
| Precision (Skip-Gram) | 0.948 | 0.593 | 0.938 | 0.394 | 0.819 | 0.535 | 0.551 | 0.8605 |
| Training data after addition | 21709 | 2144 | 63204 | 1676 | 11698 | 1100 | 2914 | 104445 |
| Used training data | 15657 | 1787 | 50522 | 1276 | 9714 | 990 | 2224 | 82170 |
| Correct number (Skip-Gram) | 2410 | 242 | 4460 | 69 | 1098 | 74 | 182 | 8535 |
| Precision (Skip-Gram) | 0.956 | 0.903 | 0.939 | 0.367 | 0.879 | 0.86 | 0.749 | 0.9171 |
| Correct number (CBOW) | 2383 | 193 | 4417 | 72 | 1037 | 58 | 151 | 8311 |
| Precision (CBOW) | 0.945 | 0.72 | 0.93 | 0.383 | 0.83 | 0.674 | 0.621 | 0.893 |

TABLE VI: Data of Correct Situation Case

| correct/result | Subject | Before change | Object | Situation | Point of attachment | Destination | Relation |
|---|---|---|---|---|---|---|---|
| Situation case | 0 | 4 | 34 | 69 | 75 | 3 | 3 |

TABLE VII: Particle Number Assigned to Object Case

| Particle | を (wo) | に (ni) | へ (he) | は (ha) |
|---|---|---|---|---|
| Incorrect num | 25 | 7 | 1 | 1 |

TABLE VIII: Particle Number Assigned to Point of Attachment Case

| Particle | に (ni) | で (de) |
|---|---|---|
| Incorrect num | 71 | 4 |

Model 1 is explained in Section IV-A. We trained Model 2 without particle information because the particle weight was too large in Model 1. We trained Model 3 using only particle information as the input to confirm that the analysis using Model 1 was not work only by particle information. In Model 4, we attempted analysis using word embedding corresponding to particles because we created word embedding not to remove stop words and obtain word embedding corresponding to particles. We used the sum of word embedding corresponding to a noun, a verb and a particle to train the ANN.

TABLE IX: ANN INPUT

| Model name | Feature amount | Dimension |
|---|---|---|
| 1 | Word embedding corresponding to nouns and verbs + one-hot vector corresponding to particles | (word embedding dimension) *2+(particle num) |
| 2 | Word embedding corresponding to nouns and verbs | (word embedding dimension) *2 |
| 3 | One-hot vector corresponding to particles | particle num |
| 4 | Word embedding corresponding to nouns, verbs, and particles | (word embedding dimension) *3 |
| 5 | Sum of word embedding corresponding to nouns, verbs, and particles | word embedding dimension |

The results of each model are shown in Table X. As can be seen, the precision of Model 2 was less than that of Model 1. Ide *et al*. defined new deep cases by clustering original deep cases based on the co-occurrence frequency of deep cases and particles; therefore, particle information should be very important relative to the identification of deep cases. The precision of Model 3 was also less than that of Model 1. This demonstrates that word embedding corresponding to nouns and verbs is an effective feature to assign deep cases to nouns. The precision of Model 4 was nearly the same as that of Model 2. This demonstrates that a word embedding corresponding to particles does not include semantic information and that Model 4 worked to use word embedding corresponding to particles to just judge which particle is input. The results of Model 5 were nearly the same as those of Models 1 and 4. This indicates that the sum of a word embedding corresponding to a noun, a verb, and a particle holds the information of the three original word embeddings. Word embedding kept words' information in different word embedding dimensions for each part of speech, and the domains of vectors for different parts of speech did not overlap and the sum of word embedding vectors held the original information.

## V. CONCLUSION

In this paper, we have proposed a deep case identification method that used Word2Vec. In our model, the accuracy rate of each deep case exceeded 70% except for situation case and the accuracy of all data was 90%.

We observed that differences occur in the results depending on the input to the models. A model that used word embedding for nouns, verbs, and particles demonstrated results that were nearly the same as a model that replaced particles with one-hot vectors and a model that used the sum of vectors obtained as word embedding corresponding to nouns, verbs, and particles. This demonstrates that the included vectors hold the original vector's information.

In the future, we are planning to improve the precision of the situation case. In our experiments, we used formatted data with only a single noun in each sentence. However, common sentences can contain plural nouns and compound nouns; therefore, we are planning to study how to apply the proposed method to common text data. For example, it may be possible to represent compound nouns by the sum of vectors corresponding to noun word embeddings.

TABLE X: RESULT OF DEEP CASE IDENTIFICATION MODELS

| Model name | Subject | Before change | Object | Situation | Point of attachment | Destination | Relation | Sum |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.956 | 0.903 | 0.939 | 0.367 | 0.879 | 0.86 | 0.749 | 0.917 |
| 2 | 0.745 | 0.403 | 0.847 | 0.266 | 0.515 | 0.384 | 0.358 | 0.733 |
| 3 | 0.997 | 0.826 | 0.650 | 0.000 | 0.977 | 0.968 | 0.543 | 0.783 |
| 4 | 0.960 | 0.922 | 0.938 | 0.388 | 0.882 | 0.872 | 0.749 | 0.919 |
| 5 | 0.949 | 0.884 | 0.939 | 0.362 | 0.873 | 0.849 | 0.691 | 0.912 |

## REFERENCES

[1] D. Ide and M. Kimura, "The method to identify the deep cases based on relationships between nouns, particles, and verbs (2nd report)," in *Proc. the 32nd Fuzzy System Symposium*, 2016, pp. 695–700.

[2] Word2vec. [Online]. Available: https://code.google.com/archive/p/word2vec/

[3] GloVe: Global Vectors for Word Representation. [Online]. Available: https://nlp.stanford.edu/projects/glove/

[4] FastText. [Online]. Available: https://fasttext.cc/

[5] MeCab: Yet Another Part-of-Speech and Mo-rphological Analyzer. [Online]. Available: http://taku910.github.io/mecab/

[6] Y. Ishihara and K. Takeushi, "Construction of Japanese semantic role labeling system using hierarchical tag context trees extracted from tail expressions of dependency elements," *Transactions of Information Processing Society of Japan*, vol. 57, no. 7, pp. 1611-1626, 2016.

[7] M. Harada and T. Mizuno, "Japanese semantic analysis system SAGE using EDR," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 16, no. 1, pp. 85-93, 2001.

[8] H. Ouchi, H. Shindo, and Y. Matsumoto, "Span selection model for semantic role labeling," *The Special Interest Group Technical Reports of IPSJ*, vol. 236, no. 9, pp. 1-13, 2018

[9] K. Takeuchi, S. Tsuchiyama, M. Moriya, and Y. Moriyasu, "Construction of argument structure analyzer toward searching same situations and actions," *IEICE Technical Report*, vol. 109, no. 390, pp. 1-6, 2010.

[10] T. Okamura, K. Takeuchi, and Y. Ishihara, "Construction of Japanese semantic role labeling system using neural network," in *Proc. the 24th Annual Conference of The Association for Natural Language Processing*, 2018, pp. 105-108.

[11] L. He, K. Lee, M. Lewis, and L. Zettlemoyer, "Deep semantic role labeling: What works and what's next," in *Proc. the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, vol. 1, pp. 473-483.

[12] D. Marcheggiani, A. Frolov, and I. Titov, "A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling," in *Proc. CoNLL*, 2017, pp. 411-420.

[13] J. Zhou and W. Xu, "End-to-end learning of semantic role labeling using recurrent neural networks," in *Proc. the 53rd Annual Meeting of*

*the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, vol. 1, pp. 1127-1137.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *ICLR*, 2013.

[15] Index of /jawiki/. [Online]. Available: https://dumps.wikimedia.org/jawiki/

[16] Wikiextractor. [Online]. Available: https://github.com/attardi/wikiextractor

[17] K. Takeuchi, S. Tsuchiyama, M. Moriya, Y. Moriyasu, and K. Satoh, "Verb sense disambiguation based on thesaurus of predicate-argument structure," in *Proc. International Conference on Knowledge Engineering and Ontology Development*, 2011, pp. 208-213.

[18] Classification vocabulary table – revised version database. [Online]. Available: http://pj.ninjal.ac.jp/corpus_center/goihyo.html

[19] F. Bond, T. Baldwin, R. Fothergill, and K. Uchimoto, "Japanese SemCor: A sense-tagged corpus of Japanese," presented in the 6th International Conference of the Global WordNet Association, 2012.

**Takumi Kawasaki** was born in Japan in 1994. He received the B.E. degree in information science and engineering from Shibaura Institute of Technology, Tokyo, Japan in 2017. In 2017, he joined the Department of Electrical Engineering and Computer Science, Shibaura Institute of Technology, as a M.E. degree.

**Masaomi Kimura** received the Sc.D degree in physics from the University of Tokyo, Japan in 1999. From 1999 to 2004, he worked for IBM as an IT specialist. From 2004, he was a lecturer, an associated professor from 2007, a professor from 2013 in Shibaura Institute of Technology, Japan.