Modelling Word Meaning using Efficient Tensor Representations

Mike Symonds[a], Peter Bruza[a], Laurianne Sitbon[a], and Ian Turner[b]

[a] Department of Computer Science, Queensland University of Technology,
Brisbane, Queensland, Australia
{m.symonds, p.bruza, l.sitbon}@qut.edu.au
[b] Department of Mathematics, Queensland University of Technology,
Brisbane, Queensland, Australia
i.turner@qut.edu.au

Abstract. Models of word meaning, built from a corpus of text, have demonstrated success in emulating human performance on a number of cognitive tasks. Many of these models use geometric representations of words to store semantic associations between words. Often word order information is not captured in these models. The lack of structural information used by these models has been raised as a weakness when performing cognitive tasks.

This paper presents an efficient tensor based approach to modelling word meaning that builds on recent attempts to encode word order information, while providing flexible methods for extracting task specific semantic information.

Keywords: Semantic Space, Unsupervised Learning, Tensor, Word Order

1 Introduction

Research in the area of natural language processing has demonstrated that psychologically relevant models of word meaning can be learnt from exposure to natural language (Landauer and Dumais, 1997; Lund and Burgess, 1996; McRoy, 1992; Turney, 2008). Many of these models are based on vector representations built from word co-occurrence statistics that aim to model various semantic relationships. Even though these semantic space models appear to identify words with similar meanings, it has been argued that they do not incorporate syntax or achieve other basic cognitive language abilities (Perfetti, 1998).

Recently, a number of semantic space models (that do not use pre-existing knowledge) have been developed that encode word order into the semantic space, hence capturing more structural information about word associations (Jones and Mewhort, 2007; Sahlgren *et al.*, 2008). Jones and Mewhort (2007) concluded that a model that pays attention to both context and word order while learning, stands a greater chance of matching the trends found in human data. The strength of a geometric approach to encode word order is in the ability to work within a mathematically well defined framework, including the availability of many existing operators from linear algebra, such as Kronecker products. However, to our knowledge there has been very few efficient methods for implementing uncompressed Kronecker products when encoding word order information within a semantic space.

The main contribution of this paper is to present a novel, efficient approach to using Kronecker products to encode word order information within a semantic space. The other significant contribution is to demonstrate how applications can use our single representation to access various task specific semantic information.

2 Related Work

The main areas of research that provide a theoretical framework for our model include: (i) the structuralist approaches to defining word meaning, and (ii) the use of semantic spaces to model word meaning.

2.1 Word Meaning

Ferdinand de Saussure (1916) argued that meaning arose from the relationships between words. He called the two types of relationships that created this meaning: (i) syntagmatic and (ii) paradigmatic associations. Saussure's structuralist ideas provide a relatively clean linguistic framework, free of psychology, sociology and anthropology, within which we can distinguish between two types of word associations that can be used to model word meaning (Holland, 1992). This structuralist approach to linguistics has been used to motivate other semantic space models (Sahlgren *et al.*, 2008).

A syntagmatic association exists between two words if they co-occur more frequently than expected from chance. Some common examples may include "coffee-drink" and "sun-hot". A paradigmatic association exists between two words if they can substitute for one another in a sentence without affecting the grammaticality or acceptability of the sentence. Some common examples may include "drink-eat" and "quick-fast" (Rapp, 2002).

2.2 Semantic Space Models

Linked to structuralist ideas of linguistics, researchers have argued that word meaning can be modelled by comparing the distributions of words within text (Schütze, 1993). A popular approach to representing these word distributions is to collect word occurrence frequencies and place them in high-dimensional *context* vectors (Turney and Pantel, 2010). This approach allows techniques from linear algebra to be used to model relationships between objects, including semantic associations, within the geometric space.

Two of the most well-known semantic space models in literature are HAL (Hyperspace Analogue to Language; Lund and Burgess (1996)) and LSA (Latent Semantic Analysis; Landauer and Dumais (1997)). These two models differ in the way they build their context vectors. HAL builds context vectors by storing pre- and post-order word co-occurrence frequencies in a word-by-word matrix. Consider the HAL matrix, shown in table 1, created by the sentence "*a dog bit the mailman*", using a sliding context window with radius 2. The co-occurrence information preceding and post-ceding each word are recorded separately by the row and column vectors.

Table 1: Example HAL Space

	a	dog	bit	the
dog	2	0	0	0
bit	1	2	0	0
the	0	1	2	0
mailman	0	0	1	2

LSA differs from HAL in that LSA's context vectors are formed by collecting the word occurrence frequencies within each document to create a word-document matrix. A costly technique, known as single value decomposition (SVD), is then used to reduce the dimensions of the word-document matrix to the k most significant latent concepts. Even though models based on LSA and HAL have been shown to simulate human performance on a number of cognitive tasks, it has been argued by Perfetti (1998) that these models do

not capture concepts such as syntax or achieve other basic cognitive language abilities. A relevant example, includes the fact that LSA chose *nurse* over *doctor* when asked to determine the closest match to *physician* in a synonym judgement test. The lack of word order information in LSA is a result of the way in which it builds its context vectors, however, even though HAL would appear to hold word order information, it has been argued by Jones and Mewhort (2007) that HAL does not explicitly encode order information.

A number of recent semantic space models have tried to increase the amount of structural information encoded within the representations. These include the *Bound Encoding of the Aggregate Language Environment* (BEAGLE) model (Jones and Mewhort, 2007) and a permutation model (Sahlgren *et al.*, 2008) based on Random Indexing (RI) (Kanerva *et al.*, 2000). Both BEAGLE and the permuted RI model build their semantic spaces from a set of *fixed length* environment vectors. This approach allows the dimensionality of the semantic space to be contained. These fixed dimension approaches rely on the random assignment of environment vectors to create an approximately orthogonal basis, which is required to use many of the popular geometric distance measures.

In addition to forming context vectors, by summing environment vectors for terms that co-occur within the sliding context window, both BEAGLE and the permuted RI model create *order* vectors. To build order vectors BEAGLE binds the environment vectors using a circular convolution operation ($\circledast$), which is a mathematical function that compresses the Kronecker (outer) product of two vectors. The compression avoids the explosion in tensor order associated with Kronecker products, and is achieved by summing along the trans-diagonal elements of the outer product, giving rise to a vector dubbed a *holographic reduced representation* (HHR) (Plate, 1991). The resulting HHR created by the n-grams within the context window are added to the term's order vector. Circular convolution is non-commutative, such that $\boldsymbol{a} \circledast \boldsymbol{b} \neq \boldsymbol{b} \circledast \boldsymbol{a}$ for distinct vectors $\boldsymbol{a}, \boldsymbol{b}$. Non-commutativity is crucial as word order is usually not commutative.

The main drawback of BEAGLE's encoding method comes from the cost of the binding process and the loss of information through compression of the Kronecker products (Mitchell and Lapata, 2010). In the case of the permuted RI model, word order encoding is performed by rotating the coordinates of the sparse environment vectors in the direction of the co-occurrence (with preceding opposite to post-ceding) before summing the result with the order vector. This approach is much more efficient than circular convolution. The results of both BEAGLE and the permuted RI model show that including order information improves performance on a synonym judgement task over context information alone. We now present a model that formally encodes word order and provides the ability to compute semantic associations that underpin word meaning.

3 Building the Tensor Encoding Model's Semantic Space

Our tensor encoding (TE) model builds its semantic space using an efficient binding process based on Kronecker products of *theoretically* unbounded unit vectors. The result contains both context and order information in a single representation we call the *memory tensor*.

3.1 The TE Binding Process

The way in which the TE model encodes word order is illustrated by considering our binding process for the following example sentence, "*A dog bit the mailman*", where *A* and *the* are considered to be stop words (noisy, low information terms that are ignored) and hence will not be included in the vocabulary. The resulting vocabulary includes:

Term-id	Term	Environment vector
1	dog	$\boldsymbol{e}_{dog} = (1 \quad 0 \quad 0)^T$
2	bit	$\boldsymbol{e}_{bit} = (0 \quad 1 \quad 0)^T$
3	mailman	$\boldsymbol{e}_{mailman} = (0 \quad 0 \quad 1)^T$

The *memory tensor* for each term in the vocabulary is constructed by summing the resulting Kronecker products of the environment vectors within a sliding context window over the text. The number of environment vectors bound using Kronecker products impacts the order of the memory tensors. For this research a second order binding process was used, and results in second order tensors (matrices) being formed. Higher order TE models, which capture the co-occurrence frequencies of n-tuples, are left for future work. The second order binding process for the TE model is defined by:

$$\boldsymbol{M}_w = \sum_{k \in CW}^{k \prec w} \boldsymbol{e}_k \otimes \boldsymbol{e}_w^T + \sum_{k \in CW}^{k \succ w} \boldsymbol{e}_w \otimes \boldsymbol{e}_k^T, \tag{1}$$

where w is the target term, k is a non-stop word found within the sliding context window (CW), $k \prec w$ indicates that term k appears before term w in the context window, and $k \succ w$ indicates that term k appears after term w. Note, stop words are not bound, but they are included when determining the window boundaries. Consider the memory matrices created for the vocabulary terms using a sliding context window with radius 2.

Binding Step 1: $\overbrace{A_s \quad [dog] \quad bit \quad the_s} \quad mailman$

$$\boldsymbol{M}_{dog} = \boldsymbol{e}_{dog} \otimes \boldsymbol{e}_{bit}^T = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (0 \quad 1 \quad 0) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Binding Step 2: $\overbrace{A_s \quad dog \quad [bit] \quad the_s \quad mailman}$

$$\boldsymbol{M}_{bit} = \boldsymbol{e}_{dog} \otimes \boldsymbol{e}_{bit}^T + \boldsymbol{e}_{bit} \otimes \boldsymbol{e}_{mailman}^T = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (0\ 1\ 0) + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} (0\ 0\ 1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Binding Step 3: $A_s \quad dog \quad \overbrace{bit \quad the_s \quad [mailman]}$

$$\boldsymbol{M}_{mailman} = \boldsymbol{e}_{bit} \otimes \boldsymbol{e}_{mailman}^T = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} (0 \quad 0 \quad 1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

The resulting pattern is that all non-zero elements are situated on the row or column corresponding to the target term's term-id. If this vocabulary building process was performed over the entire corpus the general form of a *memory matrix* would be:

$$\boldsymbol{M}_w = \begin{pmatrix} 0, & \ldots, 0, & f_{1w}, & 0, & \ldots, 0 \\ & & \ldots & & \\ 0, & \ldots, 0, & f_{(w-1)w}, & 0, & \ldots, 0 \\ f_{w1}, \ldots, f_{w(w-1)}, & & f_{ww}, & f_{w(w+1)}, \ldots, & f_{wN} \\ 0, & \ldots, 0, & f_{(w+1)w}, & 0, & \ldots, 0 \\ & & \ldots & & \\ 0, & \ldots, 0, & f_{Nw}, & 0, & \ldots, 0 \end{pmatrix},$$

where f_{iw} is the value in row i column w of the matrix which represents the ordered co-occurrence frequencies of term i before term w, f_{wj} is the value in row w column j of the matrix that represents the ordered co-occurrence of term j after term w, and N is the number of unique terms in the vocabulary.

3.2 Capturing Stronger Proximity Information

Similar to HAL, our TE model captures stronger proximity information by weighting the strength of a co-occurrence inversely proportional to the distance between the target term and the interacting term. Formally, the binding process in equation (1) becomes:

$$\boldsymbol{M}_w = \sum_{k \in CW}^{k \prec w} (R - d_k + 1).\boldsymbol{e}_k \otimes \boldsymbol{e}_w^T + \sum_{k \in CW}^{k \succ w} (R - d_k + 1).\boldsymbol{e}_w \otimes \boldsymbol{e}_k^T, \tag{2}$$

where R is the radius of the sliding context window, and d_k is the distance between term k and target term w. To demonstrate, consider our previous example sentence, noting *bit* and *mailman* are 2 words apart in the sentence (as stop words are included when calculating distance within the context window):

$$\textbf{Binding Step (with proximity scaling):} \quad \overbrace{A_s \quad dog \quad [bit] \quad the_s \quad mailman}$$

$$\boldsymbol{M}_{bit} = 2 \times \boldsymbol{e}_{dog} \otimes \boldsymbol{e}_{bit}^T + \boldsymbol{e}_{bit} \otimes \boldsymbol{e}_{mailman}^T = 2 \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (0\ 1\ 0) + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} (0\ 0\ 1) = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

3.3 Flexible use of Word Order

Unlike BEAGLE and the permuted RI model, the TE model has the ability to access explicit context and order information within the one geometric representations. This means that order information can be easily ignored by combining rows and columns of the memory tensors. This can be efficiently achieved within similarity measures, as will be demonstrated in section 4.

3.4 Efficient Implementation of Tensor Computations

By using environment vectors that are unit vectors, our second order binding process creates sparse N-by-N memory matrices, with the percent sparseness proportional to $1 - \frac{2}{N} + \frac{1}{N^2}$. This sparseness, along with the fact that no multiplication of elements is required in the binding process, allows memory matrices to be efficiently computed and stored at an implementation level. To demonstrate, consider the memory matrix for *bit* in the proximity scaled example above. M_{bit} can be stored as a fixed dimensional vector of term-id (T), co-occurrence frequency (CF) pairs, (T CF):

$$\text{Storage vector for } \boldsymbol{M}_{bit} = [(-1 \quad 2) \quad (3 \quad 1)],$$

where parenthesis have been added to illustrate implicit grouping of (T CF) pairs, and the sign of the T component is used to capture the word order. Knowing that a context window of radius 2 was used, the storage vector above indicates that the word *dog* (term 1) appeared directly before (as indicated by the negative sign) the word *bit*, and the word mailman (term 3) occurred two words after *bit*. By storing the memory matrix in this way, the process of building memory matrices is achieved by searching the (T CF) pair list in the focus term's storage vector, to find a matching, ordered term-id. If a match is found then the co-occurrence frequency element of the pair is incremented.

Even for applications where the vocabulary is small and the context window radius is small, there will be a number of noisy terms that co-occur with many terms. These co-occurrences with noisy terms will quickly fill the storage vectors. To ensure the model is scalable and these noisy terms are managed a number of methods are used:

1. **Stop-list**: A stop-list is used to remove common high frequency terms.
2. **Co-occurrence frequency ratio cut-offs**: Frequency cut-offs are commonly used in semantic space models (Rohde *et al.*, 2006). Traditionally, the cut-off is applied to the collection frequency of a term. In contrast, our approach is to use a *co-occurrence frequency ratio* (CFR) cut-off, and apply it during the vocabulary building process when a storage vector is full and no match on term-id exists. The CFR is used to identify a (T CF) pair to be replaced, and is determined by comparing $\frac{CF}{F_w}$, where F_w is the collection frequency of the target term w, to a threshold value. If the CFR is below the threshold value the pair is moved to the end of the list and updated with the (T CF) details of this new co-occurrence.

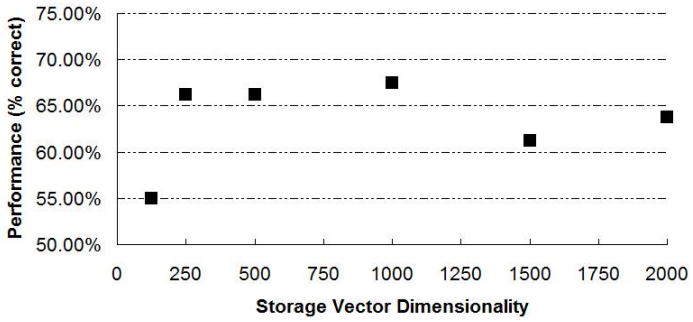

Figure 1: Performance on a synonym judgement task for storage vectors of various dimensions.

The success of these storage vector management methods can be evaluated by considering their impact on the model's performance on a synonym judgement task when the dimensionality of the storage vector is varied, as shown in figure 1. The task was taken from the synonym-finding part of the *Test of English as a Foreign Language* (TOEFL). TOEFL is a standardized test employed by American universities to evaluate foreign applicants' knowledge of the English language, and is further explained in section 4.2.

The superior performance achieved by the TE model for lower dimensionality vectors is particularly beneficial when contrasting computational complexity of the various models. Both BEAGLE and RI have been shown to achieve improved performance as the environment vector dimensionality is increased, often greater than 2000 (Sahlgren *et al.*, 2008). The relatively superior effectiveness for storage vectors with dimensionality between 250 and 1000, compared to those greater than 1000 may be due to our storage vector management technique removing low information items when the storage vector becomes full. At larger dimensions we predict that these low information terms are not removed and this may introduce noise into the TE model's synonym judgement.

The time complexity of the TE model's vocabulary building operation is determined by considering the worst case, in which the storage vector is full and a replacement operation is needed. In this case, the basic operation of the binding process becomes a full search of the (T CF) list, giving: $T_{TE}(n) = O(\frac{D_{SV}}{2})$, where D_{SV} is the storage vector dimensionality. For the synonym judgement task, optimal performance is when $D_{SV} = 1000$. The binding operation of the permuted RI model involves the summing of an environment vector with a context vector and a permuted environment vector with an order vector. Assumng the dimensionality of the vectors are D_{RI}, the time complexity of the permuted RI model would be $T_{RI}(n) = O(2.D_{RI})$, and from our discussion above $D_{RI} \geq 2000$. Therefore, our approach is argued to build the semantic space more efficiently than the permuted RI approach on the synonym judgement task.

4 Computing Word Meaning

One of the major advantages of our approach to encoding word order, compared to BEAGLE and the permuted RI model, is that it captures explicit word co-occurrence frequencies. This allows probabilistic measures to be used in addition to geometric measures when extracting information from the semantic space. The following section outlines two features that effectively measure the strength of syntagmatic or paradigmatic associations crucial in modelling word meaning.

When developing these measures we have tried to generalise the result to support the similarity between a sequence of priming words $Q = (q_1, \ldots, q_p)$ and any vocabulary term w. This was done so that the TE model could be more easily applied to a wider range

of information processing tasks. The memory matrix for the sequence of priming terms is formed by summing the memory matrices of these terms, $\boldsymbol{M_Q} = \boldsymbol{M_{q_1}} + \ldots + \boldsymbol{M_{q_p}}$.

4.1 A Measure of Syntagmatic Associations

One of the most popular measures of similarity between two geometric representations is the cosine of the angle formed between them. For the unique structure of the memory matrices used in our model, two interesting results were identified when developing a cosine measure: (i) that there exists a very efficient expression for calculating the cosine of the angle between memory matrices, and (ii) the resulting expression provides an excellent measure of the strength of syntagmatic associations between the terms.

For the extended general case and using linear algebra techniques, the cosine of the angle θ between memory matrices, $\boldsymbol{M_Q}$ and $\boldsymbol{M_w}$, is defined as:

$$\cos\theta = \frac{\sum_{\substack{j=1 \\ w\in Q}}^{N} s_w^2 f_{jw}^2 + \sum_{\substack{j=1 \\ j\neq w \\ w\in Q}}^{N} s_w^2 f_{wj}^2 + \sum_{\substack{i=q_1 \\ i\neq w}}^{q_m} (s_i^2 f_{iw}^2 + s_i^2 f_{wi}^2)}{\sqrt{\sum_{i=q_1}^{q_m} \left[\sum_{j=1}^{N} s_i^2 f_{ji}^2 + \sum_{\substack{j=1 \\ j\neq i}}^{N} s_i^2 f_{ij}^2\right]} \sqrt{\sum_{j=1}^{N} f_{jw}^2 + \sum_{\substack{j=1 \\ j\neq w}}^{N} f_{wj}^2}}, \tag{3}$$

where $q_1, \ldots, q_m$ are the list of m unique priming terms found in the sequence of all priming terms Q having $m \leq p$, s_i is the number of times term q_i appears in Q, f_{ab} is the co-occurrence frequency of term a appearing before term b in the vocabulary, f_{ba} is the co-occurrence frequency of term a appearing after term b.

The time complexity of this measure would appear to be linear with N, the size of the vocabulary. However, the storage vectors hold a maximum of $\frac{D_{SV}}{2}$ (T CF) pairs, where D_{SV} is the dimensionality of the storage vector. This means that the cosine measure has maximum time complexity when the storage vector is full and hence $T(n) = O(\frac{D_{SV}}{2}.|Q|)$, where $|Q|$ is the number of priming terms. An additional saving when computing the cosine scores for the vocabulary terms is gained by noting that the numerator in equation (3) will only be non-zero if term w has at least one interaction with a priming term $(q_1, \ldots, q_p)$, or is a priming term itself. Therefore, equation (3) will only need to be computed for term-ids found in the storage vectors of the priming terms, $(q_1, \ldots, q_p)$.

Nearest neighbours: Due to the unique construction of our memory matrices, it can be seen from equation (3) that the cosine measure extracts primarily syntagmatic associations of the priming terms and the focus term w. Access to syntagmatic relationships can be useful for many tasks including the identification of terms most likely to precede or succeed a target term. Within our representations, this can be achieved by isolating co-occurrence frequencies in the direction of interest, effectively setting elements to 0 on the row or column not of interest in the memory matrices, $\boldsymbol{M_Q}$ and $\boldsymbol{M_w}$. For example, to identify the term w that most likely precedes a sequence of priming terms Q, equation (3) becomes:

$$\cos_{\text{pr}}\theta = \frac{\sum_{\substack{j=1 \\ w\in Q}}^{N} s_w^2 f_{jw}^2 + \sum_{\substack{i=q_1 \\ i\neq w}}^{q_m} s_i^2 f_{iw}^2}{\sqrt{\sum_{i=q_1}^{q_m} \sum_{j=1}^{N} s_i^2 f_{ji}^2} \sqrt{\sum_{j=1}^{N} f_{jw}^2}}, \tag{4}$$

with an equivalent expression, using f_{wx} instead of f_{xw}, created to calculate most likely succeeding terms. Table 2 provides a list of most likely preceding and succeeding terms produced by the TE model for a list of target words identified in Jones and Mewhort (2007) for the BEAGLE model. The results illustrate the influence of the asymmetric nature of the memory matrices, and the effectiveness of the cosine measure to identify the strongest ordered syntagmatic associations.

Table 2: Top 6 lexical representations produced for a word preceding or succeeding a target word.

KING		PRESIDENT		WAR		SEA	
____ king	king ____	____ president	president ____	____ war	war ____	____ sea	sea ____
luther:0.419	jr:0.945	vice:0.905	roosevelt:0.948	civil:0.989	ii:0.918	mediterranean:0.995	level:0.972
martin:0.288	midas:0.695	elected:0.834	kennedy:0.927	world:0.851	ended:0.298	caribbean:0.857	anemone:0.315
dr:0.185	arthur:0.419	former:0.14	nixon: 0.876	revolutionary: 0.524	effort: 0.056	baltic:0.738	urchins:0.256
french:0.146	minos:0.307	new:0.07	johnson:0.613	spanish-american:0.306	began:0.038	caspian:0.714	captains:0.252
rex:0.03	queen:0.193	our:0.036	lincoln:0.522	during:0.122	between:0.024	aegean:0.675	gull:0.157
english:0.025	myron:0.165	twenty-seventh:0.012	carter:0.386	declare:0.085	broke:0.024	sargasso:0.592	gulls:0.154

4.2 A Measure of Paradigmatic Associations

One of the main advantages of our TE model, over BEAGLE and the permuted RI model, is the ability to capture explicit co-occurrence frequencies within the geometric representations. This result provides the model with the ability to use the element values of the geometric representations to calculate direct probabilistic measures between vocabulary terms. As an example, we developed an expression to estimate the strength of paradigmatic associations between a sequence of priming terms $Q = (q_1, \ldots, q_p)$ and a vocabulary term w. The measure is based on enhancing terms that co-occur with the same terms as Q, and is defined as:

$$P_{\text{par}}(w|Q) = \frac{1}{Z_{\text{par}}} \sum_{j=q_1}^{q_p} \sum_{i=1}^{N} \frac{f_{ij} f_{iw} + f_{ji} f_{wi}}{f_j f_w}, \tag{5}$$

where f_j is the vocabulary frequency of term j, f_{ji} is the ordered co-occurrence frequency of term j before term i, N is the size of the vocabulary, and $Z_{\text{par}} = \sum_{w \in V_k} \left[\sum_{j=q_1}^{q_p} \sum_{i=1}^{N} \frac{f_{ij} f_{iw} + f_{ji} f_{wi}}{f_j f_w} \right]$. Since the storage vector holds a maximum of $\frac{D_{SV}}{4}$ (T CF) pairs, the worst case time complexity of this paradigmatic measure is $T(n) = O(\frac{D_{SV}^2}{4}.|Q|)$, where D_{SV} is the dimensionality of the storage vector.

Synonym judgements: Paradigmatic associations are heavily used in the process of making synonym judgements. Therefore, we will evaluate our paradigmatic measure on the synonym-finding task in the TOEFL, after using the TASA (Touchstone Applied Science Associates, Inc.) corpus to build our semantic space. TASA contains 12-million words, and is a collection of English text articles that are reportedly equivalent to what the average college-level student has read in his or her lifetime. It has been extensively used to learn semantic relationships within semantic space models evaluated on TOEFL (Landauer and Dumais, 1997; Jones and Mewhort, 2007; Sahlgren *et al.*, 2008). In the synonym-finding part of TOEFL the participant is asked to choose one of four provided words as the most similar to the question word. It was reported that for a large sample of applicants to U.S. colleges, coming from non-English speaking countries, the average result on the synonym test was 51.6 items correct out of 80 (or 64.5%) (Landauer and Dumais, 1997).

A review of past papers using the TOEFL synonym test as a benchmark [1], suggests that the corpus used, preprocessing of documents and resulting vocabulary size may impact the performance achieved (Stone *et al.*, 2008). Therefore, comparisons of TOEFL performance between papers is likely unreliable. A more robust comparison may be achieved by

[1] http://aclweb.org/aclwiki

evaluating the models of interest on the same data configuration, hence we built BEAGLE and the permuted RI model.[2] In our experiments a 416 word stop list[3] was used, with the exception of the words *enough, often* and *alone*, which were present as a question or answer within TOEFL. We did not use any stemming on the vocabulary, however, the TOEFL question, *expeditiously*, was not found in the TASA corpus, whereas *expeditious* was, therefore that TOEFL question was updated to use *expeditious*. We also chose to remove TASA terms that contained numerics. These steps resulted in a vocabulary size of 134,054 unique terms. The performance achieved by each model is shown in figure 2.

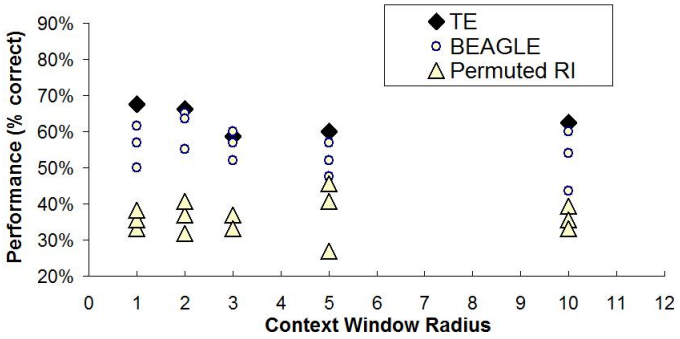

Figure 2: TOEFL performance for the Tensor Encoding, BEAGLE and permuted RI model.

Since BEAGLE and the permuted RI model use random environment vectors, a number of runs were performed to calculate the average score. The best average results were: (i) BEAGLE=61.25% (49/80) using a context window radius (cwr) of 2, and environment vector length (evl) of 2048, and (ii) permuted RI model=38% (30/80) using cwr=5 and evl=2,000. The best TE model result was 67.5% (54/80) using cwr=1 and a storage vector length of 1,000. The BEAGLE results were similar to those reported in Jones and Mewhort (2007), with any improvement likely due to the reduced context window length used in our experiments. The permuted RI model result is much lower than that reported in Sahlgren *et al.* (2008), possibly due to the difference in vocabulary size. Their TASA vocabulary was reduced to 74,100 terms by using stemming and high frequency cut-offs.

Addressing weaknesses in LSA: Landauer and Dumais (1997) indicated that some of the TOEFL errors produced by LSA, that were not made by students, may be attributed to the fact that LSA was more sensitive to paradigmatic associations, and not syntagmatic. For example, Perfetti (1998) commented that on the TOEFL, LSA chose *nurse* (0.47) over *doctor* (0.41) for the question word of *physician*. Even though this is Perfetti's selective example, we found that the TE model was more likely to choose *doctor* ($P(w|Q)$=**0.01926**) over nurse ($P(w|Q) = 0.01818$) for the same question.

5 Conclusions and Future Work

The aim of this paper has been to present a model of word meaning that goes beyond existing semantic space models by using Kronecker products to capture word order and co-occurrence information. Our TE model overcomes weaknesses in previous models attempting to encode greater structural information by reducing the information loss without computational cost. It also provides applications with more flexibility when extracting task specific semantic information without relying on pre-existing knowledge, like POS taggers.

[2] The permuted RI model functions were supplied by http://code.google.com/p/semanticvectors/
[3] Stoplist taken from the Lemur toolkit for information retrieval: http://www.lemurproject.org

The ability to extend the evaluation of this model to other information processing tasks, such as word sense disambiguation, query expansion, and document retrieval, is an area for future research. Another area for further investigation includes extending the current vocabulary binding process to form higher order tensors that would allow larger n-tuple associations to be encoded in the representations underpinning the semantic space. Using higher order TE models may have advantages similar to those highlighted by Baroni and Lenci (2010).

References

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

Holland, Norman N. 1992. *The Critical I.* Columbia University Press, New York, USA.

Jones, Michael N. and Douglas J. K. Mewhort. 2007. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114, 1–37.

Kanerva, Pentti, Jan Kristoferson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. *In Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, p. 1036.

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

Lund, K. and C. Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments and computers*, 28, 203–208.

McRoy, S. W. 1992. Using multiple knowledge sources for word sense discrimination. *Computational Linguistics*, 18(1), 1–30.

Mitchell, Jeff and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8), 1388–1429.

Perfetti, Charles A. 1998. The limits of co-occurrence: Tools and theories in language research. *Discourse Processes*, 25, 363–377.

Plate, Tony. 1991. Holographic reduced representations: Convolution algebra for compositional distributed representations. In *International Joint Conference on Artificial Intelligence*, pp. 30–35. Morgan Kaufmann.

Rapp, Reinhard. 2002. The computation of word associations: comparing syntagmatic and paradigmatic approaches. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, pp. 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Rohde, Douglas L. T., Laura M. Gonnerman, and David C. Plaut. 2006. An improved model of semantic similarity based on lexical co-occurence. *Communications of the ACM*, 8, 627–633.

Sahlgren, Magnus, Anders Holst, and Pentti Kanerva. 2008. Permutations as a means to encode order in word space. In V. Sloutsky, B. Love, and K. Mcrae, eds., *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pp. 1300–1305. Cognitive Science Society, Austin, TX.

Schütze, Hinrich. 1993. Word space. In *Advances in Neural Information Processing Systems 5*, pp. 895–902. Morgan Kaufmann.

Stone, Benjamin P., Simon J. Dennis, and Peter J. Kwantes. 2008. A systematic comparison of semantic models on human similarity rating data: The effectiveness of subspacing. In *Proceedings of the Thirteeth Conference of the Cognitive Science Society.* Cognitive Science Society.

Turney, Peter D. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pp. 905–912, Stroudsburg, PA, USA. Association for Computational Linguistics.

Turney, Peter D. and Patrick Pantel. 2010. From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, 37, 141–188, January.