Pair Distance Distribution: a Model of Semantic Representation

Yonatan Ramni **Oded Maimon** **Evgeni Khmelnitsky**
Department of Industrial Engineering
Tel-Aviv University
Tel-Aviv, Israel
{yona5;maimon;xmel}@post.tau.ac.il

Abstract

We introduce PDD (Pair Distance Distribution), a novel corpus-based model of semantic representation. Most corpus-based models are VSMs (Vector Space Models), which while being successful, suffer from both practical and theoretical shortcomings. VSM models produce very large, sparse matrices, and dimensionality reduction is usually performed, leading to high computational complexity, and obscuring the meaning of the dimensions. Similarity in VSMs is constrained to be both symmetric and transitive, contrary to evidence from human subject tests. PDD is feature-based, created automatically from corpora without producing large, sparse matrices. The dimensions along which words are compared are meaningful, enabling better understanding of the model and providing an explanation as to how any two words are similar. Similarity is neither symmetric nor transitive. The model achieved accuracy of 97.6% on a published semantic similarity test.

1 Introduction

Semantic representation models are described by Mitchell and Lapata (2008) as belonging to one of three families, semantic networks, feature-based models and semantic spaces. Briefly, semantic networks represent words as nodes in a graph and the semantic relations between them as edges, and similarity between words is represented by the path length between them. Edges may represent a variety of different relations. Feature-based models assign a list of discrete features to each word, and similarity of words is obtained from the commonalities and differences of their feature sets. As indicated by Mitchell and Lapata (2008), semantic networks and feature-based models are often manually created by modelers, so that an effort is required to produce them, and the results are subjective.

Semantic spaces, also named DSMs (distributional semantic models) by Baroni and Lenci (2010), rely on the distributional hypothesis, that words that occur in the same contexts tend to have similar meanings (Harris, 1954). At their most basic form, word co-occurrences in various contexts are used to form feature vectors of words. DSMs are divided by Baroni and Lenci (2010) into unstructured DSMs, where word co-occurrences are counted without regard to the relation between the words, and structured DSMs, where triples of two words and particular syntactic or lexico-syntactic relations between them are counted. Various feature weighting schemes are employed, and a VSM (vector space model) is usually formed using the feature vectors (topic modeling (Griffiths et al., 2007) is a notable exception). Similarities between words are measured by distances between vectors in this multi-dimensional space, usually following a dimensionality reduction. VSMs have been successful in a number of tasks, such as word similarity and word-relation similarity tests. However, VSMs have several shortcomings. Placing all words in a multi-dimensional space, with greater distance between any two words signifying lower similarity between them, implies:

- All words have some similarity with one another.
- For any word, all other words can be ordered by their similarity to the given word.
- All pairs of words can be ordered by their similarity.
- Similarity is symmetric.

Proceedings of the 1st Workshop on Representation Learning for NLP, pages 184–192,
Berlin, Germany, August 11th, 2016. ©2016 Association for Computational Linguistics

- Transitivity - if any two words are both very similar to a third word, they cannot be very dissimilar.

- All instances of a word, whether the word is ambiguous, polysemous, or attains different meanings in different contexts, are mapped to the same position in space.

It is our view that for similarity to exist between two concepts (represented in our case by words), they must have something in common, such as a common dimension along which they have (possibly different) values. With nothing in common, two concepts bear no similarity to one another, which is not the same as having little similarity. As with relatives, some are close relatives of a person, others are more distant relations of his, and yet others are not related to him at all. Furthermore, similarity is ordinal, with numerical values, when given by human subjects, serving as an aid in ranking similarity, as is done with feelings of pain, or happiness. Let's illustrate some limitations of VSMs with examples.

Example 1: is 'bank' more similar to 'embankment' or to 'stock exchange'? 'bank' is ambiguous, so a possible solution would be to map these two different senses independently to different positions in multi-dimensional space, assuming one could automatically disambiguate them.

Example 2: is 'break' more similar to 'interrupt', 'separate', 'breach', 'burst', or 'violate'? 'break' is polysemous, with WordNet[1] listing 59 senses, which are in various degrees related to one another, just for the verb. In this case, it does not seem right to map each sense independently, as they share some meaning.

Example 3: is 'queen' more similar to 'king' or to 'woman'? The court advisers may have one opinion, and the queen's physician another. It depends on context. Similarly for 'man' vs. 'woman' and 'boy', or 'cat' vs. 'stuffed cat' and 'dog'. When VSMs are formed from a corpus, context is given by the corpus for all instances of all words as a package deal, and vectors of words are based on that context.

Example 4: How similar is 'cat' to 'submarine'? If we find nothing in common, there is no similarity, and the question doesn't seem to make sense. As with partially ordered sets, some pairs of words are related to one another, while others are not.

Example 5: Is 'flat' more similar to 'apartment', or 'chair' to 'table'? 'dog' to 'cat' or 'cow' to 'sheep'? 'fork' to 'shirt' or 'stone' to 'computer'? For some questions of this kind we may have a firm opinion, for others we may not be so sure, and some questions don't really make sense.

However, a VSM will have definite answers to all questions in the above examples, regardless of sense or context. Moreover, the symmetry and transitivity of similarity imposed by VSMs contradict human similarity judgments (Tversky, 1977). These constraints of VSMs are due to the symmetry and triangle inequality conditions that must be satisfied by any distance function. In addition, Tversky and Hutchinson (1986) show that geometric models impose an upper bound on the number of points that can share the same nearest neighbor, and that particularly for conceptual data (such as categorical ratings or associations of words), values for these exceed those possible in geometric models. It has been suggested by Tversky and Gati (1982) that "similarity may be better characterized as a feature-matching process based on the weighting of common and distinctive features than as a metric-distance function". The model we propose makes use of word co-occurrence in a corpus to build a feature-based model of semantic representation. We use sentence limits as our context window, and measure the distance (counted in the number of intervening words) between pairs of words that co-occur in sentences. It is found that for a word, its mean pmf (probability mass function) of distance with its pair-words (hence termed PDD - pair distance distribution) characterizes it across corpora, and that semantically similar words have a similar mean PDD. Given a word, its features in our model are its pair-words (those that co-occur with it within sentences), together with the frequency of pair occurrence and its PDD. Thus we take into account word order, which is disregarded by 'bag-of-words' models. As no sparse matrices are created, no dimensionality reduction is required. This makes our model scalable both in computation and in storage, but more importantly, the 'dimensions' along which we compare words are the feature words, which are clear and meaningful. This stands in contrast to the dimensions obtained following a dimensionality reduction, the meanings of which often aren't clear. As words are not mapped into high-dimensional spaces, and consequently similarities

[1] http://wordnet.princeton.edu

are not measured with distances, the shortcomings of VSMs are avoided. The rest of this paper is structured as follows: Section 2 gives details of the semantic representation model. In section 3, an algorithm for evaluating similarity, based on our model, is presented. In section 4, experiments and their results are presented. Section 5 discusses the scalability of our method, and section 6 concludes the paper.

2 Model Details

Let w_1, w_2 be two distinct word forms (hence referred to as words). Given a corpus of documents *C*, let *S* be the collection of all sentences in *C* in which both w_1, w_2 appear at least once, $S = \{s_1, s_2, \ldots s_N\}$, for a total of *N* such sentences in *S*. For each sentence $s_i \in S$, let p_1, p_2 be the positions in the sentence of the words w_1, w_2 respectively. Define the distance between the two words w_1, w_2 in the sentence s_i:

$$d(s_i) = p_2 - p_1 \tag{1}$$

For sentences of maximal length *L*,

$$|d(s_i)| \leq L - 1,\ d(s_i) \neq 0 \tag{2}$$

As an example, in the sentence ”The cat drank some milk”, ’cat’ and ’milk’ are in positions 2 and 5 respectively, and the distance between ’cat’ and ’milk’ is 3. Define S_j as the collection of sentences s_i in *S* in which $d(s_i) = j$, and $|S_j|$ as the number of sentences in S_j, then for corpus *C* and word-pair $\langle w_1, w_2 \rangle$, the probability that the distance between the words (given a sentence containing both words) is *j* is:

$$pr(d(s_i) = j) = \begin{cases} \dfrac{|S_j|}{N}, & |j| \leq L - 1, j \neq 0 \\ 0, & otherwise \end{cases} \tag{3}$$

where *N* is the total number of sentences in *S*, and *L* is the maximal sentence length. (if either w_1 or w_2 appear more than once in s_i , only the nearest pair is counted). In this manner, the Corpus pmf, termed $PDD_C(w_1, w_2)$, of word-pair $\langle w_1, w_2 \rangle$ distance in corpus *C* is obtained, for any word-pair. Given *S* and the position p_1 of word w_1 for each $s_i \in S$, it is also possible to calculate the pmf of position p_2 of w_2 in each sentence s_i , assuming random distribution of p_2. Denoting the length of sentence s_i as l_i, the probability for the position p_2 of w_2 in the sentence to be *k* is:

$$pr(p_2 = k) = \begin{cases} \dfrac{1}{l_i - 1}, & 1 \leq k \leq l_i, k \neq p_1 \\ 0, & otherwise \end{cases} \tag{4}$$

From this, it follows that the probability for any distance *j* between the two words w_1, w_2 in the sentence s_i (given p_1 and assuming random distribution of p_2) is:

$$pr(d(s_i) = j) =$$

$$\begin{cases} 1/(l_i - 1), & 1 - p_1 \leq d(s_i) \leq l_i - p_1, \\ & d(s_i) \neq 0 \\ 0, & otherwise \end{cases} \tag{5}$$

The probability for any particular distance *j* for the word-pair $\langle w_1, w_2 \rangle$ in any sentence in corpus *C*, $pr(d(S) = j)$, given that the pair occurs in the sentence, the position p_1 of w_1 in the sentence and assuming random distribution of p_2 may be obtained by averaging the probability for that distance over all sentences $s_i \in S$:

$$pr(d(S) = j) = \frac{1}{N} \sum_i pr(d(s_i) = j) \tag{6}$$

Hence another pmf for word-pair $\langle w_1, w_2 \rangle$ distance in corpus *C* is obtained. Denote this, the Random PDD, as $PDD_R(w_1, w_2)$. Whereas the Corpus pmf, $PDD_C(w_1, w_2)$, is based on the corpus data, the Random pmf, $PDD_R(w_1, w_2)$, is based on the position of w_1 in sentences where the word-pair occurs and on the sentences’ lengths, and assumes random position of w_2 in those sentences. Given a corpus *C* and a word w_1, denote the set of all sentences in *C* in which w_1 appears as S_{w_1}. The company of w_1, $Co(w_1)$ are defined as those words which appear in a number of sentences in S_{w_1} above some threshold. Calculating both Corpus and Random pmfs (as outlined above) for each word-pair $\langle w_1, w_i \rangle$, $w_i \in Co(w_1)$, it is now possible to calculate the average Corpus and Random pmfs for w_1 with its companion words, $PDD_C(w_1)$ and $PDD_R(w_1)$ respectively, by averaging the distance probabilities for all companion words weighted by their frequency of occurrence in sentences of S_{w_1},

$$PDD_C(w_1) = \frac{\sum_i (PDD_C(w_1, w_i) \times n(i))}{\sum_i n(i)}\ ,$$

$$w_i \in Co(w_1) \tag{7}$$

$$PDD_R(w_1) = \frac{\sum_i (PDD_R(w_1, w_i) \times n(i))}{\sum_i n(i)},$$
$$w_i \in Co(w_1) \tag{8}$$

where $n(i)$ is the number of sentences in which the pair of words $\langle w_1, w_i \rangle$ appears. By using KLD (Kullback-Leibler Divergence) between Corpus and Random pmfs, $D(PDD_C \| PDD_R)$, we can measure the amount of information that is lost when the Random pmf of a word w_1, $PDD_R(w_1)$, is used to approximate its Corpus pmf, $PDD_C(w_1)$, as in the case of rectangular context windows of unstructured DSMs.

$$D\left(PDD_C \| PDD_R\right) =$$
$$\sum_j \left(log\left(\frac{PDD_C(j)}{PDD_R(j)}\right) PDD_C(j) \right),$$
$$1 - L \le j \le L - 1, \;\; j \ne 0 \tag{9}$$

where *j* is the distance between w_1 and its company words (all PDDs in eq. 9 are of w_1, which has been omitted for clarity). This statistic is a property of word w_1 in corpus *C*, which indicates the amount of information contained in the order of the words that are in the context of w_1, above the information that is carried by their mere presence there. It is also possible to calculate the information in any specific position in the context, by replacing the PDD_R value for that position with the PDD_C value, multiplying the remaining probabilities by a suitable factor to keep the sum of probabilities one, and calculating KLD between PDD_C and the amended PDD_R. The difference between this KLD value and the former KLD value indicates the information for that position.

Most previous research on unstructured DSMs has used, in any one study, the same context window for all words in the corpus, as regards window size, position and weights. Even when several different window parameters have been compared,(Bullinaria and Levy, 2007; Levy et al., 1999; Lund and Burgess, 1996; Sahlgren, 2006), each window configuration was used for the whole corpus, and comparison made on basis of the final results. We know of no attempt to test different window configurations for different words. However, there is no evidence to suggest that the same window configuration is the optimal one for all words. We suggest a scheme that could be useful in determining weighting due to word order. *PPMI* (positive pointwise mutual information), which compares context window co-occurrence frequency with expected frequency, has been successfully used to weight words found in rectangular context windows (Bullinaria and Levy, 2007; Church and Hanks, 1990; Niwa and Nitta, 1994), based on their occurrence regardless of order. In our case, $PDD_R(k)$ represents the chance probability of observing one of a word's company words a distance of *k* words from it, and $PDD_C(k)$ represents the probability of this occurring in the corpus (given they co-occur in a sentence). We propose the following additional weight, $Wt(k)$, be used for word w_1, for position *k* words away from it,

$$Wt(k) =$$
$$\begin{cases} log\dfrac{PDD_C(k)}{PDD_R(k)}, & PDD_C(k) > PDD_R(k), \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where PDD_C is the Corpus pmf of word w_1, and PDD_R is its Random pmf.

3 Similarity Algorithm

Though differences between word PDDs could be used to measure similarity between words, this would lead to some of difficulties associated with VSMs, as in effect we would be measuring distances in a high-dimensional space. By treating a word's company as its features, we can compare two words based on their common pair-words. The algorithm we adopt is as follows:

- For every vocabulary word w_1:

– For every other vocabulary word w_2,:

∗ Determine words $Co(w_1, w_2)$ that are in the company of both w_1 and w_2,

$$Co(w_1, w_2) = Co(w_1) \cap Co(w_2) \tag{11}$$

∗ For every $w_f \in Co(w_1, w_2)$, calculate $PDD_C(w_1, w_f)$ and $PDD_C(w_2, w_f)$ using Eq. (3), and determine the difference between w_1 and w_2, based on w_f, as the cosine "distance" between them,

$$d_{w_f}(w_1, w_2) =$$
$$1 - \frac{PDD_C(w_1, w_f) PDD_C(w_2, w_f)}{|PDD_C(w_1, w_f)||PDD_C(w_2, w_f)|} \tag{12}$$

- $*$ For all $w_f \in Co(w_1, w_2)$, sort $d_{w_f}(w_1, w_2)$ in ascending order,

  $D_{w_1,w_2} =$

  $$\left\{ d_{w_{f_1}}(w_1, w_2), d_{w_{f_2}}(w_1, w_2) \ldots \right\} \quad (13)$$

- $*$ Set the dissimilarity of w_1 to w_2 to be the sum of the first *n* elements of D_{w_1,w_2}, (*n* is an experimentally determined parameter):

  $$Diss(w_1, w_2) = \sum_{i=1}^{n} D_{w_1,w_2}(i) \quad (14)$$

  w_1 is considered to have no similarity with words w_2 that do not have common pair-words with it.

- Order all vocabulary words that have common pair-words with w_1, by increasing dissimilarity of w_1 to them.

We now have, for each vocabulary word w_1, all other vocabulary words that have common pair-words with w_1, sorted by the dissimilarity of w_1 to them. We define the dissimilarity of w_1 to any other word w_x in this list to be the rank of w_x in this sorted list, not the numerical value $Diss(w_1, w_x)$. The differences used above, in Equation (12), do not take into account the corpus frequency of the words and pairs of words, and are termed unweighted differences. Another possibility is to use weighting that expresses these frequencies. The following weighting has been shown to give good results (see Section 4):

$$\textit{Weighted Difference}_{w_f}(w_1, w_2) =$$

$$d_{w_f}(w_1, w_2) \times log\left(\frac{C_f}{C_{1,f}}\right)^d \times log(C_2)^{\frac{d}{2}} \quad (15)$$

where $d_{w_f}(w_1, w_2)$ is the unweighted difference, C_f is the corpus count of w_f, $C_{1,f}$ is the corpus count of co-occurrence of w_1 and w_f in sentences, C_2 is the corpus count of w_2, and *d* is an experimentally determined parameter. The weighted difference is then used in place of the unweighted difference in the next stages. It will be noted that $log\left(\frac{C_f}{C_{1,f}}\right)$ is non-negative, and increases as the PMI (pointwise mutual information) between w_1 and w_f decreases (the corpus count of w_1 is constant when ordering the similarity of w_1 to all other words, and is therefore omitted), so this term penalizes pair-words with low PMI to w_1. The last term penalizes, in the ordering of all corpus words by similarity of w_1 to them, words w_2 with high corpus frequency. Note that the dissimilarity $Diss(w_1, w_2)$ based on weighted differences is not symmetric with respect to w_1 and w_2. Dissimilarities based on both weighted and unweighted differences do not obey the triangle inequality, so that w_1 may be very similar to both w_2 and w_3, without requiring any minimal similarity between w_2 and w_3. Both dissimilarities also do not restrict the number of words that can share a nearest neighbor - any number of words can have w_1 as the word they are most similar to.

4 Experimental Details

4.1 Computing PDD_C

An initial experiment was carried out on 17,000 medical papers on diseases in eight different, but related, domains. The papers were returned by Google Scholar using search words relating to each domain, downloaded in pdf format and converted to text, to form eight corpora. The texts were tokenized, and lower-cased (using raw surface forms of words means different parts of speech, as well as different senses, are conflated). For vocabulary purpose, the text was filtered for stop words, numbers and any word not beginning with an alphanumeric character, and only words appearing at least 100 times were used. Sentences were delimited, and sentences with a length of over 50 words were discarded. Word-pairs were obtained from the eight corpora, for word-pair distance of up to 25 words, for word-pairs appearing in at least 10 sentences. PDD_C was calculated for all pair words, which ranged in number from 6,300 to 13,700 words for each domain, and a total number of 14,900 unique words, and 2.95 million unique pairs, for all domains combined. Fig. 1 below shows PDD_C for 'effect', 'cell', 'red' and 'show', respectively, for the eight domains superimposed. It may be seen that each word has a characteristic PDD_C across domains, and that different words have a different PDD_C.

4.2 Computing PDD_C and PDD_R

Google Scholar was again used to download articles from journals with the word 'science' in their title. Seven search words were used: 'physics', 'chemistry', 'biology', 'engineering', 'medicine', 'information' and 'environment'. Over 27,000 pdfs were downloaded, and processed as in the previous experiment. Again, only words appear-

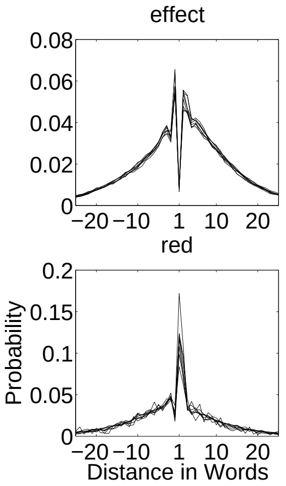

Figure 1: PDD_C for 'effect', 'cell', 'red' and 'show' across eight domains

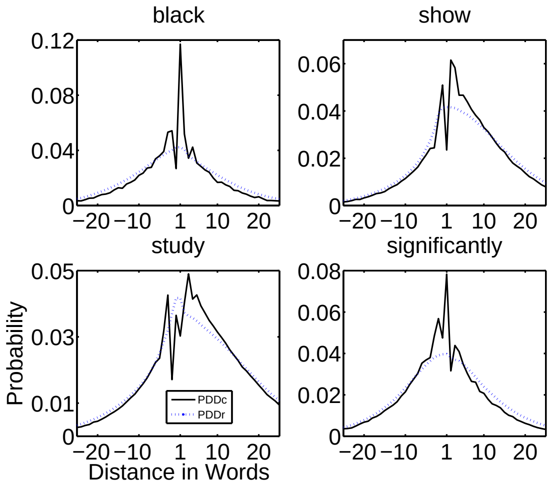

Figure 2: PDD_C and PDD_R for 'black', 'show', 'study', and 'significantly'

ing at least 100 times were used (32,341 words), pair distance of up to 25 words was considered, and PDD_C and PDD_R for all pairs appearing at least 10 times calculated. This resulted in 23,155 words and 3.9 million word pairs. Figure 2 below shows PDD_C and PDD_R for four words, 'black', 'show', 'study' and 'significantly'. The PDD_R of a word is determined by its position in sentences and the length of these sentences. The PDD_C is determined by the usage of the word with its company. It may be seen that as we move in the sentence away from the word, its PDD_C eventually follows its PDD_R from below. This is expected, as a word's company are more likely to be near it (and hence less likely to be farther away) than predicted by random chance, and because not much information is expected to be found in the order of a word's company that is not near it. However, the more interesting part is the one near the word, where the PDD_C and PDD_R differ considerably. Each word has a distinctive pattern, from which we may learn the amount of information in the order of the words around it, as detailed in Section 2. For 'black' and 'significantly', it is the following word that holds the most information, for 'show' it is the second word following, and for 'study' it is the third word following, with positions immediately around it held by company words less often than by chance (presumably because they are held by function words). This behavior is probably affected by each word's most frequent part-of-speech usage in the corpus, for example, 'black' as an adjective is likely to have a related content word following it.

Fig. 3 compares the PDD_C of 10 adjectives, 5 colors ('black', 'red', 'blue', 'white', 'green') and 5 size adjectives ('huge', 'big', 'great', 'large', 'enormous'). The top row shows the five colors and five sizes PDD_C. The bottom row shows on the left the mean color and size PDD_C, and on the right all color and size PDD_C. It may be seen that though color and size PDD_C are similar, they differ, particularly in positions nearest the word. Clustering the ten PDD_C into two clusters, using kmeans clustering and cityblock distance, separates them correctly into colors and sizes. This illustrates that (at least in this case) the PDD_C is related also to their semantic content, and not only to their part-of-speech.

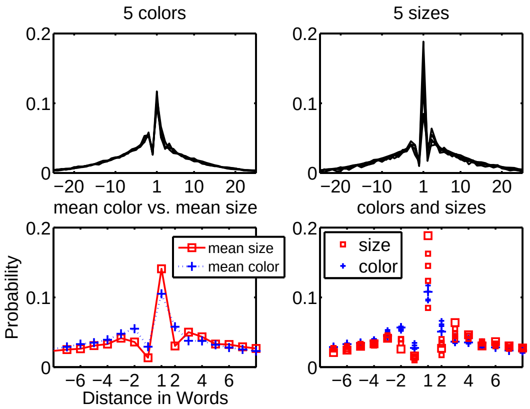

Figure 3: PDD_C Comparison for Color and Size Adjectives

4.3 Computing Weights for Positions in the Context Window

Using eq. 10, weights were calculated for positions in the context window of all words that have pairs. These weights are meant to reflect the information in the order of the words, given that they occur in the window (a fact that by itself carries information). It turns out that these weights differ from one word to another. Fig. 4.3 shows on the top row PDD_C vs. PDD_R for 'red' and 'year'. The bottom row shows the weights calculated for their context windows, together with the weights for another, semantically similar word ('black' and 'day' respectively). Weights not shown are zero. The adjectives get the greatest weight for the following word, and zero weight for the preceding word. The nouns 'year' and 'day' get zero weight for both the preceding and the following words, with 'year' getting the greatest weights for the fourth word preceding and the third word following, and 'day' for the second word preceding and the third word following. The nouns also have weights for wider contexts than the adjectives. This example shows that different words have different optimal context windows, both in width and in weight, as regards the information in word order.

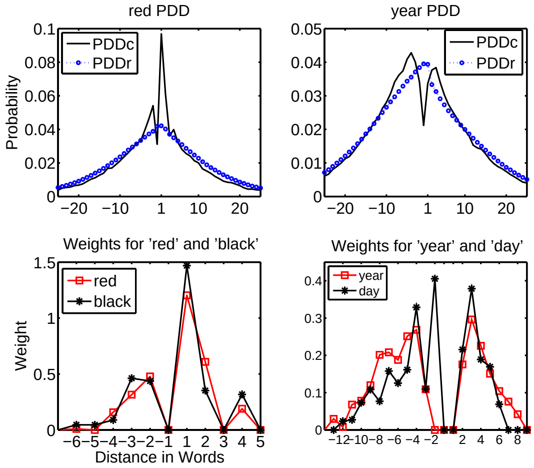

Figure 4: Context Window Weights

4.4 TOEFL Test

In order to increase vocabulary size, the ukWaC corpus[2] holding over a billion words in documents crawled from the internet was used (Baroni et al., 2009). The same processing as in the previous experiments was applied. This resulted in a vocabulary of 136,812 words with a frequency of at least 100 in the corpus. The TOEFL synonym dataset (Landauer and Dumais, 1997) consists of 80 question words, for each of which 4 answer words are given, and the task is to select the answer word most similar to the question word. The test contains 391 unique words, 7 of which were missing in our vocabulary. Three questions had one wrong answer word missing, and these were attempted without the missing word. One question had all but one wrong answer word missing, and was marked as wrong. For each TOEFL word, word-pairs that appear in at least 10 sentences in the corpus were extracted. The method we used to select the correct answer is by ordering the answer words by decreasing similarity of the question word to them as outlined in section 3, and choosing as correct the top word. Cosine distance was used, and values for *n*, the number of common feature words, from 1 to 50 were evaluated. Both unweighted and weighted differences were calculated. A grid search was performed for the best combination of *n* and *d*, and the values of 5 and 3.5 respectively give a result of 86%. However any combination of values for *n* in the range 3-9 and for *d* in the range 2-5, give a result of 80% and above. Fig. 5 shows the results with both weighted (d = 3.5) and unweighted differences, as a function of the number of feature words used. It will be noticed that with the weighted differences, it takes 3-5 feature words to get optimal results. Better results on the TOEFL test have been achieved by (Rapp, 2003; Han, 2014; Pilehvar et al., 2013; Turney et al., 2003; Bullinaria and Levy, 2012), ranging from 92.5 to 100%. Han (2014) and Turney et al. (2003) are hybrid approaches, combining the results of several methods. Pilehvar et al. (2013) relies on WordNet[3] for sense inventory of words, and uses a substantially different version of the test. Both Bullinaria and Levy (2012) and Rapp (2003), after obtaining a vocabulary from a corpus, artificially introduce into their data out-of-vocabulary TOEFL words, which would not be possible for open-ended questions.

4.5 Distance Test

This experiment uses the same corpus and the same processing as the previous experiment. The

[2] http://wacky.sslmit.unibo.it

[3] http://wordnet.princeton.edu

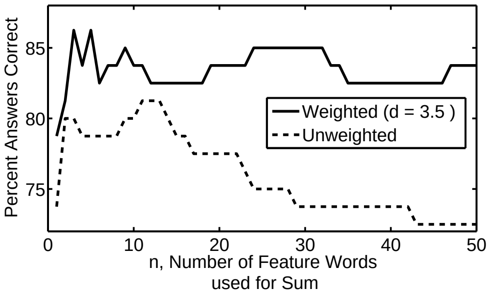

Figure 5: TOEFL Test Results

distance comparison test (Bullinaria and Levy, 2007), for which the data has kindly been made available by the authors on their website, is also similar to the TOEFL test. This test consists of 200 pairs of semantically related words. For each pair, one word is set as the question word. The other pair word, the answer word, is included in a list with 10 additional words, chosen at random from the other pairs. The task is to sort this list in order of decreasing similarity to the question word, and points are awarded according to the position of the answer word in this list (1 point for 1st position, 0.9 for 2nd, etc.). Using the same method as in the previous experiment, results for unweighted and weighted distance (*d* = 3) are shown in fig. 6 below. A value of 97.6% is obtained for this weight, *d*=3, and for *n* with a value of 7 or 8. However any combination of values for *d* in the range 2-4.5 and *n* in the range 4-11 yields a result of over 97%.

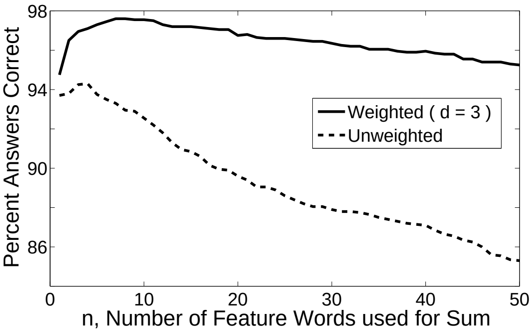

Figure 6: Distance Test Results

5 Scalability

The model size is governed by the number *P* of distinct word-pairs that occur in sentences of the corpus, which is related to the vocabulary size, *V*, which in turn depends on *N*, the number of tokens in the corpus. For the ukWac corpus, $V \sim cN^{0.53}$, and for pairs with a distance of up to 6 words, that appear at least 5 times, *P* grows as $N^{0.80}$. This shows that the model size is scalable with corpus size. With *p* as the maximal pair distance used, the complexity of building the *PDD* model is bounded by $2pN$, and is therefore $O(N)$, again scalable with corpus size. In order to arrange all vocabulary words *V* by similarity of a single word *w* to them, it is necessary to find the best *n* features each vocabulary word has in common with *w*, and calculate the similarity based on the sum of differences for the *n* features. Doing this for all vocabulary words (i.e. arranging all words in order of similarity of every word to them) is governed by *C*, the number of common features all vocabulary words have with all other vocabulary words. For the ukWac corpus, (again for pairs with a distance of up to 6 words, that appear at least 5 times), *C* grows as $N^{1.21}$. While this grows faster than corpus size, it is feasible to calculate this for the ukWac corpus. For larger corpora it may be necessary to limit, for each word, the calculation of its similarity to words that have a number of common pair-words with it above some threshold.

6 Discussion and Conclusions

We have presented a novel model of semantic representation, that is scalable and does not suffer from the shortcomings of VSMs. Two words that have no common features are not considered similar, and are not given a similarity value. Similarity is not symmetric, in accordance with human similarity judgments. Similarity is not transitive, so that a given word may be similar to two other words, to each with different senses of the given word, or in different contexts, without necessitating any similarity between the two other words. The features with which the similarity between a pair of words is evaluated are clear and meaningful - the common pair-words. The model makes it possible to select which features of a word to use when evaluating similarity, thus enabling one to take into account different senses and different contexts of a word. The model has been shown to work well on word similarity tasks. Further work could use the model for word disambiguation tasks, as different senses of a word are expected to have different PDDs. The current work has used pair distance distribution, and compared words based on their common features. Future work could use triplet distance distribution, and take into account distinctive word features as well as the common features.

References

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526.

John A Bullinaria and Joseph P Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. *Behavior research methods*, 44(3):890–907.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum. 2007. Topics in semantic representation. *Psychological review*, 114(2):211–244.

Lushan Han. 2014. *Schema free querying of semantic data*. Ph.D. thesis, University of Maryland.

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

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

Joseph P Levy, John A Bullinaria, and Malti Patel. 1999. Explorations in the derivation of word co-occurrence statistics. *South Pacific Journal of Psychology*, 10(01):99–111.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *ACL-08: HLT*, pages 236–244.

Yoshiki Niwa and Yoshihiko Nitta. 1994. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 304–309. Association for Computational Linguistics.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1341–1351.

Reinhard Rapp. 2003. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the ninth machine translation summit*, pages 315–322.

Magnus Sahlgren. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, Stockholm University.

Peter Turney, Michael L Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, pages 482–489.

Amos Tversky and Itamar Gati. 1982. Similarity, separability, and the triangle inequality. *Psychological review*, 89(2):123–154.

Amos Tversky and J Hutchinson. 1986. Nearest neighbor analysis of psychological spaces. *Psychological review*, 93(1):3–22.

Amos Tversky. 1977. Features of similarity. *Psychological review*, 84(4):327–352.