Empirical analysis of the effect of dimension reduction and word order on semantic vectors

Laurianne Sitbon
Queensland University of Technology, Australia
laurianne.sitbon@qut.edu.au

Peter D. Bruza
Queensland University of Technology, Australia
p.bruza@qut.edu.au

Christian Prokopp
Queensland University of Technology, Australia
prokopp@gmail.com

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

The aim of this paper is to provide a comparison of various algorithms and parameters to build reduced semantic spaces. The effect of dimension reduction, the stability of the representation and the effect of word order are examined in the context of the five algorithms bearing on semantic vectors: Random projection (RP), singular value decomposition (SVD), non-negative matrix factorization (NMF), permutations and holographic reduced representations (HRR). The quality of semantic representation was tested by means of synonym finding task using the TOEFL test on the TASA corpus. Dimension reduction was found to improve the quality of semantic representation but it is hard to find the optimal parameter settings. Even though dimension reduction by RP was found to be more generally applicable than SVD, the semantic vectors produced by RP are somewhat unstable. The effect of encoding word order into the semantic vector representation via HRR did not lead to any increase in scores over vectors constructed from word co-occurrence in context information. In this regard, very small context windows resulted in better semantic vectors for the TOEFL test.

Keywords: Semantic spaces; Wordspace models; SVD; Random Indexing; Non-negative matrix factorization; Holographic Reduced Representation; permutations

1. Introduction

In computational linguistics, information retrieval and applied cognition, words are often represented as vectors in a high dimensional space computed from a corpus of text. A recent survey of existing approaches to such representations [31] presents a wide range of applications of them, such as word sense disambiguation, context-sensitive spelling correction or query expansion. In a variety of studies from cognitive science there have been encouraging results using such representations to

replicate human word association norms, for example, semantic association (See, for example, [19], [14], [18], [32]). Such models are often referred to as "semantic space" models. These studies provide evidence that the vector representations within semantic space models do capture the semantics of words in a way which accords with those we carry around "in our heads". This opens the door to exploiting such models for developing information processing technologies which are at least partially sensitive to cognitive semantics.

One of the key features of producing effective semantic representations is dimension reduction. The initial input is a matrix, the precise details of which differ depending on which type of semantic space model is being constructed. For example, a prominent semantic space model called Latent Semantic Analysis (LSA) Simulations reported in [15] showed the original matrix resulting in a 15.8% precision on a given semantic task, whereas the dimensionally reduced matrix yielded "near maximum performance of 45-53% [...]Thus choosing the dimensionality of the reconstructed representation well approximately tripled the number of words the model learned as compared to using the dimensionality of the raw data". A crucial point here is how to chose the reduced dimensionality "well" - an issue we will return to later.

Other semantic space models, such as Hyperspace Analogue to Language (HAL) do not employ dimension reduction [6]. However, what is significant about studies with HAL is the matrix has been primed with a large amount of data - a corpus of 160 million words drawn from all Usenet news in 1995[20]. This position is echoed in a machine learning study whereby a point-wise mutual information model was more effective than a reduced semantic space model (LSA) on an automatic synonym detection task. In this case there were large amounts of data to prime the maximum likelihood estimates underpinning the probabilistic model [29] However, Turney concluded, "Perhaps the strength of LSA is that it can achieve relatively good performance with relatively little text".

What can be concluded from the above? Although, the literature does not show that dimension reduction is *necessary* to realize effective semantic vector representations, it does show in various studies that dimension reduction does benefit the expressivity of the semantic vectors, particularly when the semantic space is derived from relatively small amounts of text. Dimension reduction can capture higher order associations, and studies using LSA suggest that such associations play a role in simulating semantic tasks. Higher order associations between terms are induced because the upper space is being squeezed into a space of much lower dimensionality and individual semantic vector representations begin to blur. On this point Landauer states, "That is, if a particular stimulus, X, (e.g., a word) has been associated with some other stimulus, Y, by being frequently found in joint context (i.e., contiguity), and Y is associated with Z, then the condensation can cause X and Z to have similar representations". In other words, as a by-product of the dimension reduction, a higher-order association is being formed between X and Z. The abil-

ity of LSA to systematically capture such higher-order associations was verified in empirical studies [14].

The aim of this article is to compare a number of algorithms to compute reduced vector based word representations with respect to their ability to capture "semantics". Such representations will henceforth be termed "semantic vectors". Different weighting schemes to prime semantic vectors as well as various distance measures to compute semantic distance between vectors have been previously been extensively studied [4].

This article goes beyond this study by conducting a systematic analysis of the effect of dimension reduction on semantic representation across a number of prominent semantic space models, whilst keeping the semantic task for evaluation constant. This task, called TOEFL is a synonym determination task first employed by Landauer and Dumais [15] and subsequently employed quite extensively in a number of studies in applied cognition and computational linguistics. By explaining the various dimension reduction algorithms and subjecting them to empirical evaluation on the TOEFL task, we also aim to provide pragmatic guidelines for the choice of one or the other algorithm in practical settings.

In the literature, there are a number of facets which have been shown to have an effect on the quality of semantic vectors. In this paper, two are analyzed: (1) the effect of dimension reduction, and (2) the effect of word order.

Models such as Latent Semantic Analysis [15] and Random Indexing [24] rely on vector spaces to produce semantic vectors. In the former case, a sparse space can be dimensionally reduced by Singular Value Decomposition (SVD) and in the latter Random Projection (RP) offers a computationally inexpensive alternative to generate compact spaces from the beginning [33]. On the other hand, the semantic vectors produced by SVD are stable whereas those computed by RP are not due to the resulting semantic vector representations depending on initial random seeds. Lee and Seung [16] introduced a factorization method of non-negative matrices called NMF (non-negative matrix factorization). As is the case in SVD, the k chosen in NMF is usually much smaller than the dimensionality of the source matrix A. NMF has been shown to be effective and straightforward to use in text mining tasks [34]. In the light of foregoing background, the first question to be addressed is which dimension reduction technique produces semantic vectors of better quality? Secondly, what is the stability of the semantic vectors computed by RP?

Most semantic vector representations are based on "bags of words". The basis of the claim of such representations being able to capture meaning is as follows: The vector $\vec{w}$ corresponding to word w encodes co-occurence information of words co-occuring with w in context and therefore the vector can be viewed as a computational manifestation of Firth's famous quote, "You shall know a word by the company it keeps" [7]. Recently, the BEAGLE model has redressed this by using holographic reduced representations to encode word order as well as co-occurrence (context) information into the semantic vector [12]. More recently [25] have introduced a derived model for encoding word order based on random permutations.

This raises the question as to how much word order information contributes to the quality of the semantic representation of words.

The structure of this article is as follows. In the next section five semantic vector models will be described. Thereafter the models will be compared using the TOEFL test as a means for comparing the respective semantic vector representations in relation to the questions just raised.

2. Semantic space models

The models chosen for analysis are practically oriented models in the sense they use few bells and whistles, such a lemmatization etc. These are models which are more or less "off the shelf" and can be pointed at an arbitrary corpus of text with a minimum of fuss. All models have a pedigree in the literature and they differ in the details how they produce semantic vectors.To the best of the authors' knowledge, these models have not been systematically compared.

The basis of the semantic space is an $n \times m$ matrix denoted by S. The value of the cell $S[i, j]$ reflects the strength of occurrence of word i in context j. A context could be another word, a fragment of text, or even a whole document. Various weighting schemes can be applied to the values in S, the most basic one consisting in the the number of times a particular word appears in the context of another word (square context window). Algorithm 1 explains the process for a text $Text$ and a context window of size $window\_size$ where the matrix S is initialized with zeros.

Let's consider a small corpus comprising the following two sentences to prime the model S:

s1 : After school, the kid left his book on the table
s2 : Coming back from school, the child left his book on his desk

For the purposes of this example, the nouns are assumed information bearing resulting in six words ($n = 6$). A window size of 10 is assumed. The resulting semantic space S is depicted in Figure 1. Even with this straightforward semantic space model, such strengths can be viewed as quantifying Firth's intuition mentioned above. In addition, those words more often appearing in the context windows around "book" contribute more to its semantics.

2.0.1. *Pointwise Mutual Information weighting*

The frequency based approach for modelling collocations between words just presented includes an inherent bias. Because frequent terms have more chance of co-occurring together than rare terms, one can consider that the frequency of co-occurrence of two terms should be relative to the individual frequencies of the words in the corpus. The point-wise mutual information (PMI) measure, described in [21], provides a formula to weight co-occurrences this way. Given two terms a and b, Eq. (1) provides the weight of the co-occurrence a, b.

	school	kid	book	table	child	desk
school		1	2	1	1	1
kid	1		1	1	0	0
book	2	1		1	1	1
table	1	1	1		0	0
child	1	0	1	0		1
desk	1	0	1	0	1	

Table 1.

Algorithm 1

```
for all position in Text do
  term1 = Text(position)
  for i = 1 to context_size do
    term2 = Text(position-i)
    S(term1, term2) += 1
    term3 = Text(position+i)
    S(term1,term3) += 1
  end for
end for
```

$$PMI(a,b) = log\frac{p(a,b)}{p(a)p(b)} = \log\frac{p(a|b)}{p(a)} \tag{1}$$

A simplification of the PMI by removing the log has shown consistently good performance across both the TOEFL synonym task and semantic categorization [4], hence we will use this simplification as a baseline.

In addition, the semantic space is used only to compute distances between terms. The common fraction terms (including the total number of terms in the corpus) can be removed from the weighting, resulting in the weighting of Eq. (2) where $freq$ is the number of occurrences of a term or a couple of terms.

$$w(a,b) = \frac{p(a|b)}{p(a)} = \frac{\frac{freq(a,b)}{freq(b)}}{\frac{freq(a)}{|corpus|}} = \frac{freq(a,b)}{freq(a)freq(b)} \tag{2}$$

2.0.2. *HAL representation*

Hyperspace Analogue to Language (HAL) [19] adopts a different approach to weighting term co-occurrences. Also referred to as triangular window weighting, HAL takes into account the distances between the words into the context window.

When two words $w1$ and $w2$ co-occur within a certain context window, their current score is incremented with the inverse of their distance in the sliding window as detailed in Algorithm 2.

Algorithm 2

```
for all position in Text do
  term1 = Text(position)
  for i = 1 to context_size do
    term2 = Text(position-i)
    S(term1, term2) += 1/i
    term3 = Text(position+i)
    S(term1,term3) += 1/i
  end for
end for
```

2.1. *Dimension reduction of Semantic Space*

2.1.1. *Singular Value Decomposition*

SVD is a powerful result from linear algebra [8] which has been widely applied in areas such as text mining, image analysis, information retrieval and cognitive science to name but a few. The SVD theorem states that any $n \times m$ matrix S with rank r can be decomposed into three matrices: $S = UDV^T$ where U and V are unitary $n \times r$ and $m \times r$ matrices respectively. The matrix D is an $r \times r$ diagonal matrix whose values are monotonically decreasing singular values of S. The columns of U and V are the eigenvectors of SS^T and S^TS respectively.

Dimension reduction is performed by taking only the first k eigenvectors ($k < m$) and singular values to approximate S by $S_k = U_k D_k V_k^T$, where U_k and V_k are $n \times k$ and $m \times k$ matrices composed of the first k columns of U and V respectively. The Eckart-Young theorem states that S_k is the closest rank-k approximation to S in the sense of the matrix 2-norm. Stated formally, $S_k = \mathbf{min}_{\mathbf{rank}(B)=k} ||S - B||_2$.

An intuition which can be ascribed to the Eckhart-Young theorem is that SVD tries to capture as much of the variation in the data in S within the reduced number of dimensions specified by k. One of the assumptions behind semantic space models is that words with similar meanings will tend to cluster together in the space.

In the widely employed Latent Semantic Analysis (LSA) model [15] (also known as Latent Semantic Indexing in the field of information retrieval), the initial matrix, the semantic space S is approximated by a matrix S_k, with a value of $k = 300$. This value was determined by running simulations of human synonym detection using the TOEFL test. In the experiments to follow, we will provide a much finer grain of analysis of the value k than is reported in [15], where k was manipulated in increments of 50. Because we are interested in reducing the initial representations,

the re-multiplication of the reduced matrices doesn't make sense since it leads to a matrix with the initial dimensions. At the same time, the matrix U_k represent the initial terms according to a reduced number of dimensions. Because σ contains the eigenvalues of the decomposition, we believe this matrix can as well carry important semantic values in regards to comparison between terms.

2.1.2. *Non-negative Matrix factorization*

The non-negative matrix factorization algorithms propose a way to approximately decompose an initial matrix V into a product of two matrices W and H that contain no negative values as in Eq. (3). These two matrices can have one dimension of the initial matrix and one reduced dimension. Then to each term the corresponding values into the reduced number of dimension can be interpreted as the degree of belonging to different semantic classes. This approach was initially developed for the semantic decomposition of images [16].

$$V \approx WH \tag{3}$$

The algorithm for non-negative matrix factorisation that we have experimented here is a Euclidean iterative approach. The matrix W is randomly initialized, the matrix H is arbitrarily initialized to non-zero positive values (e.g. 0.001) and at each iteration these two matrices are updated with multiplication rules in order to have their product converge towards V.

$$H = H.\frac{W^T V}{W^T W H} \qquad\qquad W = W.\frac{V H^T}{W H H^T} \tag{4}$$

The multiplicative update rules applied at each iteration are those of Eq. (4). The convergence is evaluated with the Euclidean distance between V and WH developed in Eq. (5) for the Euclidean distance between two matrices A and B of similar dimensions. The algorithm has been proven to converge to a local optimum [17].

$$||A - B||^2 = \sum_{ij} (A_{ij} - B_{ij})^2 \tag{5}$$

2.1.3. *Random Projection*

Random Projection (RP) is based on the fact that a term-document matrix computed from a corpus is sparse, allowing the vector representations to be projected onto a basis comprising a smaller number of randomly allocated vectors. Due to sparseness condition, the basis of random vectors has, in general, a high probability of being orthonormal [2]. The algorithm proceeds in 3 steps after the creation of a term-term matrix according to figure 1 :

- Create an empty matrix where rows represent terms and columns the new random vectors of dimension t,
- Randomly insert in each term vector $t/6$ of positive seeds and $t/6$ of negative seeds,
- Generate a matrix where rows are terms and columns new dimensions by adding the corresponding random vector to a term each time it appears in a context window,

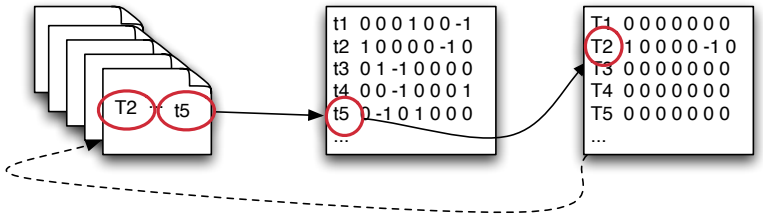

Fig. 1. Random projection of high dimensional vectors.

This can be seen mathematically as the new representation M^{random} of an initial term-term matrix M representing n terms reduced to k dimensions through a random matrix as in Eq. (6).

$$M^{random}_{k \times n} = Random_{k \times n} M_{n \times n} \tag{6}$$

The number of positive and negative random seeds initially followed a Gaussian distribution but it has been shown [1] that a probabilistic distribution with 1/6 is equivalent.

2.2. *Taking word order into account*

The preceding approaches to computing semantic vectors only deal with bags of words -no notion of the ordering of the words is taken into account. However since we deal here with natural language one might think that word order is important to take into account (since a *red wine* is quite different than a *wine red* ...).

Two methods are introduced for encoding word order that both rely on high-dimensional random projection. The second one uses a permutation of vectors and has been inspired by the first one that uses a convolution of vectors to encode *n-grams*.

2.2.1. *BEAGLE*

BEAGLE, or Bound Encoding of the Aggregate Language Environment [12] is one of the more recent examples of a computational model of word meaning. The major

advance offered by BEAGLE is the word representations include *both* a consideration of order information (i.e., structure) in addition to word context information (i.e., meaning).

The basis for encoding structure is an outer product of two vectors resulting in a matrix which is then compressed into a vector representation via an operation known as convolution. By way of illustration, assume the word s is represented by the three dimensional vector $b = (1, 0, 2)$, and the word w to be represented by the vector $d = (3, 1, 2)$. The association between these words can be represented as the outer product of these two vectors. More specifically, the transpose of the vector d is a column vector of three rows and when multiplied by row vector b gives rise to a 3×3 matrix:

$$\begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} \times (3, 1, 2) = \begin{pmatrix} 3 & 1 & 2 \\ 0 & 0 & 0 \\ 6 & 2 & 4 \end{pmatrix}$$

The resulting matrix represents an ordered association between d and b denoted $d \otimes b$. Such matrices have been used to model ordered word associations in human memory, e.g., [10] and more recently in BEAGLE.

The above scheme using outer products can be generalized into representing arbitrarily long sequences of words by using the Kronecker (tensor) product (note that the "outer" or "dyadic" is a special case of the tensor product between two vectors of the same dimension), but the tensor representations explode rapidly in dimensionality. One approach is to constrain the dimensionality by compressing the outer product into a vector representation of the same dimensionality as constituent vectors in the outer product. This is the approach adopted in the holographic reduced representations used to encode word order information in BEAGLE [22]. An operation known as circular convolution is used to achieve the desired compression. This operation cycles through the outer product. For example, given two vectors $x = (x_0, \ldots, x_n)$ and $y = (y_0, \ldots, y_n)$, the circular convolution $y \circledast x$ results in an n-dimensional vector $z = (z_0, \ldots, z_n)$ whereby each component z_i of the vector representation is computed according to Eq. (7).

$$z_i = \sum_{j=0}^{n} x_{j \ mod_{n+1}} . y_{(i-j) mod_{n+1}} \tag{7}$$

The above equation can be visualized as depicted in figure 2.

BEAGLE uses two different vectors for each word w in the model: a) an environmental vector, and b) a memory vector. The environmental vector is a word signature vector with elements of the vector sampled from a normal distribution with mean 0.0 and variance $1/n$, where n is the dimensionality of the vector. The information in the memory vector can also be stored in two separate vectors, one for context and one for structure.

The context vector is a standard co-occurrence vector for w the components of which give a weighted representation of how words are co-occurring with word w

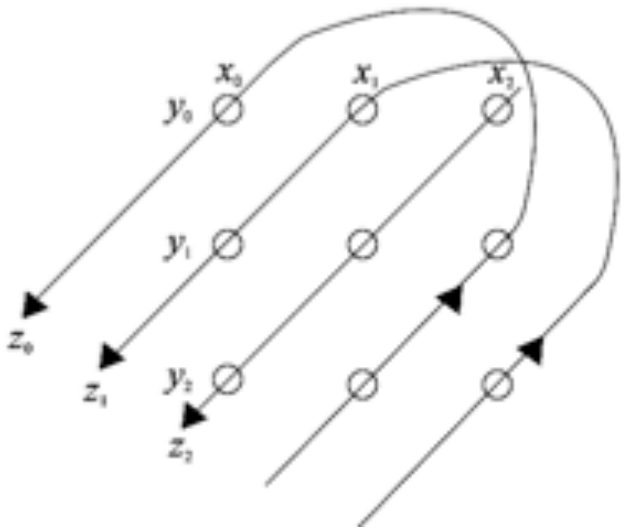

Fig. 2. Circular convolution

using a sentence as the unit of context.

The structure vector o_w is used to accumulate word order information formed by the superposition of vectors representing $n-$grams involving word w. For example, consider the sentence "A dog bit the mailman". The structure vector o_{dog} is built up as a sum of so-called "bindings", each of which is defined in terms of a convolution. For example, for bi-grams ($n = 2$),

$$\text{bind}_{\text{dog},1} = e_{\text{a}} \circledast \Phi$$
$$\text{bind}_{\text{dog},2} = \Phi \circledast e_{\text{bit}}$$

The position of the word being coded is represented by a constant random placeholder vector, Φ (sampled from the same element distribution from which the environment vectors e were constructed).

All n-gram vectors $2 \leqslant n \leqslant 7$ are thus formed are then superposed into a single vector, normalised, and then added to the structure vector for the target word. For example,

$$o_{\text{dog}} = \sum_{j=1}^{7} \text{bind}_{\text{dog},j}$$

Once again, when all words in the corpus have been processed, the structure vector for each word is normalised, and this normalised vector represents the structure signal for that word in the context of the corpus.

The context vector and structure vectors can then be mixed in some way so as to produce a single vector representation for each word. The advantage of this is that

the model becomes more flexible and different mixtures of context and structure information can be examined. This aspect will be manipulated in the experiments to follow in order to determine the optimum mixture between structure and context for semantic vectors.

2.2.2. *Permutations*

The permutation model proposed by [25] also attempts to represent order information in the context of the random projection model described above. For each word, a random permutation that shuffles the coordinates is generated, and applied as many times as necessary. The random permutation allows to generate a nearly orthogonal representation.

To encode the fact that some word w is in nth position in an n-gram, the initial vector of w will be permuted n times and then added to the other vectors in the n-gram. Let π be a permutation function for a the random vector of a word, then a structure vector of a word w_i is the sum of the permuted vectors of its neighbours w_{i-n} to w_{i-1} and w_{i+1} to w_{i+n} with a context window of size $2n + 1$.

$$w_i = \pi^{-n} w_{i-n} + ... + \pi^{-1} w_{i-1} + 0 + \pi w_{i+1} + ... + \pi^n w_{i+n} \tag{8}$$

These representations can be reversed in order to retrieve some words occurring in a specific context, or they can be used to retrieve words occurring in similar contexts given a word.

3. Experiments

3.1. *Experimental setup*

As a means to compare the semantic vector models above, the TOEFL synonym task on the TASA corpus was used. The basic hypothesis is the higher the TOEFL score, the better the quality of the underlying semantic vector. This choice follows many similar evaluations in the literature and allows our results to be placed in the perspective of other published results.

The TOEFL synonym test comprises 80 questions. Each question is multiple choices, made of a question word and four potential answers. A question is "incomplete" if the question term is unknown to the model in question, for example, because the question words were not present in the model. In the main experiment both the number of correct answers and the number of answerable questions will be reported. In the best results section the scores will be calculated according to the measure introduced in [15] where non-answerable questions will be scored 0.25 each thereby simulating guessing.

The TASA contains 44,486 documents of "General Reading up to 1st year college". It is assumed American students can learn relevant vocabulary and language usage from these readings. These documents contain 148,221 different non-stop terms for a total of 8,605,497 words.

Some pilot studies let us determine for each method if stemming was enhancing the results. The answer varies across the methods, the best solution being used for each method.

3.2. *Baseline*

As a baseline we have implemented what seemed one of the best approaches without dimension reduction [4]. The approach is the ratios inspired from PMI as described in Eq. (2). The approach has been experimented on the non-stemmed TASA corpus and the graph on Figure 3 reports the number of correct answers to the TOEFL test when the model uses various frequency limits (resulting in various number of terms). Since it was not clear which window size had been used in their main experiments, we experimented here with context windows of 3 and 5 words (e.g., 1 and 2 words on each side).

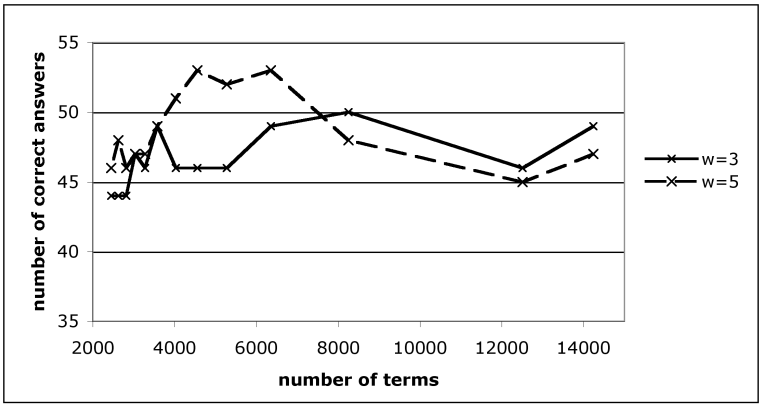

Fig. 3. Accuracy of PMI models with context window of 3 and 5 words and different number of terms used (including the terms from the TOEFL experiment).

The number of terms has been set according to various minimal term frequency thresholds. The results show that the stability of the results is not as good as it appears in the experiments reported in [4]. The number of correct answers on the TOEFL test is lower than their findings although we didn't process with as many words. The context window of size 5 seem to provide overall better results. This size will therefore be used in the following experiments on dimension reduction.

3.3. *The effect of dimension reduction*

3.3.1. *Random Projection*

We employed an implementation of Random Projection provided by the semantic vectors package (http://code.google.com/p/semanticvectors/)[33]. Both

corpus and questions were stemmed with a Porter Stemmer implementation (http://tartarus.org/ martin/PorterStemmer/) and the corpus is indexed with Lucene (http://apache.lucene.org) to generate the initial matrix. Term-context matrices were investigated with a window size of 5 words (2 on each side). The minimum frequency of terms in the initial representation is set to 2 leading to 41,845 terms. The values of the initial seeds are either -1 or +1. Over the 80 questions of the TOEFL test, two are incomplete within all models constructed using Random Projection with stemming.

3.3.2. *Dimensionally reduced HAL*

For NMF and SVD experiments, the initial semantic space could not contain all terms in the corpus due to computational limitations. The term frequency has been used for the selection of the terms for the initial HAL matrix. In order to ensure the presence of target terms with a restrained number of terms all the terms of the TOEFL experiments (400) have been added to the terms over the frequency limit imposed.

The algorithm used for SVD reduction is based on the INFOMAP (http://infomap-nlp.sourceforge.net/) implementation which, in turn, is based on a Word Sense Discrimination model [26]. In this approach, dimension reduction by SVD is central to producing semantic vectors. The HAL matrix used for this experiment was 1,000 contexts × 15,000 terms and has been processed on non-stemmed terms. Rows and columns have been sorted by term frequency and 50 most frequent terms have been dropped from columns as stop words.

The NMF algorithm has been implemented using the semantic vectors package. The HAL matrix is computed directly from a Lucene index and NMF classes from TCT software (http://mlg.ucd.ie/nmf) are computing the reduction. The cosine measure implemented in the package has then been used for comparisons. The semantic space is generated according to a term frequency threshold of 500 occurrences leading to 2,522 terms in the initial HAL matrix (including the 400 TOEFL experiment terms).

3.3.3. *Results*

This first series of experiments intends to examine the quality and stability of the semantic vectors under dimension reduction. For evaluating the stability of semantic representation, the RP model involved 5 distinct runs since each run involves a different random basis.

The average results of testing RP with various dimensions are reported on Figure 4. These are results for non-stemmed data. Lower dimensions and stemmed data are not represented here since they scored far lower results. Some other experiments have been conducted with different ratios between the number of dimensions and the number of seeds (1/2, 1/4) and the results were very similar. The five individual

results for each run exhibit consistently quite a large variation suggesting the underlying semantic vector representation is not very stable. The average performance

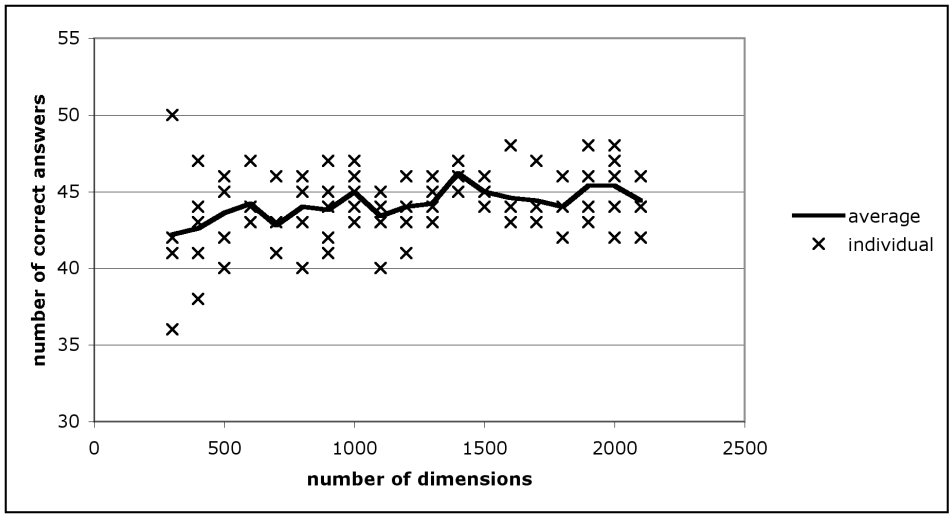

Fig. 4. Accuracy of Random Projection with a context window 5 for various numbers of dimensions.

does seem to increase with the dimensionality with 46.2 correct answers on average for 1,400 dimensions, and thereafter decreases. However, the lack of stability doesn't allow for any firm conclusions since the best result of 50 correct answers is obtained with 300 dimensions which also leads to the lowest average accuracy.

For evaluating the stability of semantic representation, the SVD model has been tested with all consecutive dimensions k between 0 and 1,000. SVD has been experimented with the use of the first matrix in the decomposition (see Section 2.1.1) for all dimensions from 0 to 1,000. Pilot experiments showed that the use of the singular values was consistently followed by a lower accuracy. Figure 5 provides the number of correct answers for each dimension reduction. Best results are obtained with a number of dimensions around 200 and decrease to less than 20 correct answers past 800 dimensions. The graph shows variations of 5 correct answers from one dimension to the immediately preceding and following ones. The stabilizing of the results while increasing k was to be expected as the singular values are sorted in decreasing order (with a growing k the added information is of decreasing impact).

For evaluating the stability of semantic representation, the NMF model involved 5 distinct runs since each run involves a different random initialisation of the matrix W (see Eq. (4)). Figure 6 displays the results of the NMF approach between 300 and 2,100 dimensions. Apart from the reduction to 300 dimensions that scores a very low average number of correct answers (41), the number of dimensions between 500 and 2,100 does not really drive the quality of the results. The best results in

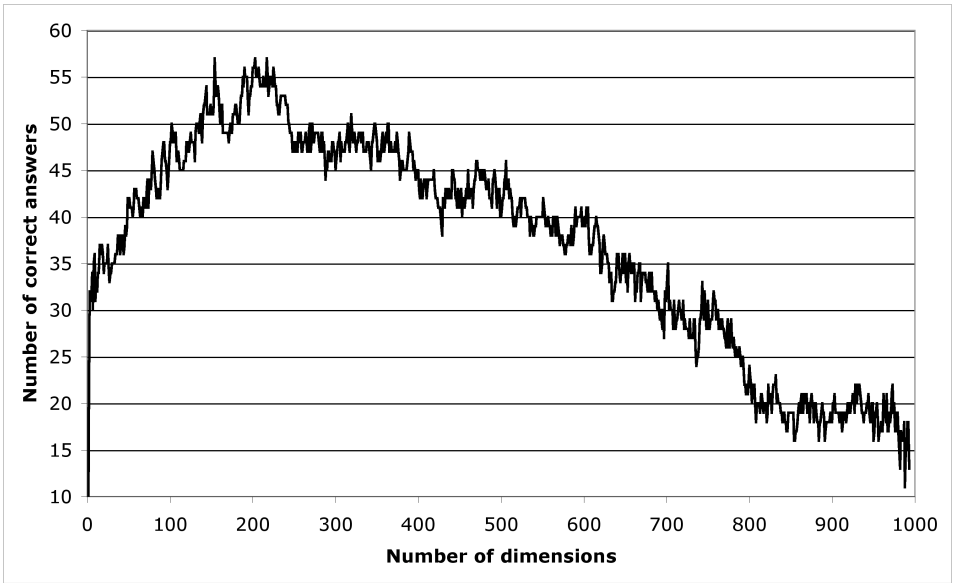

Fig. 5. Accuracy of singular value decomposition with a context window 5 for various numbers of dimensions.

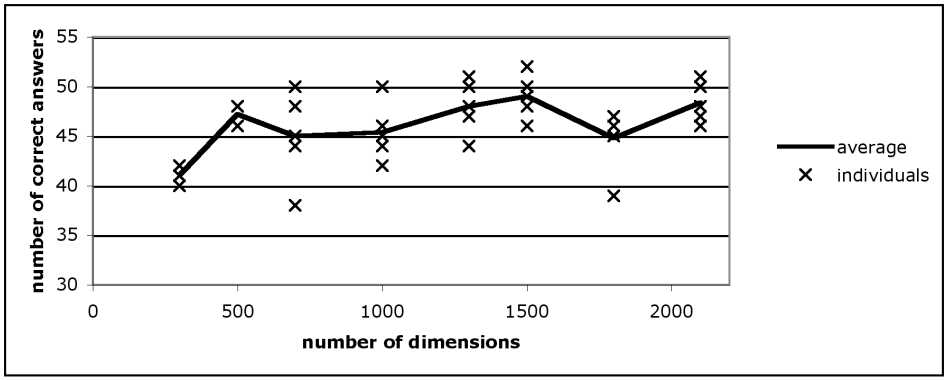

Fig. 6. Accuracy non-negative matrix factorization with a context window 5 for various numbers of dimensions.

average are obtained with 1,400 dimensions for 49 correct answers. The best model is built with 1,400 as well and scores 52 correct answers. The stability of results was good for low dimensions between 300 and 500 and suddenly very low with a more than 10 correct answer variation for 800 dimensions.

3.4. *The effect of word order*

3.4.1. *The size of the context window*

Random Projection was used for this investigation as it allows large numbers of systematic experiments to be rapidly conducted.

Figure 7 shows the average results for various context window sizes with random vectors. The random vectors have been computed with 1,800 dimensions since a pilot study not involving context windows with random projection shown this dimension leading to best results. The window sizes refer to the total number of words taken into account including the target word. The results show that the smallest context

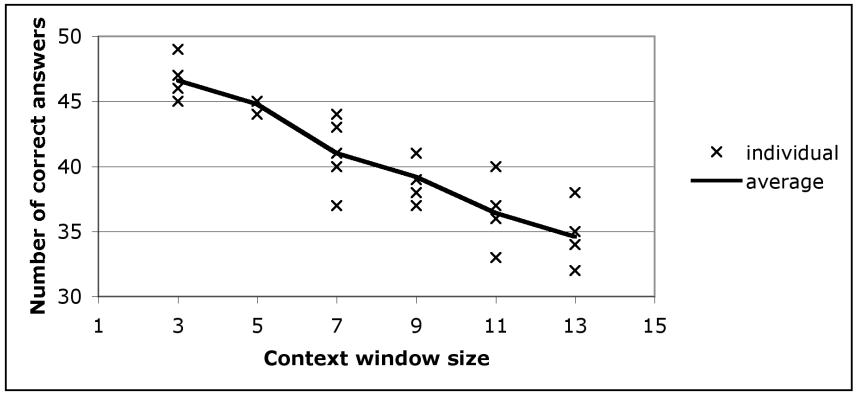

Fig. 7. Accuracy of Random Projection using a positional index (a word-word matrix) with different context window sizes.

window (3 words) provides the most accurate results on the TOEFL test with 45 correct answers out of 78 in average. This implies that the model constructed based on the co-occurrences of the words only with the previous and the next word (these not being stop words) performs the best for the synonym test. With a context window size of up to 9, the results are higher (40 correct answers out of 78) than when using whole documents as contexts. It is also important to note that context based models are more computationally expensive since they involve positional information about words in the documents.

The optimal window size varies between 3 and 5 words depending on the model. These rather small context windows are always better than larger context windows as accuracy decreases rapidly when the context window is enlarged. This is consistent with experiments reported by others using the TOEFL test.

3.4.2. *BEAGLE*

The BEAGLE model used was implemented in Java using the open source numerical libraries OpenNLP (http://opennlp.sourceforge.net/) and Parallel Colt (http://piotr.wendykier.googlepages.com/parallelcolt). Since the BEAGLE architecture doesn't allow us to force some terms into the model, the maximum number of possible correct answers was 73. The remaining 7 have been credited 0.25 points each according to Landauer's proposed "guessing" scheme.

The TOEFL test was first run on the implemented BEAGLE model using a range of mixtures of context and structure vectors. The structure vectors for words were computed with 5-grams. Recall in the BEAGLE model, the representation of words can be manipulated as mixtures of structure (word order) and context (co-occurrence). Results for various percentages of vector representation being contributed by the context vector are displayed on the graph on Figure 8. This first

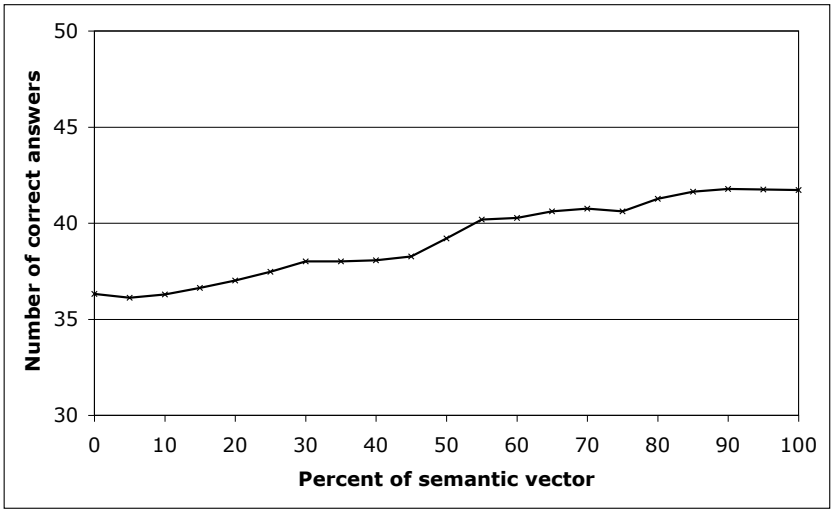

Fig. 8. Average accuracy of BEAGLE models by percent of context vector

experiment shows that best average accuracy is obtained with 90% of the vector representation for the word coming from the context vector and 10% from word order (structure). The figure also shows that between the 30% and 100% levels the increase in accuracy performance is not linear. It is not clear why this is the case.

For the second series of experiments on variation of dimension we used 90%

contribution from the context vector as this provided best average results. The detailed and average results are displayed on the graph on Figure 9. Average results for the same series of models computed using a 70% contribution from the context vector are also displayed on the same graph as the percentage is the best of the immediately inferior threshold on Figure 8. The results show that the stability is

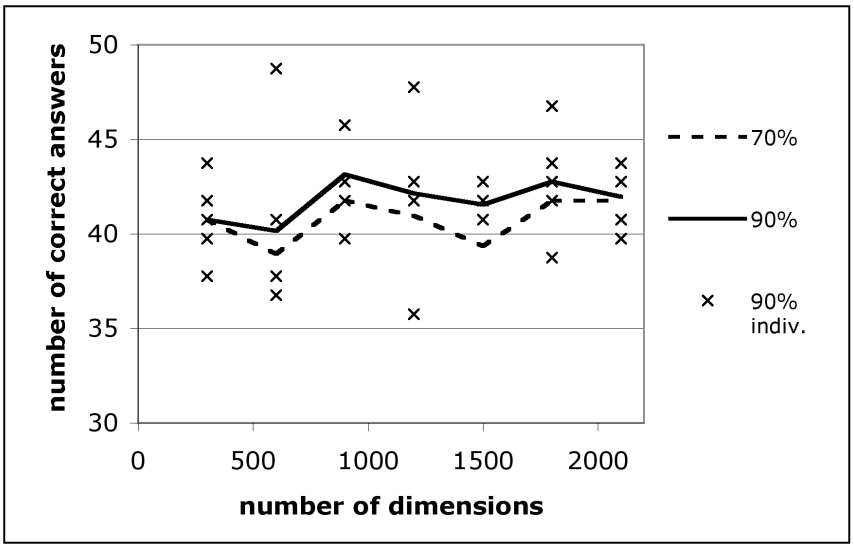

Fig. 9. Accuracy of BEAGLE models with 90% context vector and average accuracy of BEAGLE models with 90% and 70% context vector by number of dimensions.

even lower than for previous models with variations up to 12 correct answers for one set of parameters. The average performance increase against the number of dimensions is not linear. The consistency between the two curves suggests that the random initialisation of the environment vectors may be responsible for the variations and that the proportion of context has not much impact on the accuracy.

3.4.3. *Permutations*

The permutation model is available in the semantic vectors package. Although the main incentive for using permutations is to compute the most likely neighbouring words, we have used the permuted representation to search for most similar words. Different sizes of context window have been experimented between 3 and 9 words. The results are shown on figure 10. The smaller context window the more accurate the results, as was the case above. The permutation model performs at best 48 correct answers for 2,100 dimensions with a context window of 3.

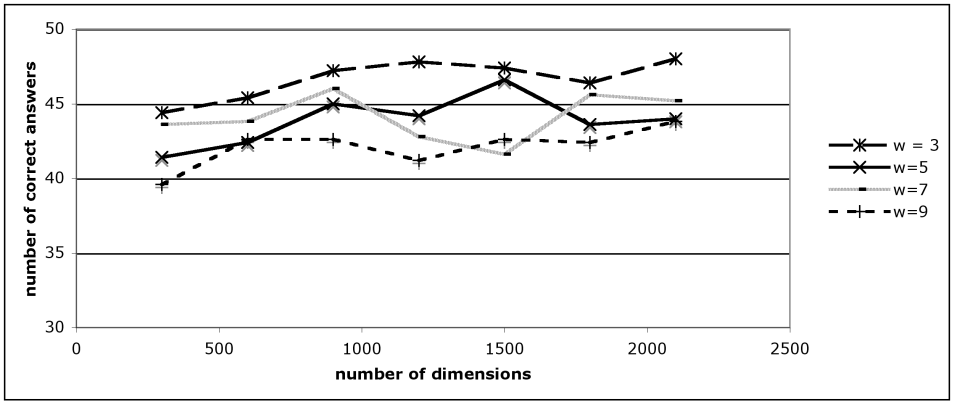

Fig. 10. Average number of correct answers of permutation models with window sizes from 3 to 9 and different number of dimensions used, without stemming.

3.5. *Best results*

Under what circumstances is each model performing best, whereby the measure is normalized by adding 0.25 for each question that could not be answered by a given model? Table 2 reports the best scores achieved by each model previously presented. Only the SVD model outperforms the best PMI model. All of the models can achieve more than 60% of correct answers provided best conditions are met.

The optimal number of dimensions for reduction varies between models from 300 for one RP model to 1,500 for NMF models.

Model	Correct	PCorrect	minN
$PMIw_3$	50	62.5%	8,250
$PMIw_5$	53	66.25%	4,500
RP_{single}	50	62.5%	300
$RP_{average}$	46.2	57.75%	1,400
SVD	57	71.25%	154
NMF_{single}	52	65%	1,500
$NMF_{average}$	49	61.25%	1,500
$Permw_3$	48	60%	2,100
B_{single}	48.75	60.9%	600
$B_{average}$	43.15	53.9%	900

Table 2. Best Performance of various models measured by the number of correct answers (Correct), the percentage of correct answers (PCorrect) and the minimal number of dimensions achieving these performances (minN).

Percentages of correct answers are provided in order to allow a comparison with

former results available in the literature.

4. Discussion

4.1. *comparison with previous results*

Comparison with previous published work should be viewed in light of doubt regarding the size of the underlying corpus. In this paper, the TASA corpus used to build the semantic spaces comprises 44,486 documents whereas in other studies reported for the same task and corpus in the literature and cited below, the size is either 37,600 or 30,473 articles. We are unable to explain this discrepancy. Several results have been reported on the use of LSA: [15] reported 64.5% correct answers, [12] report results of 55.31% correctly answered questions for LSA and [9] found 63.6% of correct responses using the cosine similarity and 61.5% using an inner product instead. Random Indexing [13] using word contexts gave 35-44% with unnormalised 1800 dimensional vectors and 48-51% with normalised vectors. [12] report results of 55.6% for BEAGLE without context information and 57.81% for BEAGLE with both context and order information. An other approach has been experimented in literature consisting of probabilistic generative model based on LDA [3] and applied to semantic correlation by [9] in *Topics*. They experimented both forward (the probability of the answer knowing the question) and backward (the probability of the question knowing the answer) probabilities for modelling. Their best results over 44 computable questions is 63.6% for the former and 70.5% for the latter. In short, the best runs of the models reported in this article are competitive in relation to previously published TOEFL scores on a similar corpus.

It should be noted that much higher scores have been reported in the literature using either external resources such as lexicons in combination with semantic spaces [30] or simply more balanced and bigger corpora [23]. A result of 100% was recently achieved using a larger corpus and model based on SVD [5].

4.2. *Dimension reduction*

The results above do indicate that dimension reduction does have a positive impact on the quality of semantic vector representation. This is in-line with previous research using SVD [26], however, our study seems to show the quality of semantic representation produced by SVD to be superior to that of Random Projection. In both cases, however, it is hard to find the optimal reduced dimensionality, so in practical application, there may be little to choose between them. SVD reduction seems better suited to a fixed application in relatively static and closed domains with predictable needs. Random Projection would be better for dynamic applications where new data have to be dynamically taken into account in the model and where needs are evolving and unpredictable.

SVD uses a unique and ordered decomposition of the initial matrix, and it is generally understood that the eigenvectors corresponding to the highest eigenvalues

represent the diversity of the corpus, the following ones are most likely to be the core components of its semantics, and the last ones are bearing the noise. It is important to keep in mind that adding more dimensions does not change the previous ones. Such an interpretation would explain the general behaviour when varying the number of dimensions. It is also based on these premises that the highest performance could be reached (altough also building on larger corpora) by emphasizing the mid-range eigenvectors [5]. Non negative matrix factorization, unlike SVD, does not value some dimensions more than the others. Also, adding more dimensions reorganise all the previous ones. All the dimensions can thus be interpreted as a given semantic cluster of information present in the corpus. In other words, the dimensions encode level of commonness rather than levels of variation. This most likely factors for good performances reached for higher number of dimensions.

4.3. *Stability of semantic representation*

The use of random vectors in RP raises the issue of the stability of semantic representations produced by these models. The quite wide range of scores obtained with the different runs of Random Projection models as detailed on Figures 4 and 7 suggests a degree of variability between the respective representations.

In [28] we report an experiment for improving the stability of results with RP. The approach consists in repetitively retraining the models by repeating the last two steps of the process introduced in section 2.1.3. The results showed that this method doesn't impact the stability of the results across several runs. SVD does not rely on any random initialisation but there can be variations up to 5 correct answers between 2 consecutive dimensions.

The lack of stability of performance is probably one of the main factors for the variation of best results from one experiment to another as reported so far in the literature. This also implies that any model must be refined to the best set of parameters according to the task it is built for.

4.4. *The effect of word order*

The best result of BEAGLE with a component of word order (10%) performs closely to other models only using co-occurrence. Thus word order doesn't seem to play much of a role in the semantics needed for the synonym-finding task. The results with the permutation model are similar in this respect.

The encoding of word order provides some integration of syntactic information and is most thought for the encoding of multi-word expression. In the TOEFL task, words of similar syntactic function are compared and it is their behaviour as a single word that is of interest. As such it is likely that either they will tend to occur in similar types of construction (ie. in the same order) or, on the case of syntactic re-ordering (question form, passive voice) word order is not indicative of semantics.

The experiments tended to show that very small context windows produced the best results. This statement is a cautious one as context window size was not

systematically manipulated within the models. Intuitively, however, the statement does seem reasonable, as synonyms tend to appear in similar contexts. This, then, raises the question of the suitability of the TOEFL test as a means of investigating semantic vectors. We conclude TOEFL to be incomplete in this regard, but nevertheless it does provide a useful backdrop for comparison.

5. Conclusion and future work

The TOEFL test has been used as tentative for objectively measure the performance of each approach and compare them. The experiments have confirmed that small context windows tend to provide the best results. The best results on TOEFL test have been performed by the Singular Value Decomposition (SVD) model using only the row representation of its singular value decomposition where 71.25% of the questions were correctly answered. The results showed that the differences are not great and their significance is lowered by the low stability of results across slight changes (for SVD) or across different random initialisation (for Random Projection (RP) or Non-negative Matrix Factorization (NMF)). Hence someone who would face the choice of one of these methods should consider a) the specific advantages of each method and b) benchmark parameters to find the most efficient representation for a particular task.

The advantage of using random projection is that it is computationally cheap and the model can be dynamically updated while new data are appearing. This would be especially useful for applications that require updated knowledge as in broadcast news processing, trends analysis and blogs processing. Random Projection gave its highest results of 62.5%. The advantage of NMF is that the model can provide a description of the inherent dimensions and be used to extract themes across the corpus as shown by [34]. This can be useful when an interface is required as it is self-explaining. The best accuracy achieved by NMF in our experiments is 65%. The advantage of SVD is that it works well with small amounts of data. Moreover, due to the associated computational cost they can best be applied in closed domains. The experiments reported in this paper demonstrate that dimension reduction with SVD is likely to produce semantic vectors of better quality. The advantage of permutation and BEAGLE models is its ability to cater for structure (word order) and co-occurrence information (meaning) within the single representation. However, the experiments also highlighted that the integration of word order within semantic vectors does not improve scores in the TOEFL test over semantic vectors built from straight co-occurrence in context.

In the future, it will also be worth systematically investigating how stable semantic vectors are with slight corpus changes, or on larger corpora. Potential other tasks for examining semantic vectors are replications of free association norms [27], word priming [11] and semantic categorization [5].

Acknowledgments

This project was supported in part by the Australian Research Council Discovery project DP0663272. We are grateful to Lance DeVine for implementing the BEAGLE model and running the tests. We are grateful to Tom Landauer for providing the TASA corpus.

References

[1] Achlioptas, D., 2001. Database-friendly random projections. In: Proceedings of the Symposium on Principles of Database Systems. pp. 274–281.
[2] Bingham, E., Mannila, H., 2001. Random projection in dimensionality reduction : applications to image and text data. In: ACM (Ed.), Proceedings of the 7th international conference on Knowledge discovery and data mining. New York, NY, USA, pp. 245–250.
[3] Blei, D. M., Ng, A. Y., Jordan, M. I., 2003. Latent dirichlet allocation. Journal of machine learning research 3, 993–1022.
[4] Bullinaria, J. A., Levy, J. P., August 2007. Extracting semantic representations from word co-occurrence statistics : a computational study. Behavior research methods 39 (3), 510–526.
[5] Bullinaria, J. A., Levy, J. P., 2012. Extracting semantic representations from word co-occurrence statistics : Stop-lists, Stemming and SVD. Behavior research methods 44, to appear.
[6] Burgess, C., Livesay, K., Lund, K., 1998. Explorations in context space: words, sentences, discourse. Discourse Processes 25 (2&3), 211–257.
[7] Firth, J. R., 1957. A synopsis of linguistic theory, 1930-1955. Studies in linguistic analysis, 1–32.
[8] Golub, G., Loan, C. v., 1996. Matrix Computations. John Hopkins University Ptess.
[9] Griffiths, T. L., Steyvers, M., 2007. Topics in semantic representation. Psychological review 114 (2), 211–244.
[10] Humphreys, M., Bain, J., Pike, R., 1989. Different ways to cue a coherent memory system: A theory for episodic, semantic and procedural tasks. Psychological Review 96, 208–233.
[11] Jones, M. N., Kintsch, W., Mewhort, D. J. K., 2006. High-dimensional semantic space accounts of priming. Journal of memory and language 55, 534–552.
[12] Jones, M. N., Mewhort, D. J. K., 2007. Representing word meaning and order information in a composite holographic lexicon. Psychological review 114 (1), 1–37.
[13] Kanerva, P., Kristoferson, J., Holst, A., 2000. Random indexing of text samples for latent semantic analysis. In: Erlbaum (Ed.), Proceedings of the 22nd annual conference of the cognitive science society. New Jersey, USA.
[14] Landauer, T., 2002. On the computational basis of learning and cognition: Arguments from lsa. In: Ross, B. (Ed.), The Psychology of Learning and Motivation. Vol. 41. Academic Press, pp. 43–84.
[15] Landauer, T., Dumais, S. T., 1997. A solution to plato's problem : the latent semantic analysis theory of acquisitionm induction and representation of knowledge. Psychological review 104 (2), 211–240.
[16] Lee, D. D., Seung, H. S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791.
[17] Lee, D. D., Seung, H. S., 2001. Algorithms for non negative matrix factorization. Advances in neural information processing systems 13, 556–562.

[18] Lowe, W., 2001. Towards a theory of semantic space. In: Moore, J. D., Stenning, K. (Eds.), Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society. Lawrence Erlbaum Associates, pp. 576–581.

[19] Lund, K., Burgess, C., 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. Behaviour research methods, instruments and computers 28 (2), 203–208.

[20] Lund, K., Burgess, C., 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior Research Methods, Instruments & Computers 28 (2), 203–208.

[21] Manning, C. D., Schütze, H., 1999. Foundations of statistical natural language processing. MIT Press, Cambridge, MA.

[22] Plate, T. A., 2003. Holographic Reduced Representation: Distributed Representation for Cognitive Structures. Vol. 150. CSLI Lecture Notes.

[23] Rapp, R., 2003. Word sense discovery based on sense descriptor dissimilarity. In: Proceedings of the Ninth Machine Translation Summit. New Orleans, USA, pp. 315-322.

[24] Sahlgren, M., 2005. An introduction to random indexing. In: Proceedings of Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering. Copenhagen, Denmark.

[25] Sahlgren, M., Holst, A., Kanerva, P., July 23-26 2008. Permutations as a means to encode order in word space. In: Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci'08). Washington D.C., USA.

[26] Schütze, H., 1998. Automatic word sense discrimination. Computational linguistics 24 (1), 97–123.

[27] Sitbon, L., Bellot, P., Blache, P., 2008. Evaluation of lexical resources and semantic networks on a corpus of mental associations. In: Proceedings of the 6th Language Resources and Evaluation Conference. Marrakech, Morocco.

[28] Sitbon, L., Bruza, P., December 2008. On the relevance of documents for semantic representation. In: Proceedings of the 13th Australiasian Document Computing Symposium. Hobart, Australia, pp. 19–22.

[29] Turney, P., 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In: De Raedt, L., Flach, P. (Eds.), Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001). pp. 491–501.

[30] Turney, P.D., Littman, M.L., Bigham, J., Shnayder, V., 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). Borovets, Bulgaria, pp. 482-489.

[31] Turney, P., Pantel, P.,2010. From frequency to meaning: vector space models of semantic. Journal of Artificial Intelligence Research 37, 141–188

[32] Widdows, D., 2004. Geometry and Meaning. CSLI Publications.

[33] Widdows, D., Ferraro, K., 2008. Semantic vectors : a scalable open source package and online technology management application. In: Proceedings of the 6th Language Resources and Evaluation Conference. Marrakech, Morocco.

[34] Xu, W., Liu, X., gong, Y., 2003. Document clustering based on non-negative matrix factorization. In: Proceedings of SIGIR'03. Toronto, Canada, pp. 267–273.