EMBEDDING PARALLELEPIPED IN CO-OCCURRENCE MATRIX: SIMULATION AND EMPIRICAL EVIDENCE

Takuma Torii[*1], Akihiro Maeda[1], and Shohei Hidaka[*1]

[*]Corresponding Authors: {tak.torii,shhidaka}@jaist.ac.jp
[1]Japan Advanced Institute of Science and Technology, Ishikawa, Japan

Recent natural language processing technologies are based on the vector space models of language, in which each word is represented with a vector in high dimensional space. One of the earliest successes using the vector space models is the four-term analogical reasoning task. A certain quadruple of word vectors forms a "parallelogram" in the vector space, with which the fourth word vector is inferred as an answer to a given triplet of query words. Despite the large body of successful applications of the vector space models, it has remained unknown what the parallelogram means. This study aims to reveal this mystery. Our analysis suggested that analogical reasoning is possible by decomposition of the bigram co-occurrence matrix, and we demonstrated the formation of a parallelepiped by creating a miniature corpus and its word vectors. This analysis and demonstration imply a sort of symmetry or exchangeability in word-word co-occurrence structure.

1. Introduction: Distributional Models of Language

'Evolution' of the language processing capability of the machine is dramatic in this decade. The accuracy of machine translation has reached the human level or perhaps more than educated non-native speakers. These successes of machine-learning language models have suggested how natural languages are organized.

Language is usually considered as an organized system that exhibits the capability of determining a class of words given the context of a word to be determined. This theoretical idea is called *the distributional hypothesis* (Harris, 1954). The distributional hypothesis postulates that words that occur in similar contexts tend to have similar meanings. For example, 'an apple' and 'a banana' both are allowed to appear in similar contexts, e.g., "she eats _____ every morning" and "_____ is a fruit". However, they may not appear in similar contexts of 'a bus' and 'a train', e.g., "she takes _____ home". When we think of the fill-in-the-blank problem "she eats _____ every morning", the words that refer to something edible, women like, and common in breakfast would be coming up with, like those specified by the context of the blank to be filled.

One of the pervasive methods to implement the distributional hypothesis is counting the co-occurrence of words in the pairs, triplets, or n-grams. Such naive co-occurrence counting has, however, a few technical issues: the combinatorial

space of word pairs is too large to sample sufficiently (e.g., a bigram (pair) table has 10^{12} cells for 10^6 word types), and it causes underestimation on the co-occurrence probability. Thus, one needs further *compressed* representations of the co-occurrence table, that compressed representations hopefully preserve the distributional structure of the words in the table and the language. Latent Semantic Analysis (LSA) (Landauer & Dumais, 1997) is one of such earliest attempts. The underlying idea of LSA is that a sparse co-occurrence matrix M can be approximated by vector representation of words, called *word vectors*. It has been demonstrated that word-vector algorithms can solve semantic tasks, although their performances were limited (see Lenci (2018) for review).

More recently, Mikolov et al. (2013) discovered that four-term analogy problems can be solved accurately by their artificial neural network called *skip-gram*, which is an instance of the *word2vec* class of models. Four-term analogy problem questions "what is d to c as b is to a?" denoted by, a : b :: c : d. Formally, the model needs to predict word d given the triple of query words a, b, and c. For example, the question, man : woman :: king : _____, should be answered with 'queen'. Importantly, the word2vec was not optimized to solve the four-term analogy questions, but it was optimized to predict the context words for each word. However, with the learned word vectors, e.g., $v_{\text{king}}, v_{\text{man}}, v_{\text{woman}}$, one can answer the analogy task by vector arithmetic $v_{\text{king}} - v_{\text{man}} + v_{\text{woman}} \approx v_{\text{queen}}$. Since analogical reasoning requires not only syntactic but also semantic aspects of language, their successes in the analogy task have been viewed as strong support for the distributional hypothesis. And since analogy was considered to be 'uniquely' humans, this discovery gave a strong impact on a variety of research fields.

To solve such analogical questions, word2vec needs to successfully extract latent and distributional structures of the language, which is represented in the vector form. Since then, researchers of related fields have been attracted to resolve this "mystery" of word2vec, e.g., (Levy et al., 2015; Arora et al., 2015; Hashimoto et al., 2016). Most of them have concluded that the emergence of parallelograms is due to the sophisticated learning algorithm. The current consensus (see, e.g., (Lenci, 2018)) is the conclusion by (Levy et al., 2015) that the analogy performance of word2vec can be explained as a result of a factorization of the PPMI (positive pointwise mutual information), one of the most popular preprocess of co-occurrence matrix in natural language processing (NLP).

In this paper, we take a different approach to the mystery of word2vec. We hypothesized that the word co-occurrence matrix itself, rather than some transformation of it such as PPMI, has sufficient information required to solve linguistic tasks. Namely, we take one of the simplest forms of implementation of the distributional hypothesis. This approach has been rarely taken in the existing literature.

Connecting the co-occurrence matrix to analogical parallelograms directly naturally leads constructive approach — simulation to test which type of co-occurrence may embed a parallelogram in the word vector space. Thus, we take

the two types of approaches, data-driven analysis of co-occurrence matrix and constructive simulation creating and manipulating a small corpus.

In what follows, we briefly introduce the word2vec model in Section 2, followed by an analysis of a co-occurrence matrix in Section 3, and the constructive approach in Section 4. Lastly, we discuss future directions toward the understanding of the semantic nature of underlying word co-occurrence.

2. word2vec: The Word Embedding Algorithm

We briefly introduce the key ideas of word2vec, specifically of the skip-gram artificial neural network architecture. The skip-gram model consists of the three layers, n input units, d hidden units, and n output units, where n is the vocabulary size. Initially, every word w in vocabulary W is represented by a so-called one-hot vector e_w of length n. Given a long sequence of words represented by one-hot vectors, the goal of optimization is to obtain a d dimensional compressed representation v_w, called word vector, for every word $w \in W$, where $d \ll n$. Denote by w_t a word at the position t in the corpus. The skip-gram seeks the corpus to identify every subsequence $(w_{t-k}, \ldots, w_{t-1}, w_t, w_{t+1}, \ldots, w_{t+k})$, the k preceding and k following context words around the centre word w_t. The skip-gram model is trained, to optimize the latent word vectors $\{v_w\}_{w \in W}$, for each w_t to predict their all context words $\vec{w}_t = (w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+k})$ simultaneously throughout the corpus. Mikolov et al. (2013) defined for the skip-gram model the conditional probability of occurring y in the context of x as follows:

$$P(y|x) = \frac{\exp(v_y \cdot v_x)}{\sum_{w \in W} \exp(v_w \cdot v_x)} \ , \tag{1}$$

where $v_y \cdot v_x$ is the inner product of word vectors v_x and v_y.

Using a trained word2vec, Mikolov et al. (2013) demonstrated that it can solve their four-term analogy questions. Consider, for example, the problem, man : woman :: king : _____, and the correct answer is 'queen'. Given the word vectors $v_{\text{man}}, v_{\text{woman}}, v_{\text{king}}$ for the cue words, decide the most likely word y by calculating the cosine similarity measure $\text{cosine}(v_x, v_y) = \frac{v_x \cdot v_y}{\|v_x\| \, \|v_y\|}$ for all words x:

$$v_y = \underset{v_x \,:\, x \in W}{\text{argmax}} \ \text{cosine}(v_{\text{man}} - v_{\text{woman}} + v_{\text{king}}, \ v_x) \ . \tag{2}$$

It is defined correct, if the word vector v_y is v_{queen}. The overall percentage of correct answers is about 66% for the 19,544 questions.

If any model answers correctly for a quadruple using Equation (2), these four word vectors need to form a parallelogram in the vector space. Indeed, Mikolov et al. (2013) graphically showed parallelograms in a lower-dimensional subspace.

3. Analogical Reasoning with Raw Co-occurrence Matrix

The past studies exploring the analogical reasoning based on the word2vec or others (Levy & Goldberg, 2014; Hashimoto et al., 2016; Arora et al., 2015) have es-

sentially hypothesized and concluded that word2vec or other transformation such as PPMI is crucial to have a good analogy performance. In this study, however, we hypothesize that a raw co-occurrence matrix itself or its matrix decomposition would be sufficient for analogical reasoning.

3.1. *Method*

To test our hypothesis, we directly counted the frequencies of pairwise co-occurrence of all words in the English Wikipedia dump corpus 20171001. The text data contains approximately 7.9 billion words, of which 2.6 million words are unique. The window size for word pair counting was $k = 5$. Although we counted them all, algebraic operations using the full co-occurrence matrix were impossible due to our computational power. Hence, for the analogical task, we only used the sub-matrices composed of the top 1,000 (or 10,000) unique words, in addition to the 905 unique words in the question set. Denote this co-occurrence matrix by $M \in \mathbb{R}_{\geq 0}^{n \times n}$ with vocabulary size n.

In NLP, it is commonly recognized that application of singular-value decomposition (SVD) to the co-occurrence matrix improves performance of linguistic tasks. Technically, SVD is a method for decomposition of a real matrix M of arbitrary finite size to the form $M = U\Sigma V^\top$, where matrix U and V are real orthogonal matrices and the diagonal matrix Σ contains singular values in its diagonal elements. By taking the first d dimensions, the d dimensional word vectors for n words are obtained as $U_d\, \Sigma_d^{1/2} \in \mathbb{R}^{n \times d}$. Since the word2vec was trained to construct 300 dimensional word vectors, $d = 300$ was used in this paper.

We trained our word2vec (skip-gram) model using the sample code of Python library Gensim (Rehurek & Sojka, 2010). We used instead our own preprocessed text data as described above. The window size $k = 5$ is the same. Only the words that occur more than or equal to 100 times in the corpus were used for training the model. The number of unique words was approximately 0.32 million.

3.2. *Results*

Figure 1 shows the performance for the four-term analogy task using the distributional models. As shown by Mikolov et al. (2013), the performance of word2vec is 66%. We treat this as a benchmark. For the models freq, the rows of the co-occurrence frequency matrix M were directly used as word vectors. The models showed accuracy below 5%. For the models logfreq, the logarithms of the rows of M were used as word vectors. By taking the logarithms, the model performances got significantly increased about 40% and 35%. We think the logarithm worked as a smoothing against the Zipf's law. This partially supports our hypothesis that information required to solve linguistic tasks is inside the corpus data. However, there is room for further improvement (could be) induced by word2vec. To eliminate this possibility, we applied SVD, a classic word embedding method,

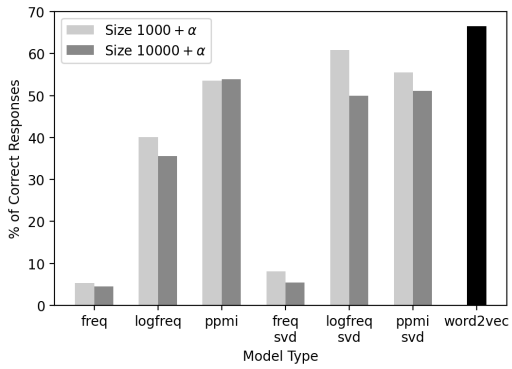

Figure 1. Four-term analogy performances of distributional models

to the log-frequency matrices of M. Since SVD is linear, although the word2vec is nonlinear, it would be helpful to resolve the mystery of word2vec. Surprisingly, the performances of the logfreq svd models are above 60% comparable to word2vec. This result supports the other half of our hypothesis that there is no latent structure that can be discovered only when using word2vec.

3.3. *Discussion: Why the Decomposed Co-occurrence Matrix Suffices*

If the original word2vec (skip-gram) were successfully trained, the word vectors $V \in \mathbb{R}^{n \times d}$ determines conditional co-occurrence probability matrix $P(y|x)$ in Equation (1). By taking logarithm, $V V^\top \in \mathbb{R}^{n \times n}$ is extracted (the normalizing term was ignored), and thus the skip-gram model could be viewed as an approximate matrix decomposition of the form $V V^\top \approx \tilde{M}$ for unknown $\tilde{M}$. Given the results in Section 3.2, it suggests that "up-to-d^{th}-rank matrix decomposition of the logarithm of M" is essentially what the word2vec models do. This hypothesis differs from the previous study (Levy et al., 2015), which concluded that word2vec is equivalent to the PPMI-like smoothing, or a matrix decomposition of the PPMI-smoothed matrix of M. Our view, that word2vec as a co-occurrence matrix decomposition, can be viewed as one of the simplest and most straightforward implementations of the distributional hypothesis (Harris, 1954).

4. Constructive Approach to the Parallelograms

The analysis has suggested that there is a subspace of the co-occurrence matrix, in which a parallelogram is formed by a particular set of word vectors as each word may have multiple aspects. For example, *king* is more similar with *queen* on the *is-royal* axis, but is more similar with *man* on the *is-male* axis. Such a multi-aspect structure of the word *king* is supposed to be captured by a parallelepiped, rather

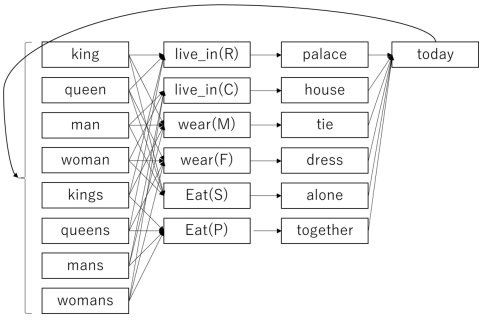

Figure 2. A hidden Markov model generating the 24 sentences in the toy corpus. Any hidden state X other than the verbs generates the word X by probability 1. For example, the state "king" generates the word "king". On the other hand, the two hidden states corresponding to the two verbs may generate the same word. For example, both states "live_in(R)" and "live_in(C)" generate the word "live_in".

than a parallelogram. Although an analogy task tests a parallelogram, a collection of analogy tasks would test a parallelepiped or more complex geometric object.

In this section, we take a constructive approach to address how this parallelepiped structure is involved with the syntactic or semantic nature of a language. Specifically, we construct a small toy corpus, that forms an idealized parallelepiped structure among the word vectors, and analyzed what condition would be essential to form some parallelepiped of word vectors.

4.1. *Demonstrating Parallelepiped Embedded in Co-occurrence Matrix*

Toy Corpus. We created a corpus of 24 artificial sentences, which are not strictly grammatical, but with a minimal syntactic and semantic structure. Each of the sentences in this corpus consists of four words in the form of Subject-Verb-Object-Adverbial, such as "king live-in palace today". There are eight subjects, three verbs, six objects, and one adverb — in total 18 words. The corpus does not have all the possible sentences out of these 18 words, $144 = 8 \times 3 \times 6$, but it has only 24 sentences (Figure 2), which implicitly represents the hypothetical semantic relationship between underlying concepts to which these words refer.

First, we analyzed the co-occurrence matrix constructed for the toy corpus with each of the sentences generated by the equal probability $1/24$. In this case, the co-occurrence matrix (up to scale and permutation similarity) can be written with the two block matrices $C_0 \in \mathbb{R}^{8\times 10}$ and $C_1 \in \mathbb{R}^{10\times 10}$ by $C = \begin{pmatrix} \mathbf{0}_{8,8} & C_0 \\ C_0^\top & C_1 \end{pmatrix}$. Note that each row vector of the block matrix C_0 is the non-zero part of word vectors of the eight subject nouns. C_0 has the rank 4, and it lives in 3 dimensional affine space. Namely, there is some linearly independent basis of three vectors $b_0, b_1, b_2, b_3 \in \mathbb{R}^8$, such that $C_0 = (b_1, b_2, b_3)A + b_0\mathbf{1}_{1,10}$ with a unique matrix $A \in \mathbb{R}^{3\times 10}$ for each choice of the affine basis $B = (b_0, b_1, b_2, b_3)$. Let $\bar{B} =$

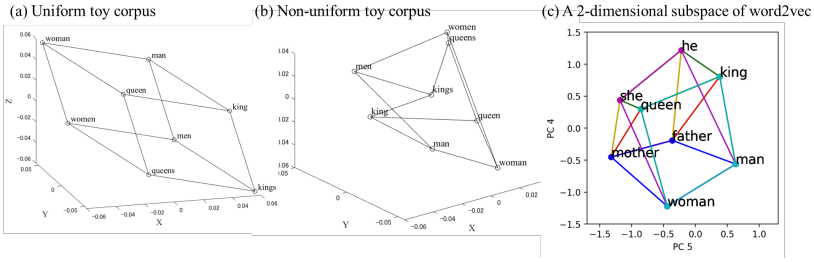

Figure 3. (Non-)parallelepipeds embedded in the co-occurrence matrix of (a) uniform toy corpus, (b) non-uniform toy corpus, and (c) a natural corpus.

(b_1, b_2, b_3) has $b_1, b_2, b_3 \in \mathbb{R}^6$ of non-zero vectors of the column vectors of $\bar{C}_0 := C_0^\top - \frac{1}{10}\mathbf{1}_{1,10}C_0^\top$. Then the three dimensional coordinates of the 8 points are given by the column vectors of $(\bar{B}^\top \bar{B})^{-1}(\bar{B}^\top \bar{C}_0)$, in which a "parallelepiped" is embedded (Figure 3(a)). Thus, this uniform toy corpus gives a sufficient condition or the existence of a way to embed a parallelepiped in the co-occurrence matrix.

4.2. *Symmetry Breaker Against parallelepiped*

It is also important to demonstrate on which condition the parallelepiped embedded in a co-occurrence matrix is *broken*, as such a demonstration gives a necessary condition for the parallelepiped formation. To do so, we consider a variation of the toy corpus, called non-uniform toy corpus, in which a certain randomly assigned probability p_i to sample the i^{th} sentence to build the co-occurrence matrix. Figure 3(b) shows the same set of the eight word vectors visualized in the same way as Figure 3(a), for a set of non-uniform random probabilities p_i. These eight word vectors form neither a parallelepiped nor parallelograms. As the only difference between the uniform and non-uniform toy corpus is their sampling probability, this result suggests that a certain symmetric relationship in the probability distributions is needed to hold the parallelepiped.

4.3. *A Parallelepiped in Natural Co-occurrence*

The demonstration with the toy corpus above suggests that a certain class of word vectors would form a parallelepiped relationship, if the class of vectors two or more show independent syntactic-semantic statistical regularities on its word usage. We test this prediction by searching whether such a parallelepiped for a class of word vectors embedded in a natural co-occurrence matrix (logfreq svd, size 1000). Figure 3(c) shows an example that we found in the set of 8-tuple word vectors of the question words in the Family category (Mikolov et al., 2013) visualized by a two-dimensional subspace of the principal component analysis. This confirms our prediction.

5. Conclusion

This study attempted to give a theoretical account of what the parallelogram means in the vector space model. Our analysis of the co-occurrence matrix suggests a sort of co-occurrence matrix decomposition can give such a parallelogram useful for analogical reasoning. This empirical observation leads us to a constructive approach to building a toy corpus that may or may not embed a parallelepiped in the co-occurrence matrix. This numerical simulation suggests that the parallelepiped is tightly related to a certain class of the sentence probability distribution, perhaps less restricted than uniform but more restricted than arbitrary.

The biolinguistic enterprise of seeking cognitive precursors to human language depends on hypotheses or views on the structure of language. Our 'parallelotope hypothesis' may provide yet another characterization of the structure of language: word representations being structured are at least utilized for analogical reasoning among words. This hypothesis makes a strong connection between the mental representation of words of a language and relational reasoning on words. This hypothesis may motivate comparative psychology research on precursors to language in terms of the ability of relational reasoning.

Acknowledgements

This work is supported by JSPS KAKENHI JP 20H04994, JST PRESTO JPMJPR20C9, Japan.

References

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2015). A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics (TACL)*, *4*, 385-399.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2–3), 146–162.

Hashimoto, T. B., Alvarez-Melis, D., & Jaakkola, T. S. (2016). Word embeddings as metric recovery in semantic spaces. *TACL*, *4*, 273–286.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory. *Psychological Review*, *104*(2), 211–240.

Lenci, A. (2018). Distributional models of word meaning. *The Annual Review of Linguistics*, *4*, 151–171.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Proceedings of NIPS 2014* (pp. 2177–2185).

Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *TACL*, *3*, 211-225.

Mikolov, T. et al. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS 2013* (pp. 3111–3119).

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC Workshop* (pp. 45–50).