The Efficiency of Corpus-based Distributional Models for Literature-based Discovery on Large Data Sets

Michael Symonds[1] **Peter Bruza**[1] **Laurianne Sitbon**[2]

[1] School of Information Systems, Queensland University of Technology,
[2] Computer Science Department, Queensland University of Technology,
Brisbane, Australia
Email: michael.symonds@qut.edu.au, p.bruza@qut.edu.au, laurianne.sitbon@qut.edu.au

Abstract

This paper evaluates the efficiency of a number of popular corpus-based distributional models in performing discovery on very large document sets, including online collections. Literature-based discovery is the process of identifying previously unknown connections from text, often published literature, that could lead to the development of new techniques or technologies. Literature-based discovery has attracted growing research interest ever since Swanson's serendipitous discovery of the therapeutic effects of fish oil on Raynaud's disease in 1986. The successful application of distributional models in automating the identification of indirect associations underpinning literature-based discovery has been heavily demonstrated in the medical domain. However, we wish to investigate the computational complexity of distributional models for literature-based discovery on much larger document collections, as they may provide computationally tractable solutions to tasks including, predicting future disruptive innovations.

In this paper we perform a computational complexity analysis on four successful corpus-based distributional models to evaluate their fit for such tasks. Our results indicate that corpus-based distributional models that store their representations in fixed dimensions provide superior efficiency on literature-based discovery tasks.

Keywords: Efficiency, literature-based discovery, corpus-based distributional models

1 Introduction

This paper examines, the often overlooked, impact of a model's computational complexity on the successful application to the task of *literature-based discovery* (LBD) on very large data sets. LBD relies on the identification of undiscovered connections between concepts in literature (including online document collections). These concepts are often linked indirectly via other concepts, as illustrated by the medical discovery process linking *illness A* and *drug C* depicted in Figure 1. This approach was first popularised by Swanson (1986) in discovering the link between *fish oil* (Drug C) and *Raynaud's disease* (Illness A).

The need for a complexity analysis of corpus-based models on the task of discovery, stems from: (i) the growing size of literature collections (especially those found online), and (ii) the low cost and flexibility of corpus-based models that do not rely on hand-crafted semantic resources, such as ontologies and thesauri. Over the past three decades cognitive science researchers have developed a class of corpus-based models, known as *corpus-based distributional models*, that automatically build representations of words from their occurrence patterns (distributions) in streams of natural language, hence their name distributional models. The most well-known of these is the *latent semantic analysis* (LSA) model (Landauer & Dumais 1997). Corpus-based distributional models have seen growing interest due to their low cost and proven effectiveness on a wide range of applications, including information retrieval (Turney & Pantel 2010, Symonds, Bruza, Zuccon, Koopman, Sitbon & Turner 2013).

The ability to efficiently and effectively perform discovery across multiple domains of knowledge (as represented by the document collections found on the web) is especially relevant in novel research areas, such as the detection of future disruptive innovations (Christensen 2006). Disruptive innovations are those which initially offer a lower performance according to the mainstream, however, offer some new performance attributes that make them prosper in a different market, during which time their performance in traditional markets improves to the point that they displaces the former technology. The mobile phone is an example of a disruptive innovation, as they initially offered poorer sound quality and were expensive. However, their advantage was their portability. As the sound quality improved and price dropped, they replaced the analogue phone.

The ability to identify future disruptive innovations may be possible through accessing many very large data sources, including patent information and online document collections (Daim et al. 2006). These data sources are likely to be very large, multi-lingual and given their nature (i.e., documenting new concepts) unlikely to have available hand-crafted semantic resources (such as dictionaries, thesauri and other hand-crafted ontologies). Therefore, LBD techniques that are underpinned by corpus-based distributional models are likely to be well suited to these emerging areas of research. Which corpus-based model would perform the best on discovery? This depends on the efficiency and effectiveness of a model. This research focuses on evaluating the efficiency of a number of successful, corpus-based distributional models for the task of discovery.

This paper is structured in the following way: (i) A review of current LBD practices, including the iden-

tification of relevant weaknesses beyond the growing complexity issues targeted in this work, (ii) A review of the complexity of four popular corpus-based distributional models and how they are positioned with respect to the existing weaknesses of LBD approaches, and (iii) A discussion of findings from the review and complexity analysis that can help researchers select corpus-based distributional models that are most likely to provide superior efficiency and effectiveness on the task of LBD.

2 Related Work

The two areas of work that underpin this research include: (i) literature-based discovery (LBD), and (ii) the use of corpus-based distributional models to identify potentially useful, undiscovered connections.

2.1 Literature-based Discovery

Since the serendipitous discovery of the therapeutic effects of fish oil on Raynaud's disease by scientist Don Swanson in 1986 (Swanson 1986), the field of literature-based discovery has seen strong interest, as evidenced by the increasing papers, conferences, workshops, books and reviews of LBD research (Weeber et al. 2005, Bruza & Weeber 2008). LBD aims to identify possible useful undiscovered connections between concepts in literature. This LBD process relies on being able to identify intermediary concepts, i.e., concepts that link two other concepts that are not directly linked to each other within the literature. For example concept A may be known to have an association with concept B (as demonstrated by their co-occurrence in one or many documents), while concept C may also be known to have an association with concept B. However, if A and C do not co-occur in any documents together their indirect association through B (the intermediary or bridging concept) may be unknown. If yet undiscovered, this link through B may provide researchers with valuable insights that could potentially lead to advances in technology or identification of disruptive innovations.

The identification of B within the LBD process allows for two modes of discovery, termed open and closed. Open discovery involves two steps: (i) starting with a known concept A, identify a limited number of possible intermediary concepts (i.e., B concepts that co-occur with A), and (ii) exploring the literature containing B concepts to identify potentially useful C concepts. In closed discovery, the process starts with a hypothesised connection between A and C, and an explanation for this observation is sought by finding an appropriate B that co-occurs with both A and C.

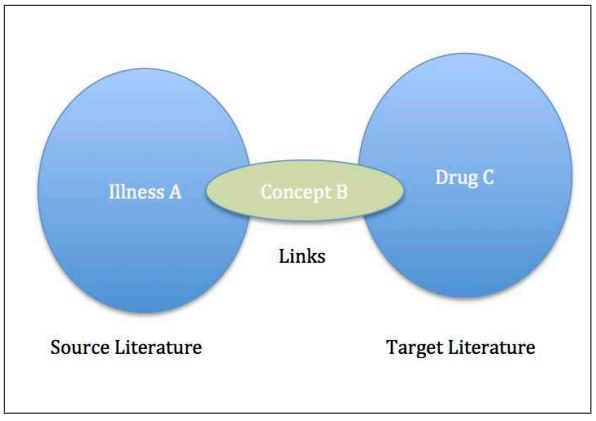

Figure 1: *Example of literature based discovery in the medical domain.*

Both of these discovery methods can lead to very large numbers of possibly useful undiscovered connections, especially within very large, modern corpora (e.g., medical or patent collections). To illustrate, consider that each concept in the corpus (referred to as A in the example above) may co-occur with thousands of other concepts (referred to as B). Given that B may also co-occur with thousands of other concepts (referred to as C) not seen with A, the number of possible useful connections may be in the order of hundreds of thousands. Therefore, the next critical step is to narrow the list of possibilities.

Early LBD researchers relied on human experts to reduce the list. However, this is a very costly process, and intractable in large modern data sets. Therefore, modern LBD researchers have relied on a number of scalable methods, based on distributional statistics, to limit the list of useful connections. Many of these rely on co-occurrence information alone (Gordon & Lindsay 1996, Gordon & Dumais 1998). However, within the medical domain researchers have argued that co-occurrence information on its own does not take advantage of higher order semantic relationships that exist between concepts (Cohen et al. 2012, Hristovski et al. 2006). One approach to extending distributional models has been to incorporate natural language tools (including ontological information) to produce a predictive LBD method that aims to identify discovery patterns (Hristovski et al. 2006). Other LBD researchers have taken these ideas and combined them with more efficient distributional approaches aimed at overcoming the computational complexity issues that arise with earlier distributional models (Cohen et al. 2012).

Two of the strongest criticisms of current LBD approaches is argued to stem from (i) the bias of distributional models toward high frequency concepts (Kostoff 2008) and (ii) the need for hand-crafted semantic resources, which may exist for medical LBD, but are not available for many other domains, and would be very expensive to create. With regard to the first criticism, Kostoff (2008) argues that high frequency concepts are less likely to be undiscovered, therefore, any system biased toward them is less likely to be effective at discovery. The solution proposed by Kostoff (2008) is to include more human intervention (primarily by authors in the fields in which the discovery process is undertaken). However, such human-dependent approaches, known as *literature related discoveries* (LRD (Kostoff 2008)), will naturally be more costly, especially as the literature for a field of knowledge naturally grows. Therefore, identifying best practice in LBD and extending these approaches to account for the weaknesses identified by researchers is required (i.e., not relying on hand-crafted semantic resources, and reducing the bias toward high frequency concepts). We argue that this is likely to be achieved using corpus-based distributional models.

2.2 Corpus-based Distributional Models

Corpus-based distributional models commonly use word order and co-occurrence information found in streams of natural language to build geometric or probabilistic representations of concepts. The premise of these models stems from the distributional hypothesis, which states that words with similar meaning will tend to co-occur with similar words (Harris 1954). Within distributional models,

the semantic associations that underpin the meaning of words can be modelled using measures of similarity, specific to the mathematical framework in which they are set. For example, in a geometric setting the strength of semantic associations can be measured from the distance between concepts in the space, hence their popular name *semantic space models*. In the past, corpus-based distributional models have been successfully applied to applications that involve relatively large data sets, including synonym judgement (Landauer & Dumais 1997, Symonds et al. 2011).

The advantage of corpus-based distributional models over those using hand-crafted semantic resources, is their reduced cost and flexibility in modelling concepts based on the context they are seen within the training corpora, as well as the ability for the model to be applied to document collections of any language. An LBD approach that relies solely on a corpus-based distributional model begins to address the concerns raised by Kostoff (2008) as long as it can be shown to have reduced bias to high frequency concepts. This point will be explicitly addressed in addition to the computational complexity calculations for each of the models reviewed in this work.

3 Computational Complexity

The implementation of the open discovery process using corpus-based distributional models can be broken down into the following computational steps:

1. Pre-processing of the documents (i.e., stemming, stopping, etc)
2. Building representations for vocabulary terms.
3. Retrieving terms based on the similarity of representations.

Within this work, we assume all models evaluated use the same methods to achieve step 1 (preprocessing). Therefore, our complexity analysis focuses on the costs of achieving steps 2 (building the representations) and 3 (computing similarity) for four successful corpus-based distributional models. The four models include (i) LSA (Latent Semantic Analysis (Landauer & Dumais 1997)), (ii) HAL (Hyperspace Analogue to Language (Burgess et al. 1998)), RI (Random Indexing (Kanerva et al. 2000, Karlgren & Sahlgren 2001)), and the TE model (Tensor Encoding (Symonds et al. 2011)). All of these approaches build representations for each term in the vocabulary of the document collection. Topic models, such as *latent dirichlet allocation* (LDA) (Blei et al. 2003)) and *probabilistic latent semantic analysis* (pLSA) (Hofmann 1999), were not included in this research as we were unable to find any previous examples using topic models for open discovery. This may be due to the reduced chance of discovery when using a limited set of latent topics in the discovery process. Increasing the number of topics is likely to have a significant impact on efficiency, in the case of LDA the complexity becomes NP-hard (Sontag & Roy 2011). A detailed investigation into the possible use of topic models for open discovery, and their efficiency is left for future work.

The complexity analysis in this work assumes that the document collection is stored on disk and that all steps of the LBD process can be achieved in main memory. For large data sets, the representations may not entirely fit within main memory. In this case it is assumed that a large main memory will cache a sufficiently large proportion of working data such that retrieval time is not affected. From an implementation point of view, however, the question of how to efficiently retrieve out of core representations is a question for future research. Storage complexity will be measured in terms of a number representation with 32 bits of precision, such as a 32 bit float or integer.

The analysis will also provide a computational estimate of performing LBD using each model on the MAREC patent document collection[1] (Table 1). MAREC is a static collection of over 19 million patent applications and granted patents from a number of international sources, spanning a range from 1976 to June 2008. As the MAREC dataset has a vocabulary size in the order of tens of millions, it is considered to be an example of a very large collection. It is also similar to collections found on the web as it covers multiple domains of knowledge in more than one language, and provides a reasonable chance of containing previously undiscovered links between concepts that could be used for discovering future disruptive innovations.

Collection	$\|D\|$	$\|V\|$	$\|C\|$
MAREC	19,386,697	74,547,422	65,611,683,654

Table 1: *Details of the MAREC document collection used as an example in the computational complexity analysis of each model. $|D|$ is the number of documents in the collection, $|V|$ represents the size of the vocabulary and $|C|$ represents the total number of terms in the collection.*

The complexity analysis considers storage complexity, denoted as $M(n)$ representing the memory requirements for a given input size n, and time complexity, denoted as $T(n)$ representing the worst case time complexity for a given input size n. The analysis begins by considering the complexity of LSA.

3.1 Latent Semantic Analysis (LSA)

LSA is probably the best known corpus-based distributional model. LSA builds latent representations of vocabulary terms from a full term-document matrix created from the training corpus (Landauer & Dumais 1997). Even though LSA uses a reduced matrix form to calculate the semantic similarity of vocabulary terms, the full term-document matrix needs to initially be constructed. Each term's vector representation is a row in the matrix whose elements are the frequency of the term in each document. For example, on the MAREC patent document collection (Table 1) the full term-document matrix would be a $75,000,000 \times 19,000,000$ matrix. The full term-document matrix can be obtained by building an index of the collection. LSA then applies a technique from linear algebra, known as *singular value decomposition* (SVD), to reduce the matrix to the k most significant latent terms (where k is the number of singular values to be used in computations). SVD is an expensive process, however, once it has been performed, only the reduced matrix needs to be used to perform similarity calculations between vocabulary terms, which is ultimately required when performing open discovery.

[1] http://www.ir-facility.org/prototypes/marec

3.1.1 Building Representations

From a storage complexity perspective, LSA still requires the full term-document matrix to be created, so initially it has a storage complexity of $M(n) = O(|V| \times |D|)$, where $|V|$ is the size of the vocabulary formed from the training document collection, and $|D|$ is the number of documents in the training collection. In the case of the MAREC data set, $M(n) = (75 \times 10^6)(19 \times 10^6) = 1.425 \times 10^{15}$.

From a time complexity perspective, the SVD process is the most costly, and has a time complexity of $T(n) = O(|V|^2|D| + |D|^3)$. Therefore, for the MAREC data set, $T(n) = (75 \times 10^6)^2(19 \times 10^6) + (19 \times 10^6)^3 = 1.14 \times 10^{23}$.

3.1.2 Computing Similarity between Terms

Similarity calculations between vocabulary terms are often achieved using geometric measures such as the cosine metric or a Minkowski norm (i.e., city block or Euclidean distance) to compare the vector representations in the latent concept space. The cosine measure has demonstrated robust effectiveness on a number of tasks, including similarity judgement (Landauer & Dumais 1997).

For computing step 3 of the open discovery mode (i.e., retrieving terms based on their similarity) in LSA the B concepts in the vocabulary can be found by (i) listing the nearest neighbours to the A concept (i.e., performing a cosine measure across the vocabulary with A), and then (ii) performing a cosine similarity between each potential B concept and those found in a reduced vocabulary created be removing all terms that appeared in any of the documents A had appeared in. The time complexity of step (i) would be $T(n) = O(|V|(d_l))$, where (d_l) is the dimensionality of the reduced vectors. Assuming there are b intermediary concepts (i.e., highest ranked B concepts), then the overall time complexity of performing open discovery with the LSA model is $T(n) = O(|V|(d_l)+(b)|V_c|(d_l))$, where $|V_c|$ is the number of vocabulary terms that did not co-occur with A. Setting $d_l = 300$, $b = 1,000$ and $|V_c| = 0.8 \times |V|$, the worst case time complexity of computing C concepts for open discovery in the MAREC dataset is $T(n) = (7.5 \times 10^7)(300) + (1 \times 10^3)(0.8 \times 7.5 \times 10^7)(300) = 1.8 \times 10^{13}$.

It is worth noting that the second step in the retrieval process may require a further SVD operation (for each A concept of interest) to be performed on the term-document matrix containing all possible C concepts (i.e., that did not co-occur in documents containing concept A). This would further increase the time complexity of building the semantic space for the LSA model. LSA has demonstrated effective performance on tasks, such as synonym judgement (Landauer & Dumais 1997), that evaluate associations between low frequency concepts, and hence LSA is likely to address the criticism raised by Kostoff (2008) that for open discovery effective distributional models should not be biased toward high frequency terms.

3.2 Hyperspace Analogue to Language (HAL)

The HAL model creates a term-term co-occurrence matrix by moving a sliding context window across the training corpus and collecting co-occurrence frequencies between terms (Burgess et al. 1998). To illustrate, consider the HAL matrix shown in Table 2, which was created for the toy sentence *A dog bit the mailman*, using a sliding context window of length 5 (i.e., 2 words either side of the focus word). The co-occurrence information preceding and following each word are recorded separately by the row and column vectors. The values assigned to each co-occurrence are scaled by their distance from the focus word, with words next to the focus word given a value of 2 (when a context window length of 5 is used), and those at the edge of the window scaled by 1.

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

Table 2: *HAL matrix for the example sentence**A dog bit the mailman**.*

3.2.1 Building Representations

This $|V| \times |V|$ matrix means that the storage complexity of the HAL model is $M(n) = O(|V| \times |V|) = O(|V|^2)$. For the MAREC corpora, $M(n) = (7.5 \times 10^7)^2 = 5.6 \times 10^{15}$. The time to build the vocabulary representation within the HAL model involves incrementing the vector elements as the context window is moved across the documents, and has a worst case time complexity equal to $T(n) = O(|C| \times s)$, where $|C|$ is the total number of terms in the collection and s is the size of the context window. For the MAREC data set where $|C| = 6.6 \times 10^{10}$, we set $s = 5$, $T(n) = (6.6 \times 10^{10}) \times 5 = 3.3 \times 10^{11}$.

3.2.2 Computing Similarity between Terms

Computing step 3 of the open discovery process (retrieving terms based on similarity) in HAL involves, (i) identifying the B concepts in the vocabulary by listing the nearest neighbours to the A concept (i.e., performing a cosine measure across the vocabulary with A), and then (ii) performing a cosine similarity between each potential B concept and those found in a reduced vocabulary created be removing all concepts that appeared in any document that contained concept A. For the HAL model, the time complexity of step (i) would be $T(n) = O(|V||V|) = O(|V|^2)$. Assuming there are b intermediary concepts (i.e., highest ranked B concepts), then the overall time complexity of performing open discovery with the HAL model is $T(n) = O(|V|^2 + (b)|V_c||V|)$, where $|V_c|$ is the number of vocabulary terms that did not co-occur with A. Setting $b = 1,000$ and $|V_c| = 0.8 \times |V|$, the worst case time complexity of computing C concepts for open discovery in the MAREC dataset is $T(n) = (7.5 \times 10^7)^2 + (1 \times 10^3)(0.8 \times 7.5 \times 10^7)(7.5 \times 10^7) = 4.5 \times 10^{18}$.

To reduce the computational complexity of the HAL model, researchers have previously only retained the dimensions of the k most frequent terms in the vocabulary, where k is often around 100,000 (Bullinaria & Levy 2007). In this way, the storage complexity becomes, $M(n) = O(|V| \times k)$. However, given low frequency concepts are most likely to produce useful discoveries (Kostoff 2008), ignoring them is unlikely to provide the most effective LBD outcomes. Therefore, the HAL complexity associated with the full term-term matrix will be used for the comparative analysis in this work.

3.3 Random Indexing (RI)

RI is a more recent semantic space approach that creates fixed dimension vector representations, the size of which are independent of the number of terms in the vocabulary. These representations are created from an approximately orthogonal basis formed by assigning each term a random environment vector of dimensionality d_c, where $d_c \ll |V|$. The final representations for vocabulary terms are created by summing the environment vectors of terms that co-occur within a sliding context window that is moved across the corpus.

3.3.1 Building Representations

Fixing the dimensions of the representations reduces the storage complexity of the model to $M(n) = O(2|V|(d_c))$, where d_c is the dimensionality of the context vectors (and environment vectors). Recent LBD research (Cohen et al. 2012), using an enhanced RI model based on bit vectors and incorporating a NLP resource, fixed the dimensionality of the storage vectors to 32,000 bits, which assuming 32 bit number representations, makes the d_c for storage complexity effectively 1,000 stored integers. We will assume these dimensions are effective on the MAREC data set and ignore the impact of pre-processing using the NLP resource for our complexity analysis. The resulting storage complexity of RI for the MAREC dataset would be $M(n) = 2 \times (7.5 \times 10^7)(1 \times 10^3) = 1.5 \times 10^{11}$. Which is almost 30,000 times less than the memory footprint for HAL.

The time complexity of building the RI semantic space is similar to HAL, except that all elements of the vectors must be summed (c.f., as opposed to incrementing a single element value), as the RI model being considered is based on dense distributed representations in which the dimensions of the vectors do not relate to a term-id or document-id. The worst case time complexity of building the representations within the RI model would be $T(n) = O(|C|(s)(d_c))$, which is d_c times greater than the HAL model. Setting $d_c = 32,000$ (as each bit needs to be considered in building and comparing vectors) and $s = 5$, the time complexity of the RI model to build the vocabulary for the MAREC training collection becomes, $T(n) = (6.6 \times 10^{10}) \times 5 \times (3.2 \times 10^5) = 1.1 \times 10^{17}$.

3.3.2 Computing Similarity between Terms

When performing similarity within the RI model, a geometric measure such as the cosine metric or Minkowski measure is often used, as done in past research applying RI to LBD (Cohen et al. 2012). For the RI model using bit-vectors, as used in Cohen et al. (2012), the cosine similarity measure is actually the hamming distance, and the process for computing step 3 of the open discovery process using RI involves (i) comparing all vocabulary terms to the representation of A (i.e., $T(n) = O(|V|(d_c))$, and (ii) computing the similarity of each B concept with all vocabulary terms that did not co-occur in documents that contained A (i.e., $T(n) = (b)|V_c|(d_c)$, assuming there are b intermediary concepts; i.e., highest ranked B concepts). Therefore, the time complexity of performing the retrieval process in open discovery with the RI model is $T(n) = O(|V|(d_c) + (b)|V_c|(d_c))$. Setting $d_c = 32,000$, $b = 1,000$ and $|V_c| = 0.8 \times |V|$, the worst case time complexity of computing C concepts for open discovery in the MAREC dataset is $T(n) = (7.5 \times 10^7)(3.2 \times 10^5) + (1 \times 10^3)(0.8 \times 7.5 \times 10^7)(3.2 \times 10^5) = 1.92 \times 10^{16}$. .

RI models used in semantic space research often require a form of frequency cut-off to be applied to achieve superior task effectiveness (Sahlgren et al. 2008, Karlgren & Sahlgren 2001, Cohen et al. 2012). Frequency cut-offs are often used to remove very high frequency terms, however, they can also be used to remove low frequency terms (Karlgren & Sahlgren 2001). Therefore, RI may have difficulty addressing the concern raised by Kostoff (2008) relating to the bias toward high frequency terms argued to exist in current LBD approaches using distributional models.

3.4 The Tensor Encoding (TE) model

The TE model is a recent model of word meaning that has demonstrated superior effectiveness over a state-of-the-art HAL-based model on a number of semantic tasks, including synonym judgement and the similarity judgement of medical concepts (Symonds et al. 2011, 2012).

3.4.1 Building Representations

The TE model builds tensor representations for vocabulary terms through a unique binding process. These sparse tensor representations are stored in low-dimensional storage vectors, whose dimensionality is independent of the vocabulary size.

To demonstrate, consider the construction of a vocabulary term *bit* for the following example sentence, *a dog bit the mailman*, and the resulting vocabulary terms and environment vectors in Table 3.[2]

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

Table 3: *Example vocabulary for the sentence:* ***A dog bit the mailman***

For the second-order TE model the representations are constructed by summing the proximity-scaled outer products of the environment vectors found within a sliding context window moved over the text. To demonstrate, consider the memory matrices created by the TE model's second order binding process for vocabulary term 2 (*bit*) where a sliding context window of radius 2 is chosen:

$$\overbrace{A_s \quad dog \quad [bit] \quad the_s \quad mailman}$$

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

The matrix representations for *bit* can be stored efficiently in the following *storage vector* (SV):

$$\text{SV}_{\text{bit}} = [(-1 \quad 2) \quad (3 \quad 1)], \quad (2)$$

[2] *A* and *the* are considered to be stop-list words (noisy, low information terms that are ignored) and hence are not included in the vocabulary in Table 3.

where parenthesis have been added to illustrate implicit grouping of (T CF) pairs, where T is the term-id of the co-occurring term with term w and CF is the cumulative, proximity-scaled, co-occurrence frequency of T with w, and $w = 2$ in this example, as bit is the second term in the vocabulary). The sign of T (term-id) indicates the word order of T with w. The information in this vector can be used to reconstruct the memory matrix using the following process:

1. If the term Id (T) is positive, the CF value is located at row w, column T in the memory tensor. Otherwise, the CF value is located at row T, column w.

At an implementation level, the construction of the second-order representations can be efficiently achieved using fixed dimension storage vectors and the following process:

1. For each co-occurrence with target term w, search the storage vector (SV_w) for a matching T value and its sign to ensure it occurs in the same word order with w.
2. If a match is found then, the CF element of the pair is increased by the scaled, co-occurrence frequency of w with T within the current context window. End process.
3. If no match is found then, check if the storage vector is full
4. If the storage vector is full then, the first low information pair in the storage vector should be removed and the new pair added to the end of the storage vector.
5. If the vector is not full then add the new pair to the end of the storage vector.

The removal of the first, low information pair in the storage vector, when the vector is full, applies a form of compression to the model. This compression has been argued to reduce the noise in the representations and leads to improvements in task effectiveness at lower dimensions (Symonds et al. 2011).

The storage complexity of the TE model is $M(n) = O(|V|(d_{sv}))$, where d_{sv} is the dimensionality of the storage vectors. The TE model has demonstrated superior effectiveness for storage vectors of 1,000 dimensions (i.e., $d_{sv} = 1,000$) on a number of semantic tasks (Symonds et al. 2011, 2012). We will assume this dimensionality is effective on the MAREC data set. For the MAREC training collection, the storage complexity of the TE model would be $M(n) = (7.5 \times 10^7)(1 \times 10^3) = 7.5 \times 10^{10}$, which is half that of Cohen's RI model and 100,000 times less than HAL.

The time complexity in building the TE model's representations is similar to HAL, except that the *tensor memory compression* technique is needed to remove low information co-occurrences when the storage vectors are full. The worst case time complexity of the TE model's vocabulary building process involves a full search of the storage vectors, and therefore is $T(n) = O(|C| \times s \times \frac{d_{sv}}{2})$. It is $\frac{d_{sv}}{2}$ as only half of the storage vector contains co-occurrence frequencies, the other half contain co-occurring term ids (refer to Symonds, Zuccon, Koopman, Bruza & Sitbon (2013)). This is $\frac{d_{sv}}{2}$ times more than the HAL building process, where typically $d_{sv} \approx 1,000$. Setting $d_{sv} = 1,000$, the time complexity of the TE model to build the vocabulary for the MAREC training collection becomes, $T(n) = (6.6 \times 10^{10}) \times 5 \times \frac{1 \times 10^3}{2} = 3.3 \times 10^{13}$.

3.4.2 Computing Similarity between Terms

When performing similarity within the TE model, two measures, modelling two types of word associations, known as syntagmatic and paradigmatic, are used and their scores interpolated. Within structural linguistics, syntagmatic and paradigmatic associations are used to induce the meaning of a word. Syntagmatic associations exist between concepts that are more likely to occur near each other in a document than by chance (e.g., *sun-hot*). While paradigmatic associations exist between concepts that are able to replace each other in a sentence without effecting the acceptability of the sentence (e.g., synonyms, or related verbs like *eat-drink*). It is the paradigmatic associations which have the greatest function for the task of LBD, because, in effect concepts A and C have a paradigmatic association via their common neighbour B. The syntagmatic association for A and C in LBD should be zero (i.e., never seen together in a document). This makes the TE model well adapted to the task of performing LBD, as both steps in the open discovery mode, discussed in Section 3.1 can be completed in one formalism. The TE model's formalism can be expressed as a conditional probability of concept C being suggested as a useful connection for a given A concept:

$$P(C|A) = \gamma S_{\text{par}}(A, C) + (1 - \gamma) S_{\text{syn}}(A, C), \quad (3)$$

where γ is a mixing parameter that combines the measures of paradigmatic and syntagmatic associations. The paradigmatic measure proposed by the TE research (Symonds et al. 2012) shows how the indirect relationship between concept A and C can be modelled via B concepts:

$$s_{\text{para}}(A, B) = \sum_{i \in V} \frac{f_{\overline{B_i A}} \cdot f_{\overline{B_i C}}}{\max(\, f_{\overline{B_i A}}\,,\, f_{\overline{B_i C}}\,,\, f_{\overline{CA}}\,)^2}, \quad (4)$$

where $f_{\overline{B_i A}}$ is the unordered co-occurrence frequency of concepts B_i and A, and the term $f_{\overline{CA}}$ in the denominator penalises the paradigmatic score if A and C have a strong syntagmatic association.

For the TE model, the time complexity of the retrieval process would be the time complexity of the syntagmatic measure and paradigmatic measures combined. Using the syntagmatic and paradigmatic measures outlined in previous TE research (Symonds et al. 2012), the time complexity to perform the retrieval process would be $T(n) = O(\frac{(d_{sv})^2}{4} + \frac{(d_{sv})^3}{8})$. For the MAREC data set, the worst case time complexity of computing the C concepts would be $T(n) = \frac{(1 \times 10^3)^2}{4} + \frac{(1 \times 10^3)^3}{8} = 1.25 \times 10^8$. Two orders faster than HAL and RI. The substantially reduced time complexity in calculating the similarity of terms within the TE model stems from the fact that the time complexity of the TE model is independent on the vocabulary size. It is the only model in our investigation with this property and whose time complexity advantage over other models would increase as the vocabulary size of the collection increases. This result is achieved by the TE model because (i) the two steps of the retrieval process can be computed in one step within the TE model, and (ii) only terms in a small set of storage vectors need be considered in the calculations as terms not in the storage vector of A are considered to have no syntagmatic association with A, and all terms not in the storage vectors of terms syntagmatically related to A are considered to have no paradigmatic associations with A.

Past research using the TE model to perform synonym judgement has shown that low frequency concepts are not discriminated against (Symonds et al. 2012). This indicates that the compression within the TE model and the similarity measures appear to effectively manage frequency bias, and hence the concern raised by Kostoff (2008) for the task of LBD.

4 Discussion

The computational complexity of each model for the task of LBD is shown in Table 4, along with its efficiency on the MAREC document collection. The important finding from Table 4 is that the time complexity of computing similarity within the TE model is independent of the vocabulary size ($|V|$). This means that as the vocabulary size increases, the time complexity for computing similarities within the TE model does not. As computing similarities is likely to be performed many more times than building the vocabulary (which only occurs once), this property of the TE model becomes more important.

The superior overall efficiency of the TE model can more easily be seen when the time complexities for building the vocabulary representations and computing the open discovery process for each of the four models are graphed (Figure 2).

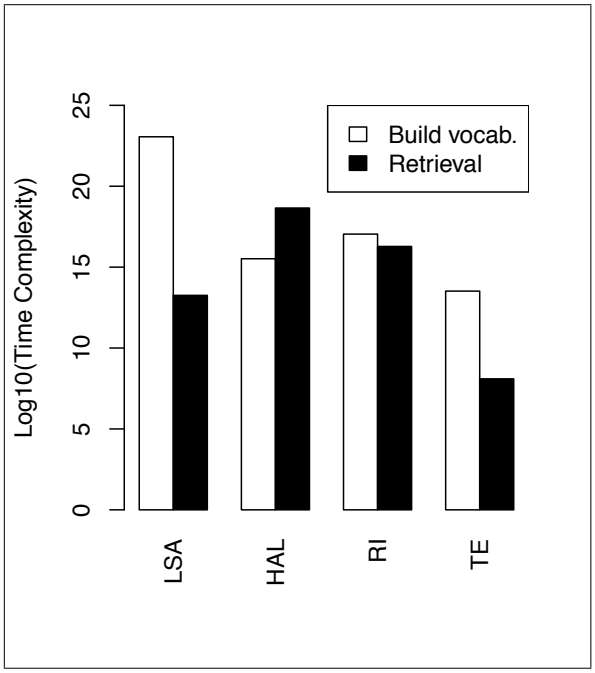

Figure 2: *Efficiency comparison of LSA, HAL, RI and TE on the MAREC dataset when considering the time complexity of building the vocabulary representations (Build vocab.) and computing the similarity of terms involved in the open discovery process (Retrieval).*

In the case where more documents are added to the collection dynamically, all models are reported to be able to update the representations with little additional overhead. It is also worth noting that the vocabulary size of the MAREC dataset was computed assuming only representations for single word terms were required. However, research indicates that including multi-word concepts in any semantic model is likely to be of value due to the compositional nature of meaning (Grefenstette & Sadrzadeh 2011). Including multi-word terms to the vocabulary and building representations for each will impact the complexity values above, through an increased in vocabulary size ($|V|$). This again highlights the benefits of using a corpus-based model whose complexity in computing similarity between vocabulary terms is independent on $|V|$.

4.1 The TE Model's Paradigmatic Measure

An initial investigation into the effectiveness of the TE model's paradigmatic measure to identify terms that do not display syntagmatic associations was tested on a data set based on the TREC 2011 MedTrack collection which consisted of clinical patient records. Following the procedure outlined by Koopman et al. (2012) the original textual documents were translated into UMLS medical concept identifiers using MetaMap, a biomedical concept identification system (Aronson & Lang 2010). After processing, the individual documents contained only UMLS concept ids. For example, the phrase *congestive heart failure* in the original document will be replaced with *C0018802* in the new document.

The TE model was then run to build representations, and then a number of sample terms, reported in past LBD research, were investigated. These included *Raynaud's disease* (C0034734) and *Migraine* (C0149931). The investigation demonstrated that of the top 800 concepts suggested by the TE model's paradigmatic measure (Equation (4)) for the concept (C0034734) representing the term *Raynaud's disease*, only 72 words (i.e., 9%) displayed any syntagmatic association with C0034734. The paradigmatic measure could be easily modified, with minimal impact on efficiency, to ensure any terms displaying syntagmatic association (i.e., exist in the storage vector of the target term) receive a paradigmatic score of zero.

Collection	$\|D\|$	$\|V\|$	$\|C\|$
Medline Concept	17,198	54,546	94082094

Table 5: Details of the reduced medline document collection, based on the TREC'11 MedTrack task, used to evaluate the initial effectiveness of the TE model in performing LBD.

This prototype investigation into the use of the TE model for LBD also found magnesium was returned for a target term of migraine. This demonstrates early support for the potential effectiveness of the TE model on the task of open discovery.

4.2 Heterogenous LBD

There is a growing trend for automated LBD models to be used in conjunction with other manual discovery processes across multiple literature sources (Kostoff 2008). The ability for automated tools, like distributional models to perform LBD across different knowledge sources, known as heterogenous LBD, may allow even faster progress in these areas. Heterogenous LBD would entail even larger combined data sets and increase the importance of automated tools being efficient. An emerging form of heterogeneous LBD is *literature related discovery and innovation* (LRDI). LRDI integrates LBD with innovation.e.g., re-invigorating prior art (Kostoff 2012).

	Model	Complexity	For MAREC collection
Storage Complexity	LSA	$M(n) = O(\|V\|\|D\|)$	$M(n) = 1.4 \times 10^{15}$
	HAL	$M(n) = O(\|V\|^2)$	$M(n) = 5.6 \times 10^{15}$
	RI	$M(n) = O(2\|V\|d_c)$	$M(n) = 1.5 \times 10^{11}$
	TE	$M(n) = O(\|V\|d_{sv})$	$M(n) = 7.5 \times 10^{10}$
Time complexity for building the vocabulary	LSA	$T_b(n) = O(\|V\|^2\|D\|)$	$T_b(n) = 1.14 \times 10^{23}$
	HAL	$T_b(n) = O(\|C\|(s))$	$T_b(n) = 3.3 \times 10^{11}$
	RI	$T_b(n) = O(\|C\|(s)(d_c))$	$T_b(n) = 1.1 \times 10^{17}$
	TE	$T_b(n) = O(\|C\|(s)(\frac{d_{sv}}{2}))$	$T_b(n) = 3.3 \times 10^{13}$
Time complexity for computing retrieval	LSA	$T_s(n) = O(\|V\|(d_l) + (b)\|V_c\|(d_l))$	$T_s(n) = 1.8 \times 10^{13}$
	HAL	$T_s(n) = O(\|V\|^2 + (b)\|V_c\|\|V\|)$	$T_s(n) = 4.5 \times 10^{18}$
	RI	$T_s(n) = O(\|V\|(d_c) + (b)\|V_c\|(d_c))$	$T_b(n) = 1.9 \times 10^{16}$
	TE	$T_s(n) = O(\frac{(d_{sv})^2}{2} + \frac{(d_{sv})^3}{4})$	$T_b(n) = 1.25 \times 10^8$

Table 4: *Complexity of the* ***Latent Semantic Analysis*** *(LSA),* ***Hyperspace Analogue to Language*** *(HAL),* ***Random Indexing*** *(RI), and the* ***Tensor Encoding*** *(TE) model for performing the building and retrieval steps for open discovery. Where $|V|$ is the size of the vocabulary ($|V| = 7.5 \times 10^6$ for MAREC), $|D|$ is the number of documents in the collection ($|D| = 6.6 \times 10^{10}$ for MAREC), d_l is the number of singular values used by LSA ($d_l = 300$), s is the size of the context window ($s = 5$), d_c is the dimensionality of the RI context vectors (d_c is equal to 1,000 and 32,000 for storage and time complexity, respectively), and d_{sv} is the dimensionality of the TE storage vectors ($d_{sv} = 1,000$).*

5 Conclusion

This paper has provided a computational complexity analysis of four successful corpus-based distributional models on the task of open discovery. These models provide a cost-effective method of identifying potentially novel, undiscovered connections between concepts within large document collections, such as those found online, and used within tasks such as *literature-based discovery* (LBD). These discoveries may ultimately lead to technological breakthroughs, or the ability to identify future disruptive innovations.

Our analysis finds that distributional approaches that store representations in fixed dimensions have a smaller memory footprint, and can allow faster computation of associations between vocabulary terms. Of particular significance is the finding that the TE model is well adapted to the task of open discovery in LBD due to its efficient method of storing representations and computing similarities from these representations. These features allow the process of open discovery to be computed with an efficiency that is independent of the vocabulary size. The findings of this work motivate a future evaluation of the TE model performing LBD based tasks, such as the discovery component of the emerging field of literature related discovery and innovation (LRDI).

The computational analysis carried out in this work contributes to the field of LBD, and more broadly information retrieval, by providing insights into the effectiveness of corpus-based models, which allows a more complete consideration of model's performance to be achieved.

6 Acknowledgements

I would like to thank the QUT HPC team for the use of the computational resources to perform the initial experiments. Thanks also to Lance De Vine for feedback on this paper.

References

Aronson, A. R. & Lang, F.-M. (2010), 'An overview of MetaMap: historical perspective and recent advances', *JAMIA* **17**(3), 229–236.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent Dirichlet Allocation', *J. Mach. Learn. Res.* **3**, 993–1022.

Bruza, P. & Weeber, M. (2008), *Literature-based Discovery*, 1 edn, Springer Publishing Company, Incorporated.

Bullinaria, J. A. & Levy, J. P. (2007), 'Extracting Semantic Representations from Word Co-occurrence Statistics: A Computational Study', *Behavior Research Methods* **39**, 510–526.

Burgess, C., Livesay, K. & Lund, K. (1998), 'Explorations in Context Space: Words, Sentences, Discourse', *Discourse Processes* **25**(2/3), 211–257.

Christensen, C. M. (2006), 'The ongoing process of building a theory of disruption', *Journal of Product Innovation Management* **23**(1), 39–55.

Cohen, T., Widdows, D., Schvaneveldt, R. W., Davies, P. & Rindflesch, T. C. (2012), 'Discovering Discovery Patterns with Predication-based Semantic Indexing', *Journal of Biomedical Informatics* **45**(6), 1049 – 1065.

Daim, T. U., Rueda, G., Martin, H. & Gerdsri, P. (2006), 'Forecasting Emerging Technologies: Use of Bibliometrics and Patent Analysis', *Technological Forecasting and Social Change* **73**(8), 981 – 1012.

Gordon, M. D. & Dumais, S. (1998), 'Using Latent Semantic Indexing for Literature based Discovery', *Journal of the American Society for Information Science* **49**(8), 674–685.

Gordon, M. D. & Lindsay, R. K. (1996), 'Toward Discovery Support Systems: A Replication, Re-Examination, and Extension of Swanson's Work on Literature-Based Discovery of a Connection between Raynaud's and Fish Oil', *Journal of the American Society for Information Science (1986-1998)* **47**(2), 116.

Grefenstette, E. & Sadrzadeh, M. (2011), 'Experimental Support for a Categorical Compositional Distributional Model of Meaning', *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* .

Harris, Z. (1954), 'Distributional Structure', *Word* **10**(23), 146–162.

Hofmann, T. (1999), Probabilistic Latent Semantic Analysis, *in* 'In Proc. of Uncertainty in Artificial Intelligence, UAI99', pp. 289–296.

Hristovski, D., Friedman, C., Rindflesch, T. C. & Peterlin, B. (2006), Exploiting semantic relations for literature-based discovery., *in* 'AIMA Annual Symposium proceedings', AIMA, Bethesda, USA, pp. 349–353.

Kanerva, P., Kristoferson, J. & Holst, A. (2000), 'Random Indexing of Text Samples for Latent Semantic Analysis', *Proceedings of the 22nd Annual Conference of the Cognitive Science Society* p. 1036.

Karlgren, J. & Sahlgren, M. (2001), From Words to Understanding, *in* 'In Uesaka, Y., Kanerva, P. and Asoh, H. (Eds.): Foundations of Real-World Intelligence', CSLI Publications, pp. 294–308.

Koopman, B., Zuccon, G., Bruza, P., Sitbon, L. & Lawley, M. (2012), An Evaluation of Corpus-driven Measures of Medical Concept Similarity for Information Retrieval, *in* 'The 21st ACM International Conference on Information Knowledge Management 2012', pp. 2439–2442.

Kostoff, R. N. (2008), 'Literature-Related Discovery (LRD): Introduction and background ', *Technological Forecasting and Social Change* **75**(2), 165 – 185.

Kostoff, R. N. (2012), 'Literature-related discovery and innovation - update', *Technological Forecasting and Social Change* **79**(4), 789 – 800.

Landauer, T. K. & Dumais, S. T. (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.

Sahlgren, M., Holst, A. & Kanerva, P. (2008), 'Permutations as a Means to Encode Order in Word Space', *Proceedings of the 30th Annual Meeting of the Cognitive Science Society* pp. 23–26.

Sontag, D. & Roy, D. (2011), Complexity of inference in latent dirichlet allocation, *in* J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira & K. Weinberger, eds, 'Advances in Neural Information Processing Systems 24', MIT Press, pp. 1008–1016.

Swanson, D. R. (1986), 'Fish oil, Raynaud's Syndrome, and Undiscovered Public Knowledge', *Perspectives in Biology and Medicine* **30(1)**, 7–18.

Symonds, M., Bruza, P. D., Sitbon, L. & Turner, I. (2012), A Tensor Encoding Model for Semantic Processing, *in* 'Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM'12)', ACM, New York, NY, USA, pp. 2267–2270.

Symonds, M., Bruza, P. D., Zuccon, G., Koopman, B., Sitbon, L. & Turner, I. (2013), 'Automatic Query Expansion : a Structural Linguistic Perspective', *Journal of the American Society for Information Science and Technology* .

Symonds, M., Bruza, P., Sitbon, L. & Turner, I. (2011), Modelling Word Meaning using Efficient Tensor Representations, *in* 'Proceedings of the 25th Pacific Asia Conference on Language, Information, and Computation (PACLIC'11)', pp. 313–322.

Symonds, M., Zuccon, G., Koopman, B., Bruza, P. D. & Sitbon, L. (2013), Term Associations in Query Expansion : a Structural Linguistic Perspective, *in* 'ACM International Conference on Information and Knowledge Management (CIKM'13)', ACM, San Francisco, CA.

Turney, P. D. & Pantel, P. (2010), 'From Frequency to Meaning: Vector Space Models of Semantics', *Journal of Artificial Intelligence Research* **37**, 141–188.

Weeber, M., Kors, J. A. & Mons, B. (2005), 'Online Tools to Support Literature-based Discovery in the Life Sciences', *Briefings in Bioinformatics* **6**(3), 277–286.