The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Using *k*-Way Co-Occurrences for Learning Word Embeddings

Danushka Bollegala,[1] **Yuichi Yoshida,**[2] **Ken-ichi Kawarabayashi**[2,3]
University of Liverpool, Liverpool, L693BX, United Kingdom[1]
National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan[2]
Japan Science and Technology Agency, ERATO, Kawarabayashi Large Graph Project[3]

Abstract

Co-occurrences between two words provide useful insights into the semantics of those words. Consequently, numerous prior work on word embedding learning has used co-occurrences between two words as the training signal for learning word embeddings. However, in natural language texts it is common for multiple words to be related and co-occurring in the same context. We extend the notion of co-occurrences to cover $k(\geq 2)$-way co-occurrences among a set of k-words. Specifically, we prove a theoretical relationship between the joint probability of $k(\geq 2)$ words, and the sum of ℓ_2 norms of their embeddings. Next, we propose a learning objective motivated by our theoretical result that utilises k-way co-occurrences for learning word embeddings. Our experimental results show that the derived theoretical relationship does indeed hold empirically, and despite data sparsity, for some smaller $k(\leq 5)$ values, k-way embeddings perform comparably or better than 2-way embeddings in a range of tasks.

1 Introduction

Word co-occurrence statistics are used extensively in a wide-range of NLP tasks for semantic modelling (Church and Hanks 1990). As the popular quote from Firth—*you shall know a word by the company it keeps* (Firth 1957), the words that co-occur with a particular word provide useful clues about the semantics of the latter word. Co-occurrences of a target word with other (context) words in some context such as a fixed-sized window, phrase, or a sentence have been used for creating word representations (Mikolov, tau Yih, and Zweig 2013; Mikolov, Chen, and Dean 2013; Pennington, Socher, and Manning 2014). For example, skip-gram with negative sampling (SGNS) (Mikolov, Chen, and Dean 2013) considers the co-occurrences of two words within some local context, whereas global vector prediction (GloVe) (Pennington, Socher, and Manning 2014) learns word embeddings that can predict the total number of co-occurrences in a corpus.

Unfortunately, much prior work in NLP is limited to the consideration of co-occurrences between two words due to the ease of modelling and data sparseness. Pairwise co-occurrences can be easily represented using a co-occurrence matrix, whereas co-occurrences involving more than two words would require a higher-order tensor (Socher et al. 2013). Moreover, co-occurrences involving more than three words tend to be sparse even in large corpora, requiring compositional approaches for representing their semantics (Van de Cruys, Poibeau, and Korhonen 2013). It remains unknown – *what statistical properties about words we can learn from k-way co-occurrences among words*. In this paper, we define the term k-way co-occurrence to denote the co-occurrence between k distinct words in some context such as a token-window, sentence, paragraph or a document.

Words do not necessarily appear as pairs in sentences. By splitting the contexts into pairs of words, we loose the rich contextual information about the nature of the co-occurrences. For example, consider the following sentences.

(a) *John and Anne are friends.*

(b) *John and David are friends.*

(c) *Anne and Mary are friends.*

Sentence (a) describes a three-way co-occurrence among (*John*, *Anne*, *friend*), which if split would result in three two-way co-occurrences: (*John*, *Anne*), (*John*, *friends*), and (*Anne*, *friends*). On the other hand, Sentences (b) and (c) would collectively produce the same two two-way co-occurrences (*John*, *friend*) and (*Anne*, *friend*), despite not mentioning any friendship between *John* and *Anne*. Therefore, by looking at the three two-way co-occurrences produced by Sentence (a) we cannot unambiguously determine whether *John* and *Anne* are friends. Therefore, we must retain the three-way co-occurrence (*John*, *Anne*, *friend*) to preserve this information.

Although considering k-way co-occurrences is useful for retaining the contextual information, there are several challenges one must overcome. First, the number of k-way co-occurrences tend to be sparse for larger k values. Such sparse co-occurrence counts might be inadequate for learning reliable and accurate semantic representations. Second, the unique number of k-way co-occurrences grows exponentially with k. This becomes problematic in terms of memory requirements when storing all k-way co-occurrences. A word embedding learning method that considers k-way co-occurrences must overcome those two challenges.

In this paper, we make several contributions towards the understanding of k-way co-occurrences.

- We prove a theoretical relationship between the joint probability of k words, and the squared sum of ℓ_2 norms of their embeddings (§3). For this purpose, we extend the work by Arora et al. (2016) for two-way co-occurrences to $k(>2)$-way co-occurrences.
- Motivated by our theoretical analysis, we propose an objective function that considers k-way co-occurrences for learning word embeddings (§4). We note that our goal in this paper is *not* to propose novel word embedding learning methods, nor we claim that k-way embeddings produce state-of-the-art results for word embedding learning. Nevertheless, we can use word embeddings learnt from k-way co-occurrences to empirically evaluate what type of information is captured by k-way co-occurrences.
- We evaluate the word embeddings created from k-way co-occurrences on multiple benchmark datasets for semantic similarity measurement, analogy detection, relation classification, and short-text classification (§5.2). Our experimental results show that, despite data sparsity, for smaller k-values such as 3 or 5, k-way embeddings outperform 2-way embeddings.

2 Related Work

The use of word co-occurrences to learn lexical semantics has a long history in NLP (Turney and Pantel 2010). Counting-based distributional models of semantics, for example, represent a target word by a high dimensional sparse vector in which the elements correspond to words that co-occur with the target word in some contextual window. Numerous word association measures such as pointwise mutual information (PMI) (Church and Hanks 1990), log-likelihood ratio (LLR) (Dunning 1993), χ^2 measure (Gale and Church 1991), etc. have been proposed to evaluate the strength of the co-occurrences between two words.

On the other hand, prediction-based approaches (Mikolov, Chen, and Dean 2013; Pennington, Socher, and Manning 2014; Collobert and Weston 2008; Mnih and Hinton 2009; Huang et al. 2012) learn low-dimensional dense embedding vectors that can be used to accurately predict the co-occurrences between words in some context. However, most prior work on co-occurrences have been limited to the consideration of two words, whereas continuous bag-of-words (CBOW) (Mikolov, Chen, and Dean 2013) model is a notable exception because it uses *all* the words in the context of a target word to predict the occurrence of the target word. The context can be modelled either as the concatenation or average of the context vectors. Models that preserve positional information in local contexts have also been proposed (Ling et al. 2015).

Co-occurrences of multiple consecutive words in the form of lexico-syntactic patterns have been successfully applied in tasks that require modelling of semantic relations between two words. For example, Latent Relational Analysis (LRA) (Turney 2006) represents the relations between word-pairs by a co-occurrence matrix where rows correspond to word-pairs and columns correspond to various lexical patterns that co-occur in some context with the word-pairs. The elements of this matrix are the co-occurrence counts between the word-pairs and lexical patterns. However, exact occurrences of n-grams tend to be sparse for large n values, resulting in a sparse co-occurrence matrix. LRA uses singular value decomposition (SVD) to reduce the dimensionality, thereby reducing sparseness.

Despite the extensive applications of word co-occurrences in NLP, theoretical relationships between co-occurrence statistics and semantic representations have been less understood. Hashimoto, Alvarez-Melis, and Jaakkola (2016) show that word embedding learning can be seen as a problem of metric recovery from log co-occurrences between words in a large corpus. Arora et al. (2016) show that log joint probability between two words is proportional to the squared sum of the ℓ_2 norms of their embeddings. However, both those work are limited to two-way co-occurrences (i.e. $k = 2$ case). In contrast, our work can be seen as extending this analysis to $k > 2$ case. In particular, we show that under the same assumptions made by Arora et al. (2016), the log joint probability of a set of k co-occurring words is proportional to the squared sum of ℓ_2 norms of their embeddings.

Averaging word embeddings to represent sentences or phrases has found to be a simple yet an accurate method (Arora, Liang, and Ma 2017; Kenter, Borisov, and de Rijke 2016) that has reported comparable performances to more complex models that consider the ordering of words (Kiros et al. 2015). For example, Arora, Liang, and Ma (2017) compute sentence embeddings as the linearly weighted sum of the constituent word embeddings, where the weights are computed using unigram probabilities. Kenter, Borisov, and de Rijke (2016) learn word embeddings such that when averaged produce accurate sentence embeddings. Such prior work hint at the existence of a relationship between the summation of the word embeddings, and the semantics of the sentence that contains those words. However, to the best of our knowledge, a theoretical connection between k-way co-occurrences and word embeddings has not been established before.

3 k-way word co-occurrences

Our analysis is based on the *random walk model* of text generation proposed by Arora et al. (2016). Let $\mathcal{V}$ be the vocabulary of words. Then, the t-th word $w_t \in \mathcal{V}$ is produced at step t by a random walk driven by a discourse vector $\boldsymbol{c}_t \in \mathbb{R}^d$. Here, d is the dimensionality of the embedding space and coordinates of $\boldsymbol{c}_t$ represent what is being talked about. Moreover, each word $w \in \mathcal{V}$ is represented by a vector (embedding) $\boldsymbol{w} \in \mathbb{R}^d$. Under this model, the probability of emitting $w \in \mathcal{V}$ at time t, given $\boldsymbol{c}_t$ given by (1).

$$\Pr[\text{emitting } w \text{ at time } t \mid \boldsymbol{c}_t] \propto \exp\left({\boldsymbol{c}_t}^\top \boldsymbol{w}\right) \qquad (1)$$

Here, a *slow* random work is assumed where $\boldsymbol{c}_{t+1}$ can be obtained from $\boldsymbol{c}_t$ by adding a small random displacement vector such that nearby words are generated under similar discourses. More specificaly, we assume that $\mid \boldsymbol{c}_{t+1} - \boldsymbol{c}_t \mid_2 \leq \epsilon_2/\sqrt{d}$ for some small $\epsilon_2 > 0$. The stationary distribution $\mathcal{C}$ of the random walk is assumed to be uniform over the unit sphere. For such a random walk, Arora et al. (2016) prove the following Lemma.

Lemma 1 (**Concentration of Partition functions** Lemma 2.1 of (Arora et al. 2016))**.** *If the word embedding vectors satisfy the Bayesian prior $\boldsymbol{v} = s\hat{\boldsymbol{v}}$, where $\hat{\boldsymbol{v}}$ is from the spherical Gaussian distribution, and s is a scalar random variable, which is always bounded by a constant, then the entire ensemble of word vectors satisfies that*

$$\Pr_{c\sim\mathcal{C}}[(1-\epsilon_z)Z \leq Z_c \leq (1+\epsilon_z)Z] \geq 1-\delta, \qquad (2)$$

for $\epsilon_z = O(1/\sqrt{n})$, and $\delta = \exp(-\Omega(\log^2 n))$, where $n \geq d$ is the number of words and Z_c is the partition function for c given by $\sum_{w\in\mathcal{V}} \exp\left(\boldsymbol{w}^\top \boldsymbol{c}\right)$.

Lemma 1 states that the partition function concentrates around a constant value Z for all c with high probability.

For d dimensional word embeddings, the relationship between the ℓ_2 norm of word embeddings $\boldsymbol{w}_i$, $||\boldsymbol{w}_i||_2$, and the joint probability of the words, $p(w_1, \ldots, w_k)$ is given by the following theorem:

Theorem 1. *Suppose the word vectors satisfy* (2)*. Then, we have*

$$\log p(w_1, \ldots, w_k) = \frac{\left|\left|\sum_{i=1}^k \boldsymbol{w}_i\right|\right|_2^2}{2d} - k\log Z \pm \epsilon. \qquad (3)$$

for $\epsilon = O(k\epsilon_z) + \widetilde{O}(1/d) + O(k^2\epsilon_2)$, where

$$Z = \sum_{(w_1,\ldots,w_k)\in\mathcal{V}^k} \sum_{c\in\mathcal{C}} \exp\left(\sum_{i=1}^k \boldsymbol{w}_i^\top \boldsymbol{c}\right). \qquad (4)$$

Note that the normalising constant (partitioning function) Z given by (4) is independent of the co-occurrences.

Proof of Theorem 1 is given in the supplementary material in (Bollegala, Yoshida, and ichi Kawarabayashi 2017). In particular, for $k = 1$ and 2, Theorem 1 reduces to the relationships proved by Arora et al. (2016). Typically the ℓ_2 norm of d dimensional word vectors is in the order of $\sqrt{d}$, implying that the order of the squared ℓ_2 norm of $\sum_{i=1}^k \boldsymbol{w}_i$ is $\mathcal{O}(d)$. Consequently, the noise level $\mathcal{O}(\epsilon)$ is significantly smaller compared to the first term in the left hand side. Later in § 5.1, we empirically verify the relationship stated in Theorem 1 and the concentration properties of the partitioning function for k-way co-occurrences.

4 Learning k-way Word Embeddings

In this Section, we propose a training objective that considers k-way co-occurrences using the relationship given by Theorem 1. By minimising the proposed objective we can obtain word embeddings that consider k-way co-occurrences among words. The word embeddings derived in this manner serve as a litmus test for empirically evaluating the validity of Theorem 1.

Let us denote the k-way co-occurrence $(w_1, \ldots, w_k) = w_1^k$, and its frequency in a corpus by $h(w_1^k)$. The joint probability $p(w_1^k)$ of such a k-way co-occurrence is given by (3). Although successive samples from a random walk are not independent, if we assume the random walk to mix fairly quickly (i.e. mixing time related to the logarithm of the vocabulary size), then the distribution of $h(w_1^k)$ can be approximated by a multinomial distribution $\mathsf{Mul}\left(\tilde{L}_k, \{p(w_1^k)\}\right)$, where $\tilde{L}_k = \sum_{w_1^k\in\mathcal{G}_k} h(w_1^k)$ and $\mathcal{G}_k$ is the set of all k-way co-occurrences. Under this approximation, Theorem 2 provides an objective for learning word embeddings from k-way co-occurrences.

Theorem 2. *The set of word embeddings $\{\boldsymbol{w}_i\}$ that minimise the objective given by* (5) *maximises the log-likelihood of k-way co-occurrences given by* (6)*. Here, C is a constant independent of the word embeddings.*

$$\sum_{w_1^k\in\mathcal{G}_k} h(w_1^k)\left(\log(h(w_1^k)) - \left|\left|\sum_{i=1}^k \boldsymbol{w}_i\right|\right|_2^2 + C\right)^2 \qquad (5)$$

$$l = \log\left(\prod_{w_1^k\in\mathcal{G}_k} p(w_1^k)^{h(w_1^k)}\right) \qquad (6)$$

The proof of of Theorem 2 is given in the supplementary of (Bollegala, Yoshida, and ichi Kawarabayashi 2017).

Minimising the objective (5) with respect to $\boldsymbol{w}_i$ and C produces word embeddings that capture the relationships in k-way co-occurrences of words in a corpus. Down-weighting very frequent co-occurrences of words has shown to be effective in prior work. This can be easily incorporated into the objective function (5) by replacing $h(w_1^k)$ by a truncated version such as $\min(h(w_1^k), \theta_k)$, where θ is a cut-off threshold, where we set $\theta = 100$ following prior work. We find the word embeddings $\boldsymbol{w}_i$ for a set of k-way co-occurrences $\mathcal{G}_k$ and the parameter C_k, by computing the partial derivative of the objective given by (5) w.r.t. those parameters, and applying Stochastic Gradient Descent (SGD) with learning rate updated using AdaGrad. The initial learning rate is set to 0.01 in all experiments. We refer to the word embeddings learnt by optimising (5) as **k-way embeddings**.

5 Experiments

We pre-processed a January 2017 dump of English Wikipedia using a Perl script[1] and used as our corpus (contains ca. 4.6B tokens). We select unigrams occurring at least 1000 times in this corpus amounting to a vocabulary of size $73,954$. Although it is possible to apply the concept of k-way co-occurrences to n-grams of any length n, for the simplicity we limit the analysis to co-occurrences among unigrams. Extracting k-way co-occurrences from a large corpus is challenging because of the large number of unique and sparse k-way co-occurrences. Note that k-way co-occurrences are however less sparse and less diverse compared to k-grams because the ordering of words is ignored in a k-way co-occurrence. Following the Apriori algorithm (Agrawal and Srikant 1994) for extracting frequent itemsets of a particular length with a pre-defined support, we extract k-way co-occurrences that occur at least 1000 times in the corpus within a 10 word window.

Specifically, we select all $(k-1)$-way co-occurrences that occur at least 1000 times and grow them by appending the selected unigrams (also occurring at least 1000 times in the corpus). We then check whether all subsets of length $(k-1)$

[1] http://mattmahoney.net/dc/textdata.html

k	no. of k-way co-occurrences
2	257,508,996
3	394,670,208
4	111,119,411
5	14,495,659

Table 1: The number of unique k-way co-occurrence with support 1000.

of a candidate k-way co-occurrence appear in the set of frequent $(k-1)$-way co-occurrences. If this requirement is satisfied, then it follows from the apriori property that the generated k-way co-occurrence must have a minimum support of 1000. Following this procedure we extract k-way co-occurrences for $k = 2, 3, 4$, and 5 as shown in Table 1.

5.1 Empirical Verification of the Model

Our proof of Theorem 1 requires the condition used in Lemma 1, which states that the partition function given by (4) must concentrate within a small range for any k. Although such concentration properties for 2-way co-occurrences have been reported before, it remains unknown whether this property holds for $k(>2)$-way co-occurrences. To test this property empirically, we uniformly randomly generate 10^5 vectors $\boldsymbol{c}$ (ℓ_2 normalised to unit length) and compute the histogram of the partition function values as shown in Figure 1 for $d = 300$ dimensional embeddings. We standardise the histogram to zero mean and unit variance for the ease of comparisons. From Figure 1, we see that the partition function concentrates around the mean for all k-values. Interestingly, the concentration is stronger for higher $k(>3)$ values. Because we compute the sum of the embeddings of individual words in (4), from the law of large numbers it follows that the summation converges towards the mean when we have more terms in the k-way co-occurrence. This result shows that the assumption on which Theorem 1 is based (i.e. concentration of the partition function for arbitrary k-way co-occurrences), is empirically justified.

Next, to empirically verify the correctness of Theorem 1, we learn $d = 300$ dimensional k-way embeddings for each k value in range $[2, 5]$ separately , and measure the Spearman correlation between $\log p(w_1, \ldots, w_k)$ and $\left|\left|\sum_{i=1}^{k} \boldsymbol{w}_i\right|\right|_2^2$ for a randomly selected 10^6 k-way co-occurrences. If (3) is correct, then we would expect a linear relationship (demonstrated by a high positive correlation) between the two sets of values for a fixed k.

Figure 2 shows the correlation plots for $k = 2, 3, 4$, and 5. From Figure 2 we see that there exist such a positive correlation in all four cases. However, the value of the correlation drops when we increase k as a result of the sparseness of k-way co-occurrences for larger k values. Although due to the limited availability of space we show results only for $d = 300$ embeddings, the above-mentioned trends could be observed across a wide range of dimensionalities ($d \in [50, 1000]$) in our experiments.

5.2 Evaluation of Word Embeddings

We re-emphasise here that our goal in this paper is *not* to propose novel word embedding learning methods but to extend the notion of 2-way co-occurrences to k-way co-occurrences. Unfortunately all existing word embedding learning methods use only 2-way co-occurrence information for learning. Moreover, direct comparisons against different word embedding learning methods that use only 2-way co-occurrences are meaningless here because the performances of those pre-trained embeddings will depend on numerous factors such as the training corpora, co-occurrence window size, word association measures, objective function being optimised, and the optimisation methods. Nevertheless, by evaluating the k-way embeddings learnt for different k values using the same resources, we can empirically evaluate the amount of information captured by k-way co-occurrences.

For this purpose, we use four tasks that have been used previously for evaluating word embeddings.

Semantic similarity measurement: We measure the similarity between two words as the cosine similarity between the corresponding embeddings, and measure the Spearman correlation coefficient against the human similarity ratings. We use Rubenstein and Goodenough (**RG**, 65 word-pairs), Miller and Charles' (**MC**, 30 word-pairs), rare words dataset (**RW**, 2034 word-pairs) (Luong, Socher, and Manning 2013), Stanford's contextual word similarities (**SCWS**, 2023 word-pairs) (Huang et al. 2012), the **MEN** dataset (3000 word-pairs) (Bruni et al. 2012), and the SimLex **SL** dataset[2] (999 word-pairs).

Word analogy detection: Using the CosAdd method, we solve word-analogy questions in the SemEval (**SE**) dataset (Jurgens et al. 2012). Specifically, for three given words a, b and c, we find a fourth word d that correctly answers the question *a to b is c to what?* such that the cosine similarity between the two vectors $(\boldsymbol{b} - \boldsymbol{a} + \boldsymbol{c})$ and $\boldsymbol{d}$ is maximised.

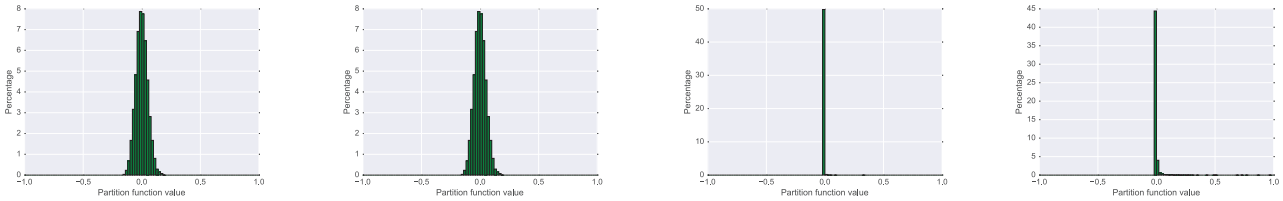
Relation classification: We use the DIFFVEC **DV** (Vylomova et al. 2016) dataset containing 12,458 triples of the form $(\text{relation}, \text{word}_1, \text{word}_2)$ covering 15 relation types. We train a 1-nearest neighbour classifier, where for each target tuple we measure the cosine similarity between the vector offset for its two word embeddings, and those of the remaining tuples in the dataset. If the top ranked tuple has the same relation as the target tuple, then it is considered to be a correct match. We compute the (micro-averaged) classification accuracy over the entire dataset as the evaluation measure.

Short-text classification: We use four binary short-text classification datasets: Stanford sentiment treebank (**TR**)[3] (903 positive test instances and 903 negative test instances), movie reviews dataset (**MR**)[4] (5331 positive instances and 5331 negative instances), customer reviews dataset (**CR**) (Hu and Liu 2004) (925 positive instances

[2] https://www.cl.cam.ac.uk/~fh295/simlex.html

[3] http://nlp.stanford.edu/sentiment/treebank.html

[4] www.cs.cornell.edu/people/pabo/movie-review-data/



(a) 2-way co-occurrences (b) 3-way co-occurrences (c) 4-way co-occurrences (d) 5-way co-occurrences

Figure 1: Histogram of the partitioning function for randomly chosen $10,000$ context vectors.

k	RG	MC	WS	RW	SCWS	MEN	SL	SE	DV	TR	MR	CR	SUBJ
2	**78.63**	79.17	59.68	41.53	**57.09**	70.42	**34.76**	37.21	75.34	72.43	68.38	79.19	82.20
≤ 3	77.51	**79.92**	59.61	**41.58**	56.69	$\mathbf{70.92}^*$	34.65	**37.42**	$\mathbf{75.96}^*$	$\mathbf{72.92}^*$	**68.71**	$\mathbf{79.52}^*$	82.35
≤ 4	75.85	72.66	59.75	41.23	56.74	70.32	34.51	37.01	74.92	72.37	67.87	78.18	82.25
≤ 5	75.19	74.63	$\mathbf{60.54}^*$	40.84	56.92	70.50	34.67	37.21	74.76	72.21	68.48	77.18	$\mathbf{82.60}^*$

Table 2: The results on word similarity, analogy, relation classification and short-text classification tasks reported by the word embeddings learnt using k-way co-occurrences for different k values.

and 569 negative instances), and the subjectivity dataset (**SUBJ**) (Pang and Lee 2004) (5000 positive instances and 5000 negative instances). Each review is represented as a bag-of-words and we compute the centroid of the embeddings for each bag to represent the review. Next, we train a binary logistic regression classifier using the train portion of each dataset, and evaluate the classification accuracy using the corresponding test portion.

Statistical significance at $p < 0.05$ level is evaluated for correlation coefficients and classification accuracies using respectively Fisher transformation and Clopper-Pearson confidence intervals.

Learning k-way embeddings from k-way co-occurrences for a single k value results in poor performance because of data sparseness. To overcome this issue we use all co-occurrences equal or below a given k value when computing k-way embeddings for a given k. Training is done in an iterative manner where we randomly initialise word embeddings when training 2-way embeddings, and subsequently use $(k-1)$-way embeddings as the initial values for training k-way embeddings. The performances reported by 300 dimensional embeddings are shown in Table 2, where best performance in each task is shown in bold and statistical significance over 2-way embeddings is indicated by an asterisk.

From Table 2, we see that for most of the tasks the best performance is reported by $k(\geq 2)$-way embeddings and not $k = 2$-way embeddings. In some of the larger datasets, the performances reported by $k \leq 3$ (for **MEN**, **DV**, and **CR**) and $k \leq 5$ way embeddings (for **WS** and **SUBJ**) are significantly better than that by the 2-way embeddings. This result supports our claim that $k(> 2)$-way co-occurrences should be used in addition to 2-way co-occurrences when learning word embeddings.

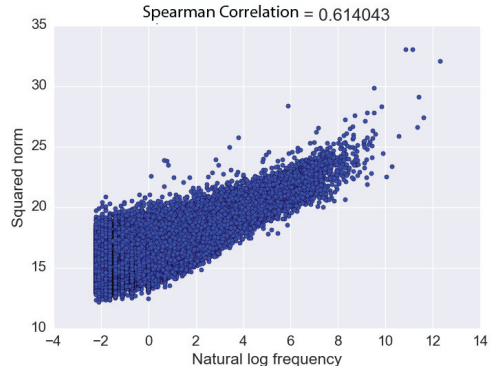
Prior work on relational similarity measurement have shown that the co-occurrence context between two words provide useful clues regarding the semantic relations that exist between those words. For example, the the phrase *is a large* in the context ***Ostrich** is a large **bird*** indicates a hypernymic relation between *ostrich* and *bird*. The two datasets **SE** and **DV** evaluate word embeddings for their ability to represent semantic relations between two words. Interestingly, we see that $k \leq 3$ embeddings perform best on those two datasets.

Text classification tasks require us to understand not only the meaning of individual words but also the overall topic in the text. For example, in a product review individual words might have both positive and negative sentiments but for different aspects of the product. Consequently, we see that $k \leq 3$ embeddings consistently outperform $k = 2$ embeddings on all short-text classification tasks. By consider all co-occurrences for $k \leq 5$ we see that we obtain the best performance on the **SUBJ** dataset.

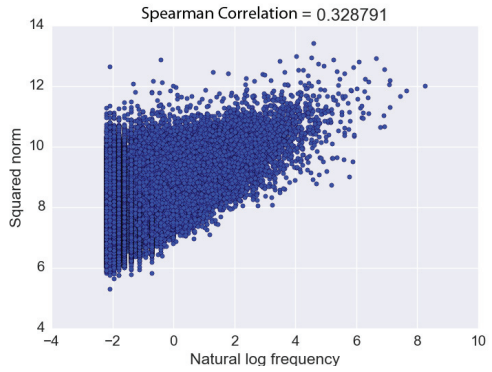
For the word similarity benchmarks, which evaluate the similarity between two words, we see that 2-way co-occurrences are sufficient to obtain the best results in most cases. A notable exception is **WS** dataset, which has a high portion of related words than datasets such as **MEN** or **SL**. Because related words can co-occur in broader contextual window and with various words, considering a $k \leq 5$ way co-occurrences seem to be effective.

5.3 Effect of Data Sparseness

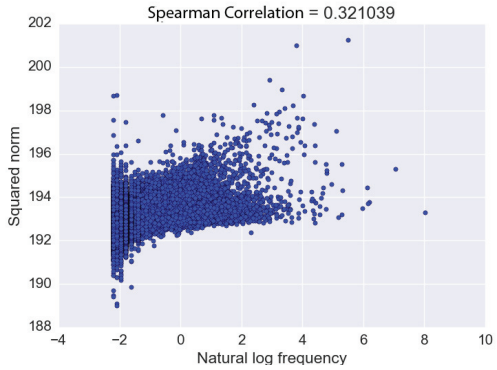
Overall, Table 2 shows that although some information regarding word semantics can be captured using k-way co-occurrences, the approach runs into data sparseness issues for high k values. Among the different k values, $k = 3$ appears to be the case that shows some improvement over $k = 2$ case at least in a subset of the different evaluation tasks when initialised with pre-trained 2-way embeddings. Learning accurate k-way embeddings for larger k values overcoming the data sparsity problems is a potential future research direction for us. Increasing the size of the dataset and decreasing the support for the co-occurrences is a direct approach to reduce the data sparseness problem, but simul-



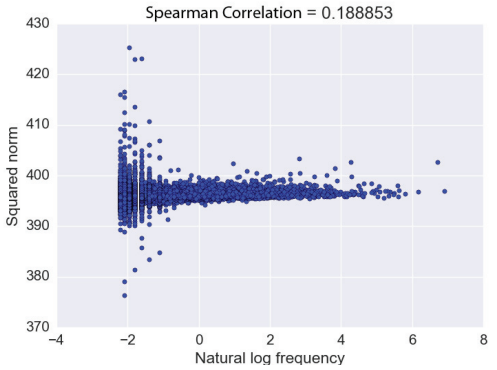
(a) 2-way co-occurrences



(b) 3-way co-occurrences



(c) 4-way co-occurrences



(d) 5-way co-occurrences

Figure 2: Correlation between the squared ℓ_2 norms of the sum of the k-way embeddings and the natural log frequency of the corresponding k-way co-occurrences are shown for different k values.

taneously increases the computational cost. In this Section, we empirically study the effect of varying the co-occurrence support while keeping the corpus size fixed. We limit our analysis to $k = 2$ and $k = 3$ -way embeddings, which appear to be the most effective according to the experimental results in the previous section.

Support	500	2000
Vocabulary	114,599	47,533
$k = 2$	307,042,130	20,382,664
$k = 3$	424,150,397	359,161,692

Table 3: The number of k-way co-occurrences for $k = 2$ and $k = 3$ settings under two different support thresholds.

We train 2-way and 3-way embeddings for co-occurrences extracted under two different support levels, 500 and 2000. Specifically, we limit the vocabulary to the unigrams that occur at least L times in the corpus, and generate all possible 2-way and 3-way co-occurrences for those unigrams. The number of extracted k-way co-occurrences are shown in Table 3. From Table 3, we see that in particular the number of 3-way co-occurrences increase significantly for 3-way co-occurrences, whereas the increase of 2-way co-occurrences is much less when we lower the support level. This shows the computational challenges involved in decreasing the support level L as a solution to overcome the sparseness in co-occurrences. We then train 2-way and 3-way embeddings from the extracted co-occurrences. Unlike in the previous Section, we do not initialise 3-way embeddings using pre-trained 2-way embeddings here because doing so would not demonstrate any issues in 3-way embeddings due to data sparseness issues, if they exist.

Performance of the 2-way and 3-way embeddings trained under $L = 500$ and $L = 2000$ support levels on different benchmark datasets is shown in Table 4. We see that 3-way embeddings perform poorly compared to 2-way embeddings when we do not initialise 3-way embeddings using 2-way embeddings. This result justifies our proposal to use all k-way co-occurrences below a particular k value when learning k-way embeddings. Moreover, we see that lowering the support threshold usually *decreases* the improve performance in all semantic similarity benchmark datasets except in **RG**. On the other hand in both relational similarity datasets **SE** and **DV** we see that lowering the support threshold improves the performance of both 2-way and 3-way embeddings. For the short-text classification datasets, we see that in **TR** and **MR** datasets, lowering the support thresh-

	RG	MC	WS	RW	SCWS	MEN	SL	SE	DV	TR	MR	CR	SUBJ
Support = 500													
k=2	78.75	80.65	52.29	36.48	54.86	65.10	31.75	37.17	78.45	75.50	71.71	75.83	86.15
k=3	45.04	39.89	38.04	13.74	43.29	41.13	19.96	34.57	68.57	70.18	64.49	74.49	80.30
Support = 2000													
k=2	76.14	80.38	63.47	41.98	59.01	71.56	35.45	36.79	73.59	75.78	71.01	78.18	85.55
k=3	34.47	41.81	41.40	27.41	46.67	44.57	21.03	33.83	64.12	68.75	64.12	75.17	81.35

Table 4: Performance of 2-way and 3-way embeddings trained using co-occurrences extracted under two different support levels.

first	second	2-way	3-way	Human	Type
giraffe	harbor	0.468	0.117	0.020	unrelated
car	hawk	0.563	0.304	0.200	
competition	relation	0.630	0.304	0.200	
happy	posted	0.658	0.386	0.280	
professor	cucumber	0.518	0.130	0.082	
museum	swim	0.516	0.222	0.120	
white	woman	0.746	0.491	0.050	collocations
computer	expert	0.717	0.523	0.071	
heart	surgery	0.702	0.447	0.089	
salt	water	0.745	0.564	0.112	
secret	weapon	0.708	0.497	0.080	
movie	star	0.779	0.557	0.190	
absence	presence	0.881	0.713	0.018	antonyms
easy	difficult	0.871	0.786	0.037	
short	long	0.920	0.873	0.104	
agree	argue	0.843	0.674	0.056	
bottom	top	0.832	0.704	0.049	
accept	reject	0.846	0.698	0.063	
south	north	0.951	0.871	0.206	

Table 5: Qualitatively comparing the 2-way and 3-way embeddings on the similarity prediction task.

old improves performance of 3-way embeddings, while decreases its performance in **CR** and **SUBJ** datasets. On the other hand, the performance of 2-way embeddings improves with the lower support in **MR** and **SUBJ** datasets.

This result shows that for semantic similarity benchmarks, lowering support threshold does not help, whereas it significantly helps for the relational similarity/classification tasks. This trend is particularly prominent for 3-way embeddings than for 2-way embeddings. Lowering the support threshold is not always a good solution to reduce data spareness because it also increases the number of unique k-way co-occurrences, thereby introducing many low-frequent k-way co-occurrences to the long-tail of the co-occurrence distribution making training difficult.

5.4 Qualitative Evaluation

Our quantitative experiments revealed that 3-way embeddings are particularly better than 2-way embeddings in multiple tasks. To qualitatively evaluate the difference between 2-way and 3-way embeddings, we conduct the following experiment.

First, we combine all word pairs in semantic similarity benchmarks to create a dataset containing 8483 word pairs with human similarity ratings. We normalise the human similarity ratings in each dataset separately to $[0, 1]$ range by subtracting the minimum rating and dividing by the difference between maximum and minimum ratings. The purpose of this normalisation is to make the ratings in different benchmark datasets comparable. Next, we compute the cosine similarity between the two words in each word pair using 2-way and 3-way embeddings separately. We then select word pairs where the difference between the two predicted similarity scores are significantly greater than one standard deviation point. This process yields 911 word pairs, which we manually inspect and classify into several categories.

Table 5 shows some randomly selected word pairs with their predicted similarity scores scaled to 0.5 means and 1.0 variance, and human ratings given in the original benchmark dataset in which the word pair appears. We found that 2-way embeddings assign high similarity scores for many unrelated word pairs, whereas by using 3-way embeddings we are able to reduce the similarity scores assigned to such unrelated word pairs. Words such as *giraffe*, *car* and *happy* are highly frequent and co-occur with many different words. Under 2-way embeddings, any word that co-occur with a target word will provide a semantic attribute to the target word. Therefore, unrelated word pairs where at least one word is frequent are likely to obtain relatively higher similarity score under 2-way embeddings.

We see that the similarity between two words in a collocation are overly estimated by 2-way embeddings. The two words forming a collocation are not necessarily semantically similar. For example, *movie* and *star* do not share many attributes in common. 3-way embeddings correctly assigns lower similarity scores for such words because many other words co-occur with a particular collocation in different contexts.

We observed that 2-way embeddings assign high similarity scores for a large number of antonym pairs. Prior work on distributional methods of word representations have shown that it is difficult to discriminate between antonyms and synonyms using their word distributions (Mohammad et al. 2013). Scheible, Schulte im Walde, and Springorum (2013) show that by restricting the contexts we use for building such distributional models, by carefully selecting context features such as by selecting verbs it is possible to overcome this problem to an extent. Recall that 3-way co-occurrences require a third word co-occurring in the contexts that contain the co-occurrence between two words we are interested in measuring similarity. Therefore, 3-way embeddings by

definition impose contextual restrictions that seem to be a promising alternative for pre-selecting contextual features. We plan to explore the possibility of using 3-way embeddings for discriminating antonyms in our future work.

6 Conclusion

We proved a theoretical relationship between the joint probability of more than two words and their embeddings and learnt word embeddings using k-way co-occurrences. Our results validated the derived relationship and show that we can learn better word embeddings for tasks that require contextual information by considering 3-way co-occurrences.

Acknowledgement

This work was supported by JST ERATO Grant Number JPMJER1201, Japan.

References

Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules in large databases. In *Proc of VLDB*, 487–499.

Arora, S.; Li, Y.; Liang, Y.; Ma, T.; and Risteski, A. 2016. A latent variable model approach to pmi-based word embeddings. *TACL* 4:385–399.

Arora, S.; Liang, Y.; and Ma, T. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proc. of ICLR*.

Bollegala, D.; Yoshida, Y.; and ichi Kawarabayashi, K. 2017. Using k-way Co-occurrences for Learning Word Embeddings. *ArXiv e-prints*.

Bruni, E.; Boleda, G.; Baroni, M.; and Tran, N. K. 2012. Distributional semantics in technicolor. In *Proc. of ACL*, 136–145.

Church, K. W., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16(1):22 – 29.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*, 160 – 167.

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19:61–74.

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

Gale, W. A., and Church, K. W. 1991. A program for aligning sentences in bilingual corpora. In *Proc. ACL*, 177–184.

Hashimoto, T.; Alvarez-Melis, D.; and Jaakkola, T. 2016. Word embeddings as metric recovery in semantic spaces. *Transactions of the Association for Computational Linguistics* 4:273–286.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proc. KDD*, 168–177.

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *Proc. of ACL*, 873–882.

Jurgens, D. A.; Mohammad, S.; Turney, P. D.; and Holyoak, K. J. 2012. Measuring degrees of relational similarity. In *Proc. of SemEval*.

Kenter, T.; Borisov, A.; and de Rijke, M. 2016. Siamese cbow: Optimizing word embeddings for sentence representations. In *Proc. of ACL*, 941–951.

Kiros, R.; Zhu, Y.; Salakhutdinov, R.; Zemel, R. S.; Torralba, A.; Urtasun, R.; and Fidler, S. 2015. Skip-thought vectors. In *Proc. of NIPS*, 3276–3284.

Ling, W.; Dyer, C.; Black, A. W.; and Trancoso, I. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proc. of NAACL-HLT*, 1299–1304.

Luong, M.-T.; Socher, R.; and Manning, C. D. 2013. Better word representations with recursive neural networks for morphology. In *Proc. of CoNLL*.

Mikolov, T.; Chen, K.; and Dean, J. 2013. Efficient estimation of word representation in vector space. In *Proc. of ICLR*.

Mikolov, T.; tau Yih, W.; and Zweig, G. 2013. Linguistic regularities in continous space word representations. In *Proc. of NAACL-HLT*, 746 – 751.

Mnih, A., and Hinton, G. E. 2009. A scalable hierarchical distributed language model. In *Proc. of NIPS*. 1081–1088.

Mohammad, S.; Dorr, B.; Hirst, G.; and Turney, P. D. 2013. Computing lexical contrast. *Computational Linguistics* 39(3):555 – 590.

Pang, B., and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. of ACL*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: global vectors for word representation. In *Proc. of EMNLP*, 1532–1543.

Scheible, S.; Schulte im Walde, S.; and Springorum, S. 2013. Uncovering distributional differences between synonyms and antonyms in a word space model. In *Proc. of IJCNLP*, 489–497.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. Y. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proc. of NIPS*.

Turney, P. D., and Pantel, P. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Aritificial Intelligence Research* 37:141 – 188.

Turney, P. 2006. Similarity of semantic relations. *Computational Linguistics* 32(3):379–416.

Van de Cruys, T.; Poibeau, T.; and Korhonen, A. 2013. A tensor-based factorization model of semantic compositionality. In *Proc. of NAACL-HLT*, 1142–1151.

Vylomova, E.; Rimell, L.; Cohn, T.; and Baldwin, T. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relational learning. In *Proc. of ACL*, 1671–1682.