Discovery of Related Terms in a corpus using Reflective Random Indexing

Venkat Rangan

Clearwell Systems, Inc.

venkat.rangan@clearwellsystems.com

ABSTRACT

A significant challenge in electronic discovery is the ability to retrieve relevant documents from a corpus of unstructured text containing emails and other written forms of human-to-human communications. For such tasks, recall suffers greatly since it is difficult to anticipate all variations of a traditional keyword search that an individual may employ to describe an event, entity or item of interest. In these situations, being able to automatically identify conceptually related terms, with the goal of augmenting an initial search, has significant value. We describe a methodology that identifies related terms using a novel approach that utilizes Reflective Random Indexing and present parameters that impact its effectiveness in addressing information retrieval needs for the TREC 2010 Enron corpus.

1. Introduction

This paper examines reflective random indexing as a way to automatically identify terms that co-occur in a corpus, with a view to offering the co-occurring terms as potential candidates for query expansion. Expanding a user's query with related terms either by interactive query expansion [1, 5] or by automatic query expansion [2] is an effective way to improve search recall. While several automatic query expansion techniques exist, they rely on usage of a linguistic aid such as thesaurus [3] or concept-based interactive query expansion [4]. Also, methods such as ad-hoc or blind relevance feedback techniques rely on an initial keyword search producing a top-n results which can then be used for query expansion.

In contrast, we explored building a semantic space using Reflective Random Indexing [6, 7] and using the semantic space as a way to identify related terms. This would then form the basis for either an interactive query expansion or an automatic query expansion phase.

Semantic space model utilizing reflective random indexing has several advantages compared to other models of building such spaces. In particular, for the specific workflows typically seen in electronic discovery context, this method offers a very practical solution.

2. Problem Description

Electronic discovery almost always involves searching for relevant and/or responsive documents. Given the importance of e-discovery search, it is imperative that the best technologies are applied for the task. Keyword based search has been the bread and butter method of searching, but its limitations have been well understood and documented in a seminal study by Blair & Moran [8]. At its most basic level, concept search technologies are designed to overcome some limitations of keyword search.

When applied to document discovery, traditional Boolean keyword search often results in sets of documents that include non-relevant items (false positives) or that exclude relevant terms (false negatives). This is primarily due to the effects of synonymy (different words with similar meanings) or polysemy (same word with multiple meanings). For polysemes, an important characteristic requirement is that they share the same etymology but their usage has evolved it into different meanings. In addition, there are also situations where words that do not share the same etymology have different meanings (e.g., river bank vs. financial bank), in which case they are classified as homonyms.

In addition to the above word forms, unstructured text content, and especially written text in emails and instant messages contain user-created code words, proper name equivalents, contextually defined substitutes, and prepositional references etc., that mask the document from being indentified using Boolean keyword search. Even simple misspellings, typos and OCR scanning errors can make it difficult to locate relevant documents.

Also common is an inherent desire of speakers to use a language that is most suited from the perspective of the speaker. The Blair Moran study illustrates this using an event which the victim's side called the event in question an "accident" or a "disaster" while the plaintiff's side called it an "event", "situation", "incident", "problem", "difficulty", etc. The combination of human emotion, language variation, and assumed context makes the challenge of retrieving these documents purely on the basis of Boolean keyword searches an inadequate approach.

Concept based searching is a very different type of search when compared to Boolean keyword search. The input to concept searching is one or more words that allow the investigator or user to express a concept. The search system is then responsible for identifying other documents that belong to the same concept. All concept searching technologies attempt to retrieve documents that belong to a concept (reduce false negatives and improve recall) while at the same time not retrieve irrelevant documents (reduce false positives and increase precision).

3. Concept Search approaches

Concept search, as applied to electronic discovery, is a search using meaning or semantics. While it is very intuitive in evoking a human reaction, expressing meaning as input to a system and applying that as a search that retrieves relevant documents is something that requires a formal model. Technologies that attempt to do this formalize both the input request and the model of storing and retrieving potentially relevant documents in a mathematical form. There are several technologies available for such treatment, with two broad overall approaches: unsupervised learning and supervised learning. We examine these briefly in the following sections.

3.1 Unsupervised learning

These systems convert input text into a semantic model, typically by employing a mathematical analysis technique over a representation called vector space model. This model captures a statistical signature of a document through its terms and their occurrences. A matrix derived from the corpus is then analyzed using a Matrix decomposition technique.

The system is unsupervised in the sense that it does not require a training set where data is pre-classified into concepts or topics. Also, such systems do not use ontology or any classification hierarchy and rely purely on the statistical patterns of terms in documents.

These systems derive their semantics through a representation of co-occurrence of terms. A primary consideration is maintaining this co-occurrence in a form that reduces impact of noise terms while capturing the essential elements of a document. For example, a document about an automobile launch may contain terms about automobiles, their marketing activity, public relations etc., but may have a few terms related to the month, location and attendees, along with frequently occurring terms such as pronouns and prepositions. Such terms do not define the concept automobile, so their impact in the definition must be reduced. To achieve such end result, unsupervised learning systems represent the matrix of document-terms and perform a mathematical transformation called dimensionality reduction. We examine these techniques in greater detail in subsequent sections.

3.2 Supervised learning

In the supervised learning model, an entirely different approach is taken. A main requirement in this model is supplying a previously established collection of documents that constitutes a training set. The training set contains several examples of documents belonging to specific concepts. The learning algorithm analyzes these documents and builds a model, which can then be applied to other documents to see if they belong to one of the several concepts that is present in the original training set. Thus, concept searching task becomes a concept learning task.

It is a machine learning task with one of the following techniques.

a) Decision Trees
b) Naïve Bayesian Classifier
c) Support Vector Machines

While supervised learning is an effective approach during document review, its usage in the context of searching has significant limitations. In many situations, a training set that covers all possible outcomes is unavailable and it is difficult to locate exemplar documents. Also, when the number of outcomes is very large and unknown, such methods are known to produce inferior results.

For further discussion, we focus on the unsupervised models, as they are more relevant for the particular use cases of concept search.

3.3 Unsupervised Classification Explored

As noted earlier, concept searching techniques are most applicable when they can reveal semantic meanings of a corpus without a supervised learning phase. To further characterize this technology, we examine various mathematical methods that are available.

3.4 Latent Semantic Indexing

Latent Semantic Indexing is one of the most well-known approaches to semantic evaluation of documents. This was first advanced in Bell Labs (1985), and later advanced by Susan Dumais and Landauer and further developed by many information retrieval researchers. The essence of the approach is to build a complete term-document matrix, which captures all the documents and the words present in each document. Typical representation is to build an N x M matrix where the N rows are the documents, and M columns are the terms in the corpus. Each cell in this matrix represents the frequency of occurrence of the term at the “column” in the document “row”.

Such a matrix is often very large – document collections in the millions and terms reaching tens of millions are not uncommon. Once such a matrix is built, mathematical technique known as Singular Value Decomposition (SVD) reduces the dimensionality of the matrix into a smaller size. This process reduces the size of the matrix and captures the essence of each document by the most important terms that co-occur in a document. In the process, the dimensionally reduced space represents the “concepts” that reflect the conceptual contexts in which the terms appear.

3.5 Principal Component Analysis

This method is very similar to latent semantic analysis in that a set of highly correlated artifacts of words and documents in which they appear, is translated into a combination of the smallest set of uncorrelated factors. These factors are the principal items of interest in defining the documents, and are determined using a singular value decomposition (SVD) technique. The mathematical treatment, application and results are similar to Latent Semantic Indexing.

A variation on this, called independent component analysis is a technique that works well with data of limited variability. However, in the context of electronic discovery documents where data varies widely, this results in poor performance.

3.6 Non-negative matrix factorization

Non-negative matrix factorization (NMF) is another technique most useful for classification and text clustering where a large collection of documents are forced into a small set of clusters. NMF constructs a document-term matrix similar to LSA and includes the word frequency of each term. This is factored into a term-feature and feature-document matrix, with the features automatically derived from the document collection. The process also constructs data clusters of related documents as part of the mathematical reduction. An example of this research is available at [2] which takes the Enron email corpus and classifies the data using NMF into 50 clusters.

3.7 Latent Dirichlet Allocation

Latent Dirichlet Allocation is a technique that combines elements of Bayesian learning and probabilistic latent semantic indexing. In this sense, it relies on a subset of documents pre-classified into a training set, and unclassified documents are classified into concepts based on a combination of models from the training set [15].

3.8 Comparison of the above technologies

Although theoretically attractive and experimentally successful, word space models are plagued with efficiency and scalability problems. This is especially true when the models are faced with real-world applications and large scale data sets. The source of these problems is the high dimensionality of the context vectors, which is a direct function of the size of the data. If we use document-based co-occurrences, the dimensionality equals the number of documents in the collection, and if we use word-based co-occurrences, the dimensionality equals the vocabulary, which tends to be even bigger than the number of documents. This means that the co-occurrence matrix will soon become computationally intractable when the vocabulary and the document collections grow.

Nearly all the technologies build a word space by building a word-document matrix with each row representing a document and column representing a word. Each cell in such a matrix represents the frequency of occurrence of the word in that document. All these technologies suffer from a memory space challenge, as these matrices grow to very large sizes. Although many cells are sparse, the initial matrix is so large that it is not possible to accommodate the computational needs of large electronic discovery collections. Any attempt to reduce this size to a manageable size is likely to inadvertently drop potentially responsive documents.

Another problem with all of these methods is that they require the entire semantic space to be constructed ahead of time, and are unable to accommodate new data that would be brought in for analysis. In most electronic discovery situations, it is routine that some part of the data is brought in as a first loading batch, and once review is started, additional batches are processed.

4. Reflective Random Indexing

Reflective random indexing (RRI) [6, 7, 11] is a new breed of algorithms that has the potential to overcome the scalability and workflow limitations of other methods. RRI builds a semantic space that incorporates a concise description of term-document co-occurrences. The basic idea of the RRI and the semantic vector space model is to achieve the same dimensionality reduction espoused by latent semantic indexing, without requiring the mathematically complex and intensive singular value decomposition and related matrix methods. RRI builds a set of semantic vectors, in one of several variations – term-term, term-document and term-locality. For this study, we built an RRI space using term-document projections, with a set of term vectors and a set of document vectors. These vectors are built using a scan of the document and term space with several data normalization steps.

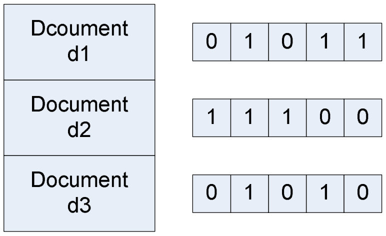
The algorithm offers many parameters for controlling the generation of semantic space to suit the needs of specific accuracy and performance targets. In the following sections, we examine the elements of this algorithm, its characteristics and various parameters that govern the outcome of the algorithm.

4.1 Semantic Space Construction

As noted earlier, the core technology is the construction of semantic space. A primary characteristic of the semantic space is a term-document matrix. Each row in this matrix represents all documents a term appears in. Each column in that matrix represents all terms a document contains. Such a representation is an initial formulation of the problem for vector-space models. Semantic relatedness is expressed in the connectedness of each matrix cell. Two documents that share the same set of terms are connected through a direct connection. It is also possible for two documents to be connected using an indirect reference.

In most cases, term-document matrix is a very sparse matrix and can grow to very large sizes for most document analysis cases. Dimensionality reduction reduces the sparse matrix into a manageable size. This achieves two purposes. First, it enables large cases to be processed in currently available computing platforms. Second, and more importantly, it captures the semantic relatedness through a mathematical model.

The RRI algorithm begins by assigning a vector of a certain dimension to each document in the corpus. These assignments are chosen essentially at random. For example, the diagram below has assigned a five-dimensional vector to each document, with specific randomly chosen numbers at each position. These numbers are not important – just selecting a unique pattern for each document is sufficient.



Dcoument d1	0	1	0	1	1
Document d2	1	1	1	0	0
Document d3	0	1	0	1	0

Figure 1: Document Vectors

From document vectors, we construct term vectors by iterating through all terms in the corpus, and for each term, we identify the documents that term appears in. In cases where the term appears multiple times in the same document, that term is given a higher weight by using its term frequency.

$$t_{i,j} = \sum_{k=0}^{i} n_k d_{k,j}$$

Each term k's frequency in the document n_k weighs in for each document vector's position. Thus, this operation projects all the documents that a term appears in, and condenses it into the dimensions allocated for that term. As is evident, this operation is a fast scan of all terms and their document positions. Using Lucene API *TermEnum* and *TermDocs*, a collection of term vectors can be derived very easily.

Once the term vectors are computed, these term vectors are projected back on to document vectors. We start afresh with a new set of document vectors, where each vector is a sum of the term vectors for all the terms that appear in that document. Once again, this operation is merely an addition of floating point numbers of each term vector, adjusting for its term frequency in that document. A single sweep of document vectors to term vector projection followed by term vectors to document vector constitutes a training cycle. Depending on needs of accuracy in the construction of semantic vectors, one may choose to run the training cycle multiple times. Upon completion of the configured number of training cycles, document and term vector spaces are persisted in a form that enables fast searching of documents during early data exploration, search, and document review.

It is evident that by constructing the semantic vector space, the output space captures the essential co-occurrence patterns embodied in the corpus. Each term vector represents a condensed version all the documents the term appears in, and each document vector captures a summary of the significant terms present in the document. Together, the collection of vectors represents the semantic nature of related terms and documents.

Once a semantic space is constructed, a search for related terms of a given query term is merely a task of locating the nearest neighbors of the term. Identifying such terms involves using the query vector to retrieve other terms in the term vector stores which are closest to it by cosine measurement. Retrieving matching documents for a query term is by identifying the closest documents to the query term's vector in document vector space, again by way of cosine similarity.

An important consideration for searching vector spaces is the performance of locating documents that are cosine-similar, without requiring a complete scan of the vector space. To facilitate this, the semantic vector space is organized in the form of clusters, with sets of the closest vectors characterized by both its centroid and the Euclidean distance of the farthest data point in the cluster. These are then used to perform a directed search eliminating the examination of a large number of clusters.

4.2 Benefits of Semantic Vector Space

From the study the semantic vector space algorithm, one can immediately notice the simplicity in realizing the semantic space. A linear scan of terms, followed by a scan of documents is sufficient to build a vector space. This simplicity in construction offers the following benefits.

a) In contrast to LSA and other dimensionality reduction techniques the semantic space construction requires much less memory and CPU resources. This is primarily because matrix operations such as singular value decomposition (SVD) are computationally intensive, and requires both the initial term-document matrix and intermediate matrices to be manipulated in memory. In contrast, semantic vectors can be built for a portion of the term space, with a portion of the index. It is also possible to scale the solution simply by employing persistence to disk at appropriate batching levels, thus scaling to unlimited term and document collections.

b) The semantic vector space building problem is more easily parallelizable and distributable across multiple systems. This allows parallel computation of the space, allowing for a distributed algorithm to work on multiple term-document spaces simultaneously. This can dramatically increase the availability of concept search capabilities to very large matters, and within time constraints that are typically associated with large electronic discovery projects..

c) Semantic space can be built incrementally, as new batches of data are received, without having to build the entire space from scratch. This is a very common scenario in electronic discovery, as an initial batch of document review needs to proceed before all batches are collected. It is also fairly common for the scope of electronic discovery to increase after early case assessment.

d) Semantic space can be tuned using parameter selection such as dimension selection, similarity function selection and selection of term-term vs. term-document projections. These capabilities allow electronic discovery project teams to weigh the costs of computational resources against the scope of documents to be retrieved by the search. If a matter requires a very narrow interpretation of relevance, the concept search algorithm can be tuned and iterated rapidly.

Like other statistical methods, semantic spaces retain their ability to work with a corpus containing documents from multiple languages, multiple data types and encoding types etc., which is a key requirement for e-discovery. This is because the system does not rely on linguistic priming or linguistic rules for its operation.

5. Performance Analysis

Resource requirements for building a semantic vector space is an important consideration. We evaluated the time and space complexity of semantic space algorithms as a function of corpus size, both from the initial construction phase and for follow-on search and retrievals.

Performance measurements for both aspects are characterized for four different corpora, as indicated below.

Corpus	**Reuters Collection**	**EDRM Enron**	**TREC Tobacco Corpus**
PST Files	-	171	-
No. of Emails	-	428072	-
No. of Attachments	21578	305508	6,270,345
No. of Term Vectors (email)	-	251110	-
No. of Document Vectors (email)	-	402607	-
No. of Term Vectors (attachments)	63210	189911	3,276,880
No. of Doc Vectors (attachments)	21578	305508	6,134,210
No. of Clusters (email)	-	3996	-
No. of Clusters (attachments)	134	2856	210,789

Table 1: Data Corpus and Semantic Vectors

As can be observed, term vectors and document vectors vary based on the characteristics of the data. While the number of document vectors closely tracks the number of documents, the number of term vectors grows more slowly. This is the case even for OCR-error prone ESI collections, where the term vector growth moderated as new documents were added to the corpus.

5.1 Performance of semantic space building phase

Space complexity of the semantic space model is linear with respect to the input size. Also, our implementation partitions the problem across certain term boundaries and persists the term and document vectors for increased scalability. The algorithm requires memory space for tracking one million term and document vectors, which is about 2GB, for a semantic vector dimension of 200.

Time for semantic space construction is linear on the number of terms and documents. For very large corpus, the space construction requires periodic persistence of partially constructed term and document vectors and their clusters. A typical configuration persist term vectors for each million terms, and documents at each million documents. As an example, the TREC tobacco corpus would require 4 term sub-space constructions, with six document partitions, yielding 24 data persistence invocations. If we consider the number of training cycles, each training cycle repeats the same processes. As an example, the TREC tobacco corpus with two training cycles involves 48 persistence invocations. For a corpus of this size, persistence adds about 30 seconds for each invocation.

Performance Item	Vector Construction (minutes)	Cluster Construction (minutes)
Reuters-21578 dataset	1	1
EDRM Enron dataset	40	15
TREC Tobacco Corpus	490	380

Table 2: Time for space construction, two training cycles (default)

These measurements were taken on commodity Dell PowerEdge R710 system, with two Quad Xeon 5500 processors at 2.1GHz CPU and 32GB amount of memory.

5.2 Performance of exploration and search

Retrieval time for a concept search and time for building semantic space exploration are also characterized for various corpus sizes and complexity of queries. To facilitate a fast access to term and document vectors, our implementation has employed a purpose-built object store. The object store offers the following.

a) Predictable and consistent access to a term or document semantic vector. Given a term or document, the object store provides random access and retrieval to its semantic vector within 10 to 30 milliseconds.

b) Predictable and consistent access to all nearest neighbors (using cosine similarity and Euclidean distance measures) of a term and document vector. The object store has built-in hierarchical k-means based clustering. The search algorithm implements a cluster exploration technique that algorithmically chooses the smallest number of clusters to examine for distance comparisons. A cluster of 1000 entries is typically examined in 100 milliseconds or less.

Given the above object store and retrieval paths, retrieval times for searches range from 2 seconds to 10 seconds, depending on large part, on the number of nearest neighbors of a term, the number of document vectors to retrieve and on the size of the corpus.

The following table illustrates observed performance for the Enron corpus, using the cluster-directed search described above.

Term vector search	Average	Stdev
Clusters Examined	417.84	274.72
Clusters Skipped	1001.25	478.98
Terms Compared	24830.38	16079.72
Terms Matched	21510.29	15930.2
Total Cluster Read Time (ms)	129.39	88.23
Total Cluster Read Count	417.84	274.72
Average Cluster Read Time (ms)	0.29	0.18
Total Search Time (ms)	274.56	187.27

Table 3: Search Performance Measurements

As is apparent from the above time measurements as well as number of clusters examined and skipped, identifying related terms can be offered to users with predictability and consistency, thereby making it possible for its usage as an interactive, exploratory tool during early data analysis, culling, analysis and review phases of electronic discovery.

6. Search Effectiveness

An important analysis is to evaluate the effectiveness of retrieval of related terms from the perspective of the search meeting the information retrieval needs of the e-discovery investigator. We begin by analyzing qualitative feel for search results by examining the related terms and by identifying the relevance of these terms. We then analyze search effectiveness using the standard measures, Precision and Recall. We also examine search effectiveness using Discounted Cumulative Gain (DCG).

6.1 Qualitative Assessment

To obtain a qualitative assessment, we consider the related terms retrieved and examine its nearness measurement, and validate the closest top terms. The nearness measure we use for this analysis is a cosine measure of the initial query vector when compared with the reported result. It is a well-understood measure of judgment of quality in that a cosine measure reflects the alignment of the two vectors, and closeness to the highest value of cosine, which is 1.0, means perfect alignment.

Table 4 shows alignment measures for two concept query terms for the EDRM Enron Dataset [12].

It is quite clear that several of the related terms are in fact logically related. In cases where the relationship is suspect, it is indeed the case that co-occurrence is properly represented. E.g., the term *offshore* and *mainly* appear in enough documents together to make it to the top 20 related terms. Similarly, we have *offshore* and *foreign* co-occur to define the concept of *offshore* on the basis of the identified related terms.

Query: drilling		**Query: offshore**	
Related Term	**Similarity**	**Related Term**	**Similarity**
refuge	0.15213	interests	0.13212
Arctic	0.12295	foreign	0.13207
wildlife	0.12229	securing	0.12597
exploration	0.11902	viable	0.12422
Rigs	0.11172	involves	0.12345
Rig	0.11079	associated	0.12320
supplies	0.11032	mainly	0.12266
Oil	0.11017	principle	0.12248
refineries	0.10943	based	0.12241
Environmentalists	0.10933	achieved	0.12220

Table 4: Illustration of two query terms and their term neighbors

We can further establish the validity of our qualitative assessment using individual document pairs and their document co-occurrence patterns. As an example, Table 5 shows cosine similarity, the number of documents the two terms appear in and the common set of documents both terms appear in, again in the EDRM Enron Dataset.

Term1	**Term2**	**Cosine**	**Docs1**	**Docs2**	**CDocs**
offshore	drilling	0.2825	1685	1348	572
governor	Davis	0.3669	2023	2877	943
brownout	power	0.0431	13	30686	13
brownout	ziemianek	0.5971	13	2	1

Table 5: Cosine similarity comparison for select terms from EDRM Enron corpus

An observation from the above data is that when the two terms compared appear in large number of documents with large overlap, the similarity is greater. In contrast, if one term is dominant in its presence in a large number of documents, and the other term is not, the presence of the two terms in all the common documents (*brownout* and *power*), the similarity is lower. Also noteworthy is if two terms are common in every document and the documents each appears in are small number (*brownout* and *ziemianek*) the similarity measure is significantly higher.

6.2 Precision and Recall Measures

Precision and recall are two widely used metrics for evaluating the correctness of a search algorithm [8]. Precision refers to the ratio of relevant results compared to the full retrieved set, and represents the number of false positives in the result. Recall on the other hand, measures the ratio of relevant results compared to the number of relevant results actually present in the collection, i.e. the number of false negatives. Usually, recall is a harder measure to determine since it would require reviewing the entire collection for identifying all the relevant items, and sample-based estimation is a substitute.

For our purposes, two critical information retrieval needs should be evaluated.

a) The ability of the system to satisfy information retrieval needs for the related concept terms.

b) The ability of the system to provide the same for documents in a concept.

We evaluated both for several specific searches using the EDRM Enron dataset, and we present our results below.

6.3 Precision and Recall for Related Concept Terms

Note that Precision and Recall are defined for related concept terms using a combination of automated procedures and manual assessment. As an example, we supplied a list of queries and their related concept terms and asked human reviewers to rate each related term result as either strongly correlated or related to the initial query, or if it is not related. This gives us an indication of precision for our results, for a given cutoff point. Evaluating recall is harder, but we utilized a combination of sampling methodology and a deeper probe into related term result. As an example of this, we evaluated precision for a cutoff at 20 terms and recall by examining 200 terms and constructing relevance graphs.

6.4 Impact of dimensions

Given that the semantic vector space performs a dimensionality reduction, we were interested in understanding the impact of dimension choice for our semantic vectors. For the default implementation, we have a vector dimension of 200, which means that each term and document has a vector of 200 floating point numbers.

To study this, we performed a study of precision and recall for the EDRM Enron dataset and tracked the precision-recall graph for four choices of dimensions. The results are indicated in Figure 2 below.

As can be observed, we did not gain significant improvement on precision and recall characteristics with a higher choice of dimension. However, for a large corpus, we expect that precision-recall graph would indicate a significantly steeper fall-off.

We also evaluated search performance relative to dimensions. As expected, there is a direct correlation between the two, which can be explained by the additional disk seeks to retrieve both cluster objects as well as semantic vectors for comparison to the query vector. This is illustrated in Figure 3 below.

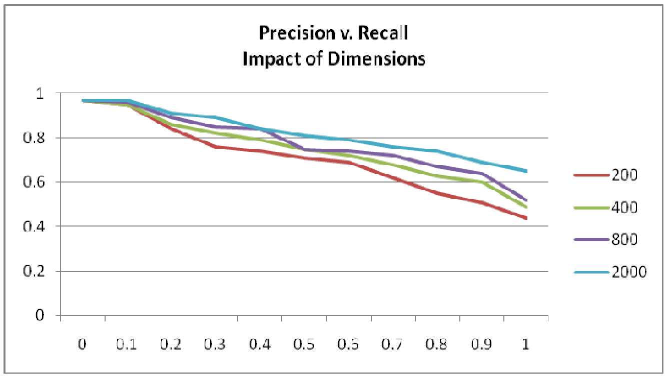


Figure 2: Precision and Recall graphs for the EDRM Enron Dataset

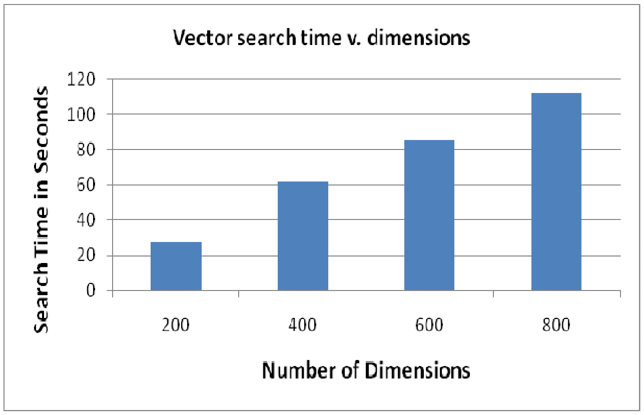


Figure 3: Characterizing Search time and dimensions for 20 random searches

A significant observation is that overall resource consumption increases substantially with increase in dimensions. Additionally, vector-based retrieval also times increase significantly. We need to consider these resource needs in the context of improvements in search recall and precision quality measures.

6.5 Discounted Cumulative Gain

In addition to Precision and Recall, we evaluated the Discounted Cumulative Gain (DCG), which is a measure of how effective the concept search related terms are [14]. It measures the relative usefulness of a concept search related term, based on its position in the result list. Given that Concept Search query produces a set of related terms and that a typical user would focus more on the higher-ranked entries, the relative position of related terms is a very significant metric.

Figure 4 illustrates the DCG measured for the EDRM Enron Dataset for a set of 20 representative searches, for four dimension choices indicated.

We evaluated the retrieval quality improvements in the context of increases in resource needs and conclude that acceptable quality is achievable even with a dimension of 200.

6.6 Impact of Training Cycles

We studied the impact of training cycles on our results. A training cycle captures the co-occurrence vectors computed in one cycle to feed into the next cycle as input vectors. As noted earlier, the document vectors for each training cycle start with randomly assigned signatures, and each successive training cycle utilizes the learned term semantic vectors and feeds it into the final document vectors for that phase. This new set of document vectors forms the input (instead of the random signatures) for the next iteration of the training cycle.

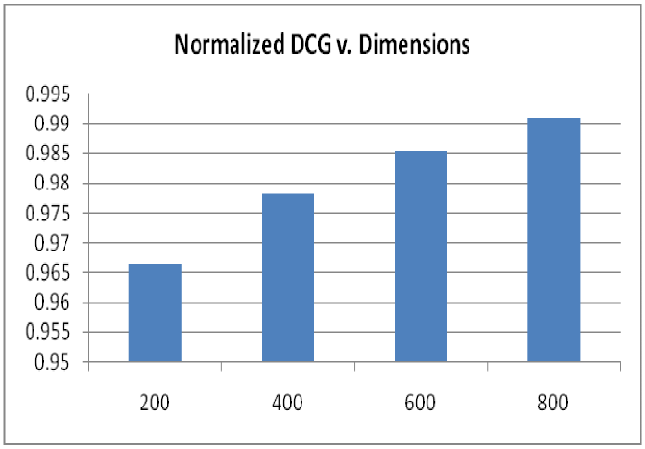


Figure 4: Normalized DCG vs. dimensions of semantic vector space

In our model, we note that term has a direct reference to another discovered term when they both appear in the same document. If they do not appear in the same document but are connected by one or more other common terms between the two documents, we categorize that as an indirect reference.

Adding training cycles has the effect of discovering new indirect references from one term to another term, while also boosting the impact of common co-occurrence. As an example, Table 6 illustrates training cycle 1 and training cycle 4 results for the term drilling. Notice that new terms appear whose co-occurrence is reinforced by several indirect references.

Another view into the relative changes to term-term similarity across training cycles is shown below. Table 7 illustrates the progression of term similarity as we increase the number of training cycles. Based on our observations, the term-term similarity settles into a reasonable range in just two cycles, and additional cycles do not offer any significant benefit.

Also noteworthy is that although the initial assignments are random, the discovered terms settle into a predictable collection of co-occurrence relationship, reinforcing the notion that initial random assignment of document vectors get subsumed by real corpus-based co-occurrence effects.

Query: drilling			
Training Cycle 1		**Training Cycle 4**	
Related Term	**Similarity**	**Related Term**	**Similarity**
Wells	0.164588	rigs	0.25300
Rigs	0.151399	wells	0.23867
viking	0.133421	offshore	0.22940
Rig	0.130347	rig	0.21610
buckeye	0.128801	exploration	0.21397
Drill	0.124669	geo	0.20181
exploration	0.123967	mcn	0.19312
richner	0.122284	producing	0.18966
producing	0.121846	ctg	0.18904
alpine	0.116825	gulf	0.17324

Table 6: Training Cycle Comparisons

Term1	**Term2**	**TC-1**	**TC-2**	**TC-3**	**TC-4**
offshore	drilling	0.2825	0.9453	0.9931	0.9981
governor	davis	0.3669	0.9395	0.9758	0.9905
brownout	power	0.0431	0.7255	0.9123	0.9648
brownout	ziemianek	0.5971	0.9715	0.9985	0.9995

Table 7: Term Similarity of training cycles (TC) for four cycles

7. CONCLUSIONS

Our empirical study of Reflective Random Indexing indicates that it is suitable for constructing a semantic space for analyzing large text corpora. Such a semantic space has the potential to augment traditional keyword-based searching with related terms as part of query expansion. Co-occurrence patterns of terms within documents are captured in a way that facilitates very easy query construction and usage. We also observed the presence of several direct and indirect co-occurrence associations, which is useful in a concept based retrieval of text documents in the context of electronic discovery. We studied the impact of dimensions and training cycles, and our validations indicate that a choice of 200 dimensions and two training cycles produced acceptable results.

8. REFERENCES

[1] Donna Harman, Towards Interactive Query Expansion, Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, Maryland

[2] Myaeng, S. H., & Li, M. (1992). Building Term Clusters by Acquiring Lexical Semantics from a Corpus. In Y. Yesha (Ed.), CIKM-92, (pp. 130-137). Baltimore, MD: ISMM.

[3] Susan Gauch and Meng Kam Chong, Automatic Word Similarity Detection for TREC 4 Query Expansion, Electrical Engineering and Computer Science, University of Kansas

[4] Yonggang Qiu, H.P.Frei, Concept-Based Query Expansion, Swiss Federal Institute of Technology, Zurich, Switzerland

[5] Ian Ruthven, Re-examining the Potential Effectiveness of Interactive Query Expansion, Department of Computer and Information Sciences, University of Strathclyde, Glasgow

[6] An Introduction to Random Indexing, Magnus Sahlgren, SICS, Swedish Institute of Computer Science.

[7] Trevor Cohen (Center for Cognitive Informatics and Decision Making, School of Health Information Sciences, University of Texas), Roger Schvaneveldt (Applied Psychology Unit, Arizona State University), Dominic Widdows (Google Inc., USA)

[8] Blair, D.C. & Moran M.E. (1985). An evaluation of retrieval effectiveness for a full-text document-retrieval system. Communications of the ACM, 28, 298-299

[9] Berry, Michael W.; Browne (October 2005). "Email Surveillance Using Non-negative Matrix Factorization". Computational & Mathematical Organization Theory 11 (3): 249–264. doi:10.1007/s10588-005-5380-5.

[10] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman (1990). "Indexing by Latent Semantic Analysis" (PDF). Journal of the American Society for Information Science 41 (6): 391–407. doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9. http://lsi.research.telcordia.com/lsi/papers/JASIS90.pdf. Original article where the model was first exposed.

[11] Widdows D, Ferraro K. Semantic vectors: a scalable open source package and online technology management application. In: 6th International conference on language resources and evaluation (LREC); 2008.

[12] EDRM Enron Dataset, http://edrm.net/resources/data-sets/enron-data-set-files

[13] Precision and Recall explained, http://en.wikipedia.org/wiki/Precision_and_recall

[14] Discounted Cumulative Gain, http://en.wikipedia.org/wiki/Discounted_cumulative_gain

[15] Latent Dirichlet Allocation, http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation