Acquiring distributed representations for verb-object pairs by using word2vec

Miki Iwai*1, Takashi Ninomiya*2, Kyo Kageura*3

Graduate School of Interdisciplinary Information Studies, The University of Tokyo*1
Graduate School of Science and Engineering, Ehime University*2
Graduate School of Education, The University of Tokyo*3
1156553643@mail.ecc.u-tokyo.ac.jp, ninomiya@cs.ehime-u.ac.jp, kyo@p.u-tokyo.ac.jp

Abstract

We propose three methods for obtaining distributed representations for verb-object pairs in predicated argument structures by using word2vec. Word2vec is a method for acquiring distributed representations for a word by retrieving a weight matrix in neural networks. First, we analyze a large amount of text with an HPSG parser; then, we obtain distributed representations for the verb-object pairs by learning neural networks from the analyzed text. We evaluated our methods by measuring the MRR score for verb-object pairs and the Spearman's rank correlation coefficient for verb-object pairs in experiments.

1 Introduction

Natural language processing (NLP) based on corpora has become more common thanks to the improving performance of computers and development of various corpora. In corpus-based NLP, word representations and language statistics are automatically extracted from large amounts of text in order to learn models for specific NLP tasks. Complex representations of words or phrases can be expected to yield a precise model, but the data sparseness problem makes it difficult to learn good models with them; complex representations tend not to appear or appear only a few times in large corpora. For example, the models of statistical machine translation are learned from various statistical information in monolingual corpora or bilingual corpora. However, low-frequency word representations are not learned well, and consequently, they are processed as unknown words, which causes mistranslations. It is necessary not only to process NLP tasks by matching surface forms but to generalize the language representations into semantic representations.

Many approaches represent words with vector space models so that texts can be analyzed using semantic representations for individual words or multi-word expressions. These methods can be classified into two approaches: the word occurrence approach and the word co-occurrence approach. The word occurrence approach includes Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997), Probabilistic LSA (PLSA) (Hofman, 1999) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003), which acquire word representations from the distributions of word frequencies in individual documents (a word-document matrix). Recently, many researchers have taken an interest in the word co-occurrence approach, including distributional representations and neural network language models (Mikolov et al., 2013a; Mikolov et al., 2013b; Mnih and Kavukcuoglu, 2013; Pennington et al., 2014). The word co-occurrence approach uses statistics of the context around a word. For example, the distributional representations for a word are defined as a vector that represents a distribution of words (word frequencies) in a fixed-size window around the word. The neural network language models, including word2vec (Mikolov et al., 2013a; Mikolov et al., 2013b), GloVe (Pennington et al., 2014) and vector Log-Bilinear Language model (vLBL) (Mnih and Kavukcuoglu, 2013), generate distributed representations, which are dense and low-dimensional vectors representing word meanings, by learning a neural network that solves a pseudo-task of predict-

29th Pacific Asia Conference on Language, Information and Computation: Posters, pages 328 - 336
Shanghai, China, October 30 - November 1, 2015

ing a word given its surrounding words. Word2vec is preferred in NLP because it learns distributed representations very efficiently. Neural network language models have semantic compositionality for word-word relations by calculating vector representations; e.g., 'king' - 'man' + 'woman' is close to 'queen.' However, they acquire the distributed representations for a word, not phrase structures such as verb and object pairs. It is necessary to obtain representations for phrases or sentences to be used as natural language representations.

We devised three methods for acquiring distributed representations for verb-object pairs by using word2vec. We experimentally verified that the distributed representations of different verb and object pairs have the same meaning. We focused on verb-object pairs consisting of verbs whose meaning is vague, such as light-verbs, e.g., the 'do' and 'dishes' pair in "do dishes". The following two sentences are examples that have similar meanings but whose phrase structures are different.

1. I wash the dishes.
2. I do the dishes.

The representations for the verb-object pairs in the first sentence is "wash(dishes)," and those for the second sentence is "do(dishes)" with the light verb 'do'. Despite the difference between the representations of these sentences, they have the same meaning "I wash the dishes." As such, there are various sentences that have the same meaning, but different representations. We examined the performance of each method by measuring the distance between distributed representations for verb-object pairs ('do' and 'dishes' pair) and those for the corresponding basic verb ('wash') or predicated argument structures ("wash(dishes)"). We also experimentally compared the previous methods and ours on the same data set used in (Mitchell and Lapata, 2008).

2 Related work

There are many methods for acquiring word representations in vector space models. These methods can be classified into two approaches: the word occurrence approach and the word co-occurrence approach.

The word occurrence approach, including Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997), Probabilistic LSA (PLSA) (Hofman, 1999) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003), presupposes that distributions of word frequencies for each document (a word-document matrix) are given as input. In the word frequency approach, word representations are learned by applying singular value decomposition to the word-document matrix in LSA, or learning probabilities for hidden variables in PLSA or LDA. However, in the word frequency approach, the word frequencies for a document are given as a bag of words (BoW), and consequently, the information on the word order or phrase structure is not considered in these models.

The co-occurrence frequency approach, including distributional representations and neural network language models, uses statistics of the context around a word. The distributional representations for a word w are defined as a vector that represents the distribution of words (word frequencies) in a fixed-size window around word w, or the distribution of dependencies of word w, following the distributional hypothesis (Firth, 1957). Alternatively, neural network language models, including word2vec (Mikolov et al., 2013a; Mikolov et al., 2013b), GloVe (Pennington et al., 2014) and the vector Log-Bilinear Language model (vLBL) (Mnih and Kavukcuoglu, 2013), generate dense and low-dimensional vectors that represent word meanings by learning a neural network that solves a pseudo-task in which the neural network predicts a word given surrounding words. After the training of the neural network on a large corpus, the word vector for w is acquired by retrieving the weights between w and the hidden variables in the neural network (Bengio et al., 2003; Collobert and Weston, 2008). Word2vec is preferred in NLP because it learns distributed representations very efficiently. The conventional methods for neural network language models take several weeks to learn their models on tens of millions sentences in Wikipedia (Collobert et al., 2011). It is likely possible for word2vec to reduce the calculation time dramatically. However, these models basically learn word-to-word relations, not phrase or sentence structures.

When we make distributed representations for phrases or sentences, it is necessary to generate constitutive distributed representations for phrases or sentences based on the principle of compositionality. Mitchell and Lapata (2008) and Mitchell and Lapata (2010) proposed the add model, which generates distributed representations for phrase structures, whereas Goller and Küchler (1996), Socher et al. (2012) and Tsubaki et al. (2013) proposed Recursive Neural Network (RNN) models for phrase structures. Recently, new models based on tensor factorization have been proposed (Baroni and Zamparelli, 2010; Grefenstette and Sadrzadeh, 2011; Kartsaklis et al., 2012).

The add model is a method to generate distributed representations for phrase structures or multi-word expressions by adding distributed representations for each word that constitutes the phrase structure. However, the word order and syntactic relations are lost as a result of the adding in the model. For example, suppose that we have the distributed representations for a verb, a subject and an object. The result of adding the distributed representations is the same if we change the order of the subject and the object. For example, consider the distributed representations for the following two sentences.

- The girl gave a present.
- A present gave the girl.

The distributed representations for these sentences are as follows.

$$\begin{aligned} & \text{v(the)} + \text{v(girl)} + \text{v(gave)} + \text{v(a)} + \text{v(present)} \\ = \; & \text{v(a)} + \text{v(present)} + \text{v(gave)} + \text{v(the)} + \text{v(girl)} \end{aligned}$$

where $v(w)$ is the distributed representations for word w. It is necessary for the models to be sensitive to the word order to make a difference between these sentences. To solve these problems, various approaches have been proposed. For example, a method that adds weights to verbal vectors appearing ahead or one that assigns word-order numbers to n-grams was proposed. RNNM can acquire distributed representations for one sentence using RNN and a given syntactic tree (Socher et al., 2011; Socher et al., 2012). RNNM makes use of syntactic trees of sentences, as shown in Figure 1. It calculates a distributed representation for the parent node from the distributed representations for the child nodes in the syntactic trees. However, it uses only the skeletal structures of the syntactic trees; category and subject information in the syntactic trees are not used. Hashimoto et al. (2014) proposed a new method that acquires distributed representations for one sentence with information on words and phrase structures by using the parse trees generated by an HPSG parser called Enju.

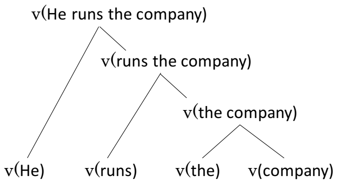

Figure 1: RNNM structure with syntax tree

Tensor factorization is a method that represents word meaning with not only vectors but also matrices. For example, a concept ‘car’ has many attributes such as information about color, shape, and functions. It seems to be difficult to represent phrases or sentences with a fixed-size vector because many concepts can appear in a sentence and each concept has its own attributes. Baroni and Zamparelli (2010) tried to represent attribute information of each word as a product of a matrix and a vector. Grefenstette and Sadrzadeh (2011) followed this approach and proposed new method that obtains the representations of verb meaning as tensors. Kartsaklis et al. (2012) proposed a method that calculates representations for sentences or phrases containing a subject, a verb and an object, based on Grefenstette and Sadrzadeh (2011)’s method. Recently, three dimensional tensors have been used for representing the relations of a subject, a verb and an object (de Cruys, 2009; de Cruys et al., 2013).

3 Word2vec

Word2vec (Mikolov et al., 2013a; Mikolov et al., 2013b) is the method to obtain distributed representations for a word by using neural networks with one hidden layer. It learns neural network models

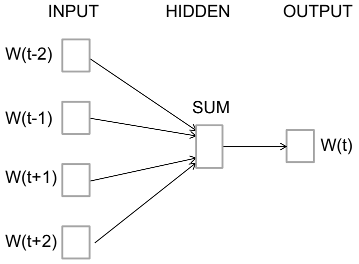

Figure 2: CBOW

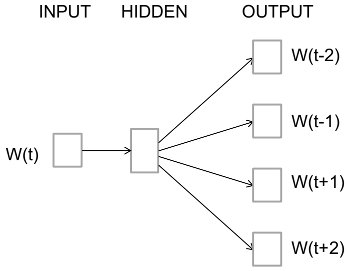

Figure 3: Skip-gram

from large texts by solving a pseudo-task to predict a word from surrounding words in the text. The word weights between the input layer and hidden layer are extracted from the network and become the distributed representation for the words. Mikolov et al. proposed two types of network for word2vec, the Continuous Bag-of-words (CBOW) model and the Skip-gram model.

3.1 CBOW model

Figure 2 shows the CBOW model's network structure. The CBOW model is a neural network with one hidden layer, where the input is surrounding words $w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_k$, and the output is w_t. The input layer and output layer are composed of nodes, each of which corresponds to a word in a dictionary; i.e., input and output vectors for a word are expressed in a 1-of-k representation. The node values in the hidden layer are calculated as the sum of the weight vectors of the surrounding words $w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_k$.

3.2 Skip-gram model

Figure 3 shows the Skip-gram model's network structure. The Skip-gram model is a neural network with one hidden layer in which a 1-of-k vector for word w_t is given as an input and 1-of-k vectors for the surrounding words $w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_k$ are output.

4 Proposed methods

This section explains the proposed methods to obtain the distributed representations for verb-object pairs by using word2vec. First, we explain the baseline for comparison of the proposed methods. Then, we describe the proposed methods.

4.1 Baseline method

The baseline method is the add model using word2vec. Word2vec is first trained with a large amount of text; then, distributed representations for each word are obtained. For example, the vector for "read" is obtained as "read = (1.016257, -1.567719, -1.891073,…,0.578905, 1.430178, 1.616185)". Distributed representations for a verb-object pair are obtained by adding the vector for the verb and the vector for the object.

4.2 Method 1

The CBOW model of word2vec is learned in a pseudo-task that predicts a word from surrounding words in the text. Thus, we expect that distributed representations for verb-object pairs can be acquired when the object is put near the verb. A large amount of training text is parsed by Enju, and new training text data is generated by inserting the object just after the verb for all verb-object pairs appearing in the corpus as follows.

(original) I did many large white and blue round dishes.

(modified) I **do dish** many large white and blue round dish.

Enju (Miyao et al., 2005; Miyao and Tsujii, 2005; Ninomiya et al., 2006) is a parser that performs high-speed and high-precision parsing and generates syntactic structures based on HPSG theory (Pollard and Sag, 1994), a sophisticated grammar theory in linguistics. In addition, Enju can generate predicate argument structures. The Stanford Parser (de Marneffe et al., 2006; Chen and D.Manning, 2014) is often used, but it can analyze only syntactic structures. Therefore, we used Enju, which can parse syntactic structures and predicate argument structures. In

Method 1, word2vec is trained from the new text data generated by using Enju's results to augment objects near verbs in the text. Then, distributed representations for verb-object pairs are generated by adding the distributed representations for the verb and the distributed representations for the object.

4.3 Method 2

We expect that distributed representations for verb-object pairs can be obtained by training word2vec with text in which each verb is concatenated with its object for all verb-object pairs. For each verb v and object o pair, v is replaced with $v : o$, where v and o are concatenated into a single word using Enju's result. The following shows an example of Method 2.

(original) I did many large white and blue round dishes.

(modified) I **do:dish** many large white and blue round dish.

Word2vec is learned using the new generated text, and distributed representations for verb-object pairs are acquired.

4.4 Method 3

The Skip-gram model is learned by solving a pseudo-task in which a word in the text is given as input, and the neural network predicts each surrounding word. It is likely that distributed representations for verb-object pairs can be acquired by providing the verb and its object to the neural networks at the same time when the input word is a verb. We performed the learning in Method 3 by using a new Skip-gram model wherein the verb-object pair is input to the neural networks when one of the input words is a verb.

Figure 4 shows the neural network model for Method 3. The model is trained from a large amount of text, and distributed representations for words are generated. Then, the distributed representations for verb-object pairs are acquired by summing the distributed representations for the verb and the distributed representations for the object in the same way as Method 1.

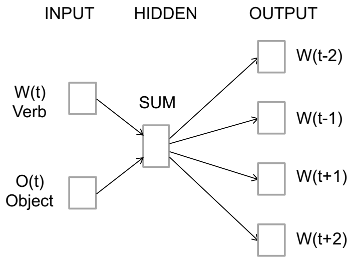

Figure 4: New Skip-gram model

5 Experiments and evaluations

We performed two experiments to evaluate the performance of Methods 1, 2, 3, and the baseline method. We used word2vec in the experiments for Methods 1, 2, and the baseline with the CBOW model option (-cbow 1) and a modified word2vec for Method 3 based on the Skip-gram model. In all methods, the maximum window size was 8 words (-window 8), the sample number for negative sampling was 25 (-negative 25), and we did not use hierarchical softmax (-hs 0). The number of nodes in the hidden layer was 200; i.e., the number of dimensions for the distributed representations was 200.

5.1 Experiment on light verb-object pairs

We performed an experiment on pairs of a light verb and an object. The training corpus consisted of the English Gigaword 4th edition (LDC2009T13, nyt_eng, 199412 - 199908), Corpus of Contemporary American English (COCA), and Corpus of Historical American English (COHA). The size of the training corpus was about 200 million words.

We developed a data set that consists of 17 triples of a light verb, an object, and a basic verb. The basic verb is one that almost has the same meaning as the corresponding light-verb and object pair. Table 1 shows examples of the data set. The pairs were selected from "Eigo Kihon Doushi Katsuyou Jiten (The dictionary of basic conjugate verbs in English)" (Watanabe, 1998) and a web site[1]. The basic verbs were selected from "Eigo Kihon Doushi Jiten (The dictionary of basic verbs in English)" (Konishi, 1980).

We evaluated each method by measuring the

[1] web page (http://english-leaders.com/hot-three-verbs/, 1/20/2015 reference)

Table 1: Examples of distributed representations for light verb-object pairs

verb-object pairs	basic verbs	examples
do-dish	wash	I do the dishes.
do-cleaning	clean	I'll do the cleaning.
do-nail	put, paint, dress	We do our hair, and then we do our nails.
do-laundry	wash	I'm doing the laundry.
have-lunch	eat	Let's have lunch.
have-tea	drink	Let's have some tea.
have-word	tell, talk, speak	I'd like to have a word with you.
make-call	call	I always get nervous whenever I make a call.
make-bed	clean, put, set	I make the bed.
hold-door	open	Hold the door.
hold-tongue	shut	Hold your tongue!
give-hand	help	Give me a hand with this box.
give-party	hold, have, throw	She is giving a party this evening.
give-news	report, present, announce	I will probably be able to give you good news.
finish-coffee	drink	He finished his coffee.
read-shakespeare	read	I read Shakespeare.
enjoy-movie	watch,see	Did you enjoy the movie?

mean reciprocal rank (MRR) score for each verb-object pair in the data set, supposing that the corresponding basic verb is the true answer for the pair. Given a verb-object pair, we calculated its MRR score as follows. First, we calculated the cosine distance between the verb-object pair and all basic verb candidates in the dictionary. Then, we ranked the basic verbs in accordance with the cosine measure. The candidates of the basic verbs were 385 words in the basic verb dictionary (Konishi, 1980).

5.2 Comparison with conventional methods

We also conducted experiments with the data set[2] provided by Mitchell and Lapata (2008). This set consists of triples ($pair1$, $pair2$, $similarity$), from which we used 1890 verb and object pairs. The semantic similarity scores in the data set are given manually and range between 1 (low similarity) to 7 (high similarity). There are three types of combinations for $pair1$ and $pair2$ in the data: adjective + noun, noun + noun, and verb + object. For example, the similarity score for "vast amount" and "large quantity" is 7, and the similarity score for "hear word" and "remember name" is 1. We calculated Spearman's rank correlation coefficient on the "verb + object" part of this data set. The similarity scores for verb-object pair $pair1$ and $pair2$ were calculated using the cosine similarity between the vector for $pair1$ and the vector for $pair2$. If a system achieved a higher correlation coefficient, this means that its judgment was similar to that of humans.

Table 2: Results for light verb-object pairs (Average of MRR)

baseline	Method 1	Method 2	Method 3
0.27	0.35	0.37	0.31

6 Results

6.1 Results for light verb-object pairs

Table 2 shows the average MRR score for each method. Method 2 achieved the best result. We consider that training with the text in which verb-object pairs were replaced with a single expression had a good effect on word2vec. Method 1 and Method 3's similarities were also higher than those of the baseline method. Therefore, it can be considered that distributed representations for verb-object pairs that were sensitive to verb-object relations were acquired by improving the training data. However, Method 1 achieved a higher MRR than that of Method 3. We consider that this is because Method 3 learned the model from heterogeneous structures; i.e., the hidden layer in the neural networks received different signals depending on whether the input was a verb or not.

Table 3 shows the details of the experimental results. From the table, we can see that Method 1 outperforms Method 2 in many cases, although the average MRR of Method 2 is greater than that of Method 1. We think that this is because Method

[2] http://homepages.inf.ed.ac.uk/s0453356/share

Table 3: Details of the experiment

VO	baseline	Method 1	Method 2	Method 3
do-dish	0.03	0.08	1	0.07
do-cleaning	0.05	0.14	0.25	0.33
do-nail	0.02	0.02	0.38	0.06
do-laundry	0.17	0.07	0.09	0.14
have-lunch	1	1	0.2	0.33
have-tea	0.5	1	1	0.5
have-word	0.12	0.07	0.05	0.12
make-call	1	1	1	1
make-bed	0.02	0.04	0.03	0.05
hold-door	0.02	0.5	0.2	0.5
hold-tongue	0.01	0.01	0.005	0.01
give-hand	0.03	0.13	0.05	0.05
give-party	0.05	0.07	0.01	0.19
give-news	0.02	0.12	0.01	0.06
finish-coffee	0.5	0.5	1	0.5
read-shakespeare	1	1	1	1
enjoy-movie	0.11	0.13	0.02	0.38

2 achieved similarity 1 in some cases, and this increased the average MRR.

6.2 Comparison with conventional method

Table 4 shows the results of Methods 1, 2, and 3 and the baseline method using Skip-gram and CBOW with Mitchell and Lapata's data set. Method 1 using CBOW and size 50 achieved the best result. The reason is the process of learning. The CBOW model predicts a word by adding the vectors of surrounding words. Therefore, Method 1 with the CBOW model predicts a word from the sum of the vectors for a verb and its object. Consequently, representations for verb-object pairs are consistent in the learning and generating processes.

Table 5 shows the comparison with other methods. BL, HB, KS, and K denote the results of the methods of Blacoe and Lapata (2012), Hermann and Blunsom (2013), Kartsaklis and Sadrzadeh (2013), and Kartsaklis et al. (2013). Kartsaklis and Sadrzadeh (2013) used the ukWaC corpus (Baroni et al., 2009), and the other methods used the British National Corpus (BNC). Word2vec is the result of Hashimoto et al. (2014). They used the POS-tagged BNC and trained 50-dimensional word vectors with the Skip-gram model. We believe that our methods can be improved by using POS-tagged texts.

Table 4: Results for verb-object pairs in Mitchell and Lapata's data set (Spearman's rank correlation coefficient)

Method	Option	Score
Base-line	CBOW, -size 50	0.323
Method1	CBOW, -size 50	0.329
Method2	CBOW, -size 50	0.233
Base-line	Skip-gram, -size 50	0.308
Method1	Skip-gram, -size 50	0.305
Method2	Skip-gram, -size 50	0.173
Method 3	Skip-gram, -size50	0.272
Base-line	CBOW, -size 200	0.321
Method1	CBOW, -size 200	0.328
Method2	CBOW, -size 200	0.201
Base-line	Skip-gram, -size 200	0.308
Method1	Skip-gram, -size 200	0.292
Method2	Skip-gram, -size 200	0.171
Method 3	Skip-gram, -size200	0.275

Table 5: Comparison with other methods

Method	Score
Method 1 with CBOW	0.329
BL w/ BNC	0.35
HB w/ BNC	0.34
KS w/ ukWaC	0.45
K w/BNC	0.41
Word2vec	0.42

7 Conclusion and future work

This paper proposed methods for obtaining distributed representations for verb-object pairs by using word2vec. We experimentally evaluated them in comparison with the baseline add method in terms of mean reciprocal rank and Spearman's rank correlation. Method 2, which concatenates verbs with their objects in the text, achieved the best MRR score in the experiment on light verb-object pairs. Method 1, which puts objects nearby verbs, achieved the best correlation coefficient in the experiment on Mitchell and Lapata's data set. We consider that the training text data in these experiments was too small. It is necessary to use a large amount of data to verify which method is best for obtaining distributed representations of verb-object pairs. Using a large amount of data and making comparisons with RNNM and tensor factorization are left as future work.

Acknowledgments

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B) Grant Number 25280084.

References

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *proceedings of the Conference on the Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 1183–1193.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *proceedings of Language Resources and Evaluation Conference (LREC 2009)*, pages 209–226.

Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022.

Danqi Chen and Christopher D.Manning. 2014. A fast and accurate dependency parser using neural networks. In *proceedings of the Conference on the Empirical Methods in Natural Language processing (EMNLP 2014)*, pages 740–750.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language proccesing: Deep neural networks with multitask learning. In *proceedings of the International Conference on Machine Learning (ICML 2008)*, pages 160–167.

Ronan Collobert, Jason Weston, Leon Bottou, Michael karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, pages 2493–2537.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2013. A tensor-based factorization model of semantic compositionality. In *proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2013) : Human Language Technologies*, pages 1142–1151.

Tim Van de Cruys. 2009. A non-negative tensor factorization model for selectional preference induction. In *proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 83–90.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D.Manning. 2006. Generating typed dependency parses from phrase structure parses. In *proceedings of Language Resources and Evaluation Conference (LREC 2006)*, pages 449–454.

John R. Firth. 1957. A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis*, pages 1–32.

Christoph Goller and Andreas Küchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. *International Conferenece on Neural Networks*.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *proceedings of the Conference on the Empirical Methods in Natural Language processing (EMNLP 2011)*, pages 1394–1404.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. 2014. Jointly learning word representations and composition functions using predicate-argument structures. In *proceedings of the Conference on the Empirical Methods in Natural Language processing (EMNLP 2014)*, pages 1544–1555.

Karl Moritz Hermann and Philip Blunsom. 2013. The role of syntax in vector space models of compositional semantics. In *Annual Meeting of the Association for Computational Linguistics*, pages 894–904.

Thomas Hofman. 1999. Probablistic latent semantic analysis. In *Uncertainity in Artificial Intelligence*.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2013. Prior disambiguation of word tensors for constructing sentence vectors. In *proceedings of the Conference on Empirical Methods in Natural Language processing (EMNLP 2013)*, pages 1590–1601.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. 2012. A unified sentence space for categorical distributional-compositional semantics: Theory and experiments. In *proceedings of the International Conference on Computational Linguistics (Coling 2012)*, pages 549–558.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. 2013. Separating disambiguation from composition in distributional semantics. In *proceedings of the Conference on Natural Language Learning (CoNLL 2013)*, pages 114–123.

Tomohichi Konishi. 1980. *Eigo Kihon Doushi Jiten (The dictionary of basic verbs in English)*. Kenkyusha publication.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *proceedings of workshop at the International Conference on Learning Representations (ICLR 2013)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *proceedings of the Association for Computational Linguistics (ACL 2008)*, pages 236–244.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive sentence*, 34(8):1388–1439.

Yusuke Miyao and Jun'ichi Tsujii. 2005. Probabilistic disambiguation models for wide-coverage hpsg parsing. In *proceedings of the Association for Computational Linguistics (ACL 2005)*, pages 83–90.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii, 2005. *Keh-Yih Su, Jun'ichi Tsujii, Jong-Hyeok Lee and Oi Yee Kwong (Eds.), Natural Language Processing - IJCNLP 2004 LNAI 3248*, chapter Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank, pages 684–693. Springer-Verlag.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Conference on Neural Information Processing System 2013*, pages 2265–2273.

Takashi Ninomiya, Takuya Matsuzaki, Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2006. Extremely lexicalized models for accurate and fast hpsg parsing. In *proceedings of the Conference on the Empirical Methods in Natural Language processing (EMNLP 2006)*.

Jeffery Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *proceedings of the Conference on the Empirical Methods in Natural Language processing (EMNLP 2014)*, pages 1532–1543.

Carl Pollard and Ivan A. Sag. 1994. Head-driven phrase structure grammar. *University of Chicago Press*.

Richard Socher, Eric H. Huang, Jeffrey Pennin, Christpher D. Manning, and Andrew Y. Ng. 2011. Dynamic pooling and unfolding recursive autoencorders for paraphrase detection. In *Advances in Neural Information Processing Systems 24*, pages 801–809.

Richard Socher, Brody Huval, Christpher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *proceedings of the Conference on the Empirical Methods in Natural Language processing (EMNLP 2012)*, pages 1201–1211.

Masashi Tsubaki, Kevin Duh, Masashi Shimbo, and Yuji Matsumoto. 2013. Modeling and learning semantic co-compositionality through prototype projections and neural netoworks. In *proceedings of the Conference on the Empirical Methods in Natural Language processing (EMNLP 2013)*, pages 130–140.

Miyoko Watanabe. 1998. *Eigo Kihon Doushi Katsuyou Jiten (The dictionary of basic conjugate verbs in English)*. Nagumo phoenix publication.