2008-11-21

An Analysis of Document Retrieval and Clustering Using an Effective Semantic Distance Measure

Nathan Scott Davis
Brigham Young University - Provo

AN ANALYSIS OF DOCUMENT RETRIEVAL AND CLUSTERING USING AN EFFECTIVE SEMANTIC DISTANCE MEASURE

by

Nathan S. Davis

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

December 2008

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Nathan S. Davis

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Christophe G. Giraud-Carrier, Chair
Date	Eric K. Ringger
Date	Sean C. Warnick

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Nathan S. Davis in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date	Christophe G. Giraud-Carrier Chair, Graduate Committee
Accepted for the Department	Kent E. Seamons Graduate Coordinator
Accepted for the College	Thomas W. Sederberg Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

AN ANALYSIS OF DOCUMENT RETRIEVAL AND CLUSTERING
USING AN EFFECTIVE SEMANTIC DISTANCE MEASURE

Nathan S. Davis

Department of Computer Science

Master of Science

As large amounts of digital information become more and more accessible, the ability to effectively find relevant information is increasingly important. Search engines have historically performed well at finding relevant information by relying primarily on lexical and word based measures. Similarly, standard approaches to organizing and categorizing large amounts of textual information have previously relied on lexical and word based measures to perform grouping or classification tasks. Quite often, however, these processes take place without respect to semantics, or word meanings. This is perhaps due to the fact that the idea of meaningful similarity is naturally qualitative, and thus difficult to incorporate into quantitative processes.

In this thesis we formally present a method for computing quantitative document-level semantic distance, which is designed to model the degree to which humans would associate two documents with respect to conceptual similarity. We show how this metric can be applied to document retrieval and clustering problems.

We conclude that while our metric is not well suited for text indexing, the use of our semantic distance metric can improve document retrieval through result set re-ranking and query expansion. We also conclude that our semantic distance metric can be used to improve document clustering in distance-based clustering algorithms.

ACKNOWLEDGMENTS

Many thanks to Dr. Christophe Giraud-Carrier, Dr. Eric Ringger, Dr. Sean Warnick, Mr. Del Jensen, Mr. Aaron Davis, and Mrs. Marilee Davis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In the last twenty years, digital information has become widely accessible on an unprecedented scale. As a result, several related, important problems have become the focus of many computer scientists and researchers. Some of these problems include how to best retrieve information that is relevant to a user's interest, as well as how to automatically organize large amounts of digital information. Many different approaches have been taken to address these problems, and some with a great deal of success. However, there remains significant room for improvement.

The information that has become easily accessible through digital media can be found in a variety of formats. Web pages, images, videos, and animations are a few of these commonly accessed information formats of which any Internet user is well aware. A majority of this information, however, is represented as text. Textual information is most commonly accessible in discrete units that we will hereafter refer to as *documents*. A web page can be considered a document, as well as a news article, or a blog post. We focus on text documents in this thesis.

Digital text can be processed automatically by computers. For example, a computer can very quickly and trivially determine if a particular word occurs in a given news article. The field of Information Retrieval (IR) has been dedicated to devising ways to quickly retrieve documents that are relevant to user queries.

Collections of information have become so large that it is impractical to attempt to find the most relevant information to a particular interest by simply browsing the collection.

A popular method of supplying user input to an information retrieval system is through keyword queries (see Figure 1.1). Web search engines, such as Google, perform this task very well. Despite the success of existing information retrieval systems, the relationships between queries and documents have traditionally been limited to lexical analysis. This is evident in the very popular Vector Space Model (VSM) [53]. Essentially the vector space model represents documents as vectors, whose element values represent some lexical measure related to words (i.e., terms) in a document, such as term frequency and inverse document frequency (TF-IDF), which is derived by observing frequencies of terms in and across documents [52]. A VSM document retrieval system can be issued a keyword query, where documents are returned whose vector representation has the smallest distance from the vector representation of the query. The vector space model does work well in information

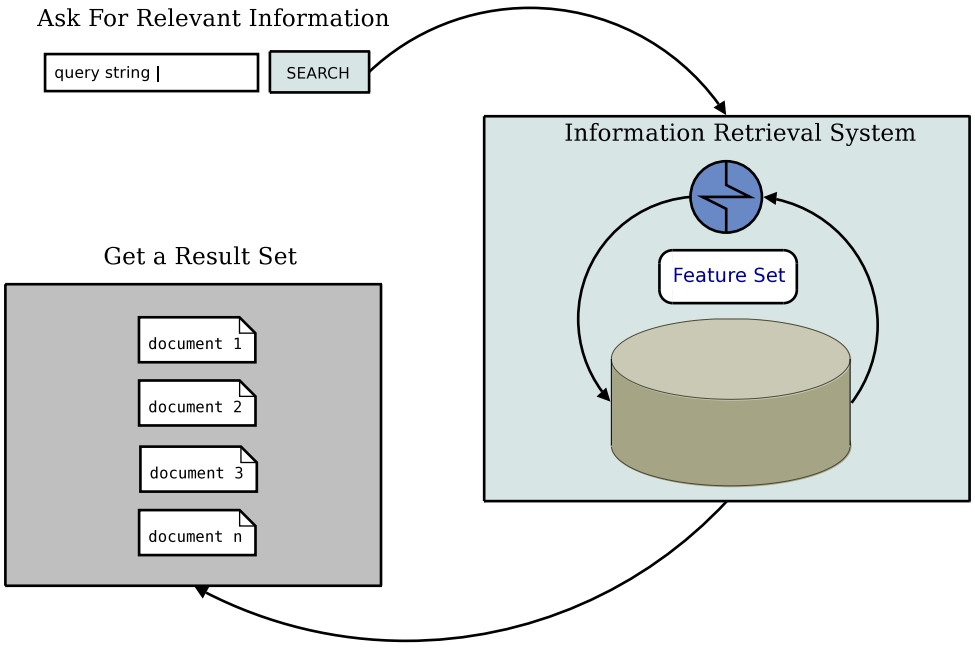

Figure 1.1: The information retrieval process.

retrieval, but is one example of several approaches that do not make use explicitly of higher-order semantics.

Document clustering is another important problem, where documents are automatically grouped according to similarity (see Figure 1.2). Natural Language Processing (NLP) systems have most recently been shown to be most effective at document clustering, however these also typically rely on lexical analysis to make assessments about the similarity of documents and do not make use of word semantics when determining if two documents are similar and should be clustered together.

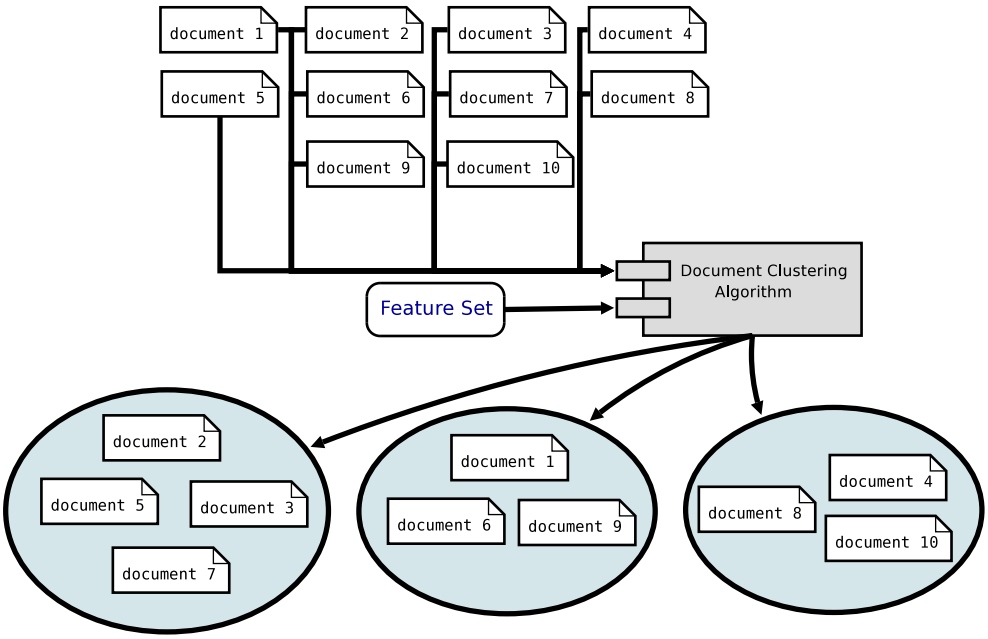

Figure 1.2: Document clustering.

We propose that by making use of word semantics, better results can be obtained in both information retrieval and document clustering.

1.2 Motivation

Common approaches to information retrieval and document clustering use some notion of relatedness with respect to text data. This idea of relatedness is often expressed as a quantitative measure of either distance or similarity (where distance is just the

inverse of similarity, and vice versa). In information retrieval one measures the distance between documents in a collection, and the user keyword query, and returns the documents that are closest to the query. In document clustering, one determines the distances between all the documents in a collection and groups those that are closest to each other. These distance metrics are document-level distance metrics because they provide a quantitative measure of distance between documents. We propose that this comparison between documents is the most natural level of comparison for these problems.

Although most distances are lexical in nature, some methods do make use of semantics. These approaches vary in their implementations and applications, but generally fall into one of two categories. The first contains corpus-based approaches, which derive similarity measures from lexical and statistical information extracted from text corpora (e.g., frequency counts), and infer *implicit* semantic relationships. For example, methods have been devised that suggest that if two words frequently occur together, then there must exist some semantic relationship between them. The second of these two categories contains knowledge-based approaches, which derive similarity measures from lexicons and ontologies that define *explicit* semantic relationships. For example, G.A. Miller of Princeton University, has developed the very comprehensive WordNet [37]. This large lexicon explicitly defines hierarchical semantic relationships between words, including hypernymy/hyponymy (is-a relationships), and meronymy (part-whole relationships).

1.3 Overview of Work

We devote our focus to improving document retrieval and clustering by using a document-level semantic distance metric. Our document level semantic distance metric is designed to model the degree to which humans would associate two documents with respect to conceptual similarity. We define additional foundational concepts

relating to semantics and examine prevalent, existing approaches to computing word-level semantic distance. We then present novel techniques that build on knowledge-based word-level semantic distance measures to compute a quantitative document-level semantic distance metric. The remainder of the thesis is comprised of a description and analysis of several different experiments in which we use our distance metric. Finally, we provide an overview of our contributions as well as a summary of the effectiveness of our approach.

Chapter 2

Related Work

2.1 Computing Semantic Distance

There has been significant work in terms of defining semantic similarity in computational linguistics. Latent Semantic Analysis (LSA) has been the most popular of these approaches that is corpus-based [29, 12]. LSA relies on co-occurrence counts of words within documents, and applies a singular value decomposition (SVD) feature dimensionality reduction approach, from which can be derived a semantic, word-level relatedness measure. The incorporation of LSA into tasks such as document clustering has also been attempted [34]. A number of corpus-based semantic measures are surveyed in [32]. LSA related approaches, such as Probabilistic Latent Semantic Analysis (PLSA), Latent Semantic Indexing (LSI), Probabilistic Latent Semantic Indexing (PLSI), and Latent Dirichlet Allocation (LDA) use lexical and probabilistic approaches to associating terms and documents [12, 18, 3].

Most knowledge-based approaches rely on WordNet [37]. Rada et al., for example, define a functional approach based on path-length in the WordNet graph, where paths are series of connected edges in the graph of semantic relationships [45]. The function uses the sum of heuristically-computed edge weights along the shortest path connecting the synsets (groups of words with similar meanings) of the words being compared. Other approaches based on adaptations of path-length are described in [7]. Resnik presents a functional approach that uses the notion of information

content [46]. This information content is defined as the probability of occurrence in a large text corpus of the first class in the noun hierarchy that takes in the classes of the words being compared. These operations work on the actual knowledge structure that describes the lexicon, to derive a semantic distance measure.

We chose to use the word-level distance metric defined by Jensen et al. as a foundation for our document-level distance metric [22]. It differs from other existing knowledge-based approaches in that it is computed after mapping the lexicon into a topological metric space. We recognize that Prince and Lafourcade have established work that is similar, yet still distinctly different [42]. In their work, they produce a conceptual vector model (CVM). In this CVM, they describe a "cooperation process between semantic networks and conceptual vectors". Using a hierarchical model of the lexicon they take a set of concepts that span the lexicon. They then define a vector for a general concept that acts as the "activation" of the spanning concepts. This allows for manipulation of concepts with the use of standard vector operations. This process is comparable to the utilization of maximal basis chains in the work by Jensen et al. as they relate to computing a linear transformation [22]. However, our approach still differs, and extends beyond their method by fully utilizing the dual space via the construction of an explicit mapping. This mapping allows us to utilize the well defined algebraic operations that exist in the metric space, and avoid basis dependency.

2.2 Applying a Semantic Distance Metric

2.2.1 Information Retrieval

We apply our semantic distance metric to typical components of an IR system, including term indexing, result set re-ranking, and query expansion.

Term Indexing

One of the most popular methods for retrieving relevant documents from a text collection is to index the text in the collection where documents containing (or related to) terms can quickly be identified. Term indexes are created in a variety of ways, using different approaches to associate terms with documents. A very popular lexical approach to creating term indexes is via the Vector Space Model (VSM) [53]. In the VSM approach, TF-IDF scores are computed for terms in all the documents during the indexing process. Query terms are then associated with documents according to the cosine distance of the vector representations of documents where the vector components are the term TF-IDF scores. Methods for improving information retrieval with higher-order information have also been defined. Work by P. Rosso et al. demonstrates using WordNet senses for improving index quality [41]. Similarly, M. Baziz et al. make use of WordNet senses by expanding single sense terms and extracting terms that occur in synonym groups [2]. Our work differs in that we use an explicit semantic distance to determine inclusion of terms in the index.

Other methods, such as work by H.M.T. Saarikoski, make use of corpus-based methods as well as the use of a knowledge-based domain ontology to produce an index consisting of concepts rather than keywords [51]. Other recent work in information retrieval has also focused on designing metrics that combine knowledge-based approaches and corpus-based approaches (e.g., see [36, 17, 48, 26]).

Search Result Set Re-ranking

Many existing information retrieval systems present results in a ranked fashion. Some approaches do result set ranking by using external data points like query log entries or usage statistics from other users in a collaborative filtering approach to determine the degree of relevancy between documents and queries [67, 47]. T. Bogers and A. van den Bosch examine the authorship of documents to re-rank search results [6].

Other approaches rely on lexical, statistical, and clustering methods for determining ranking [62, 8]. A. Natsev et al. present the re-ranking approach most similar to our re-ranking method, in that they also re-rank search results by observing semantic relationships [40]. These semantic relationships are measured as lexical and statistical inferred semantics, however, which differs from our approach.

Query Expansion

User queries can be expanded in a number of ways, the most common of which is to augment the original user query with additional terms that help the IR system to retrieve more relevant information. In many cases this has been shown to improve precision and recall (the two most common information retrieval quality metrics) [61, 11, 38, 44]. Several methods for determining which keywords should be used to augment user queries have also been defined [38, 44, 11]. We propose to augment user keyword queries with new terms and demonstrate how to select the terms by using our semantic distance metric. Our query expansion work is most similar to query expansion done by A. Natsev et al., which expands queries according to a corpus-based semantic measure [40].

We are also motivated to produce a process that can easily be added on to existing IR systems. Consequently, we have chosen a post-processing procedure that extends user keyword queries.

2.2.2 Document Clustering

Document clustering, the task of automatically discovering groups among textual entities, has proven useful in such applications as information summarization, text categorization, document collection browsing, news filtering and topic detection (e.g., [30, 33, 65, 66]). Although several clustering approaches have been proposed, including probabilistic techniques (e.g., [14, 16]) and neural network techniques (e.g.,

[28, 50]), we focus here exclusively on distance-based techniques, as these have become very popular in document clustering.

A number of researchers have attempted to go beyond the simple vector space model and add some form of semantic information to improve document clustering. In most approaches, the semantic information is leveraged as a kind of pre-processing step before clustering takes place with a standard algorithm and standard distance metric. For example, one may use some form of word sense disambiguation and use the disambiguated terms to build document vectors (e.g., [9, 60]). The pre-processing step may also involve a mapping from the original high-dimensional word vector space to a much lower-dimensional feature vector space, which not only captures semantic but also significantly decreases the computational demands on the clustering algorithm. For example, one can use frequent terms discovery [5], latent semantic analysis (LSA) [55], an ontology [19], or a hybrid of these [57], to effect a semantic-rich dimensionality reduction and cluster within the reduced space. Although our approach also reduces dimensionality prior to clustering, it differs in that it uses an explicit semantic distance function in two common distance-based document clustering algorithms.

Chapter 3

Foundational Material

3.1 Semantic Relationships between Words

Words can be related in several different ways. In controlled vocabularies, these relationships can be categorized by three classes: equivalency, associativity, and hierarchy. The primary relation in the equivalency class is that of **synonymy**. Specifically, synonymy describes the relationship between two words that have the same conceptual meaning, and can be used interchangeably without loosing any information. An example of synonymy is the relationship between "salary" and "wage". Associative semantic relations include cause and effect, such as "accident" to "injury", process and agent, such as "speed measurement" to "speedometer", and action and product, such as "writing" and "publication."

In the class of hierarchy, **hypernymy** describes the semantic relation of being superordinate or belonging to a higher rank or class, such as with "bird" to "parrot". In a reversed sense, **hyponymy** describes the semantic relation of being subordinate or belonging to a lower rank or class, such as "parrot" to "bird". Hypernymy/Hyponomy is commonly referred to as an "is-a" relationship. Another relationship that can be described hierarchically is that of **meronymy**. Meronymy describes part-whole relations, such as "brain stem" to "brain."

3.2 Structuring Semantic Relationships

3.2.1 Ontologies

The specific structure that we use to represent hierarchical semantic relationships is an ontology. An ontology looks much like a tree structure, and expresses a taxonomy. While we will provide a method for quantitatively representing any type of hierarchical semantic relationships, our primary focus is devoted to relationships of hypernymy. Consider, for example, the words "calculator" and "computer" in the English language. Figure 3.1 shows a practical hypernymy tree for "calculator", where each concept is a subclass of the concept that precedes it in the hierarchy.

```
=> entity
  => physical entity
    => object, physical object
      => whole, unit
        => artifact, artefact
          => instrumentality, instrumentation
            => device
              => machine
                => calculator
```

Figure 3.1: Hypernymy tree for "calculator."

Figure 3.2 expresses a practical hyponymy tree for "calculator", where the subclasses of "calculator" are expressed.

Figure 3.3 shows "calculator" and "computer" (a closely related term) together in an English taxonomy of hypernyms.

3.2.2 WordNet

For our experiments, we use version 3.0 of WordNet as the hypernymy ontology. WordNet version 3.0 contains 155,287 unique English words. Other such structures could be used with the methods we develop, as long as they describe hierarchical

```
=> calculator
  => abacus
  => adder
  => adding machine, totalizer, totaliser
  => counter, tabulator
    => pulse counter
      => scaler
  => hand calculator, pocket calculator
  => Napier's bones, Napier's rods
  => quipu
  => subtracter
```

Figure 3.2: Hyponymy tree for "calculator."

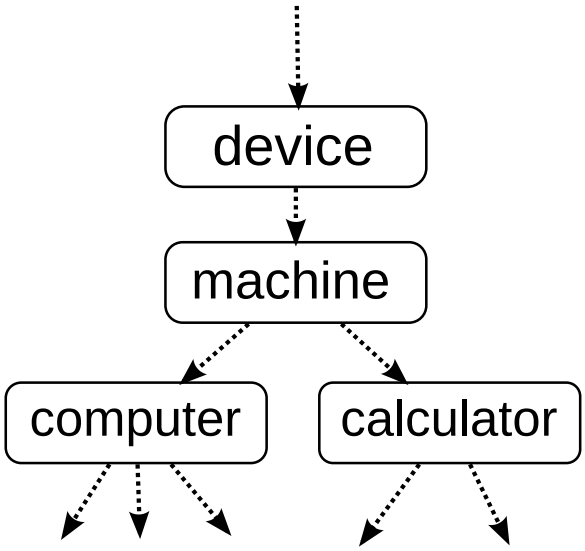

Figure 3.3: Identifying explicit hierarchical semantic relationships.

semantic relations, however we focus our efforts and experiments on using WordNet because of its rich and extensive semantic relationship hierarchy.

3.3 Topological Mapping of the Lexicon

3.3.1 Mathematical Foundation

The contributions of this thesis depend on a method for computing word-to-word semantic distances. Our work is based on the distance proposed in [22]. We give a short summary here of its design. The lexicon is mapped into a complete metric space

where there exists a natural inner product, which is then used as the distance metric. This topological mapping allows for a distance metric that quantitatively represents the qualitative relationships specified by the lexicon.

3.3.2 Basis Chains

Because it has the properties of reflexivity, anti-symmetry, and transitivity, the hyponymy relation is a partial order. Thus the corresponding graph is a lattice where it is possible to match each noun in the graph unambiguously with the contiguous (maximal) chains that connect it to the hyponymy graph's root. Maximal chains are comprised of the longest paths from nouns to the graph's root. Such contiguous chains can then be interpreted as functionals on the graph, that map points of the graph into the real numbers.

While all chains are available in principle, in practice this can result in such a large number of chains that using them to derive a linear transformation can quickly become computationally infeasible. Therefore, the feasible alternative is to choose a relatively small set of chains that can be used as a basis, which spans the complete set of chains.

Choosing different numbers of chains, and different words to create the spanning set of maximal chains, seemed to have relatively little impact on the results of our method. Because the variance in effectiveness was small, determined by the small variance in performance metrics used throughout our experiments, we chose not to more fully explore basis chains, and ways to choose them. Several ways yield very similar results. Some of the ones we tried included: using a subset of Longman's defining vocabulary of the English language [1], using the most frequently occurring words in a text corpus on which we would perform experiments, and choosing random words. For the experiments in this thesis, we report results generated by using basis

chains of words in a subset of the Longman's defining vocabulary, as those seemed to provide the best span, and yielded slightly better results.

3.3.3 Dual Spaces

The mappings of these chains are continuous under the order topology, so that "close" concepts in the graph are mapped to "close" real numbers, as one would expect. We then exploit the duality that exists between the space of conceptual elements (i.e., the WordNet nouns) and the space of chain-functions, as depicted in Figure 3.4. Finally, an inner product is adopted on the function space that is consistent with the original order topology, and use the inner product on the conjugate representations of the concept-elements to define a metric as the direction cosine distance in the dual space.

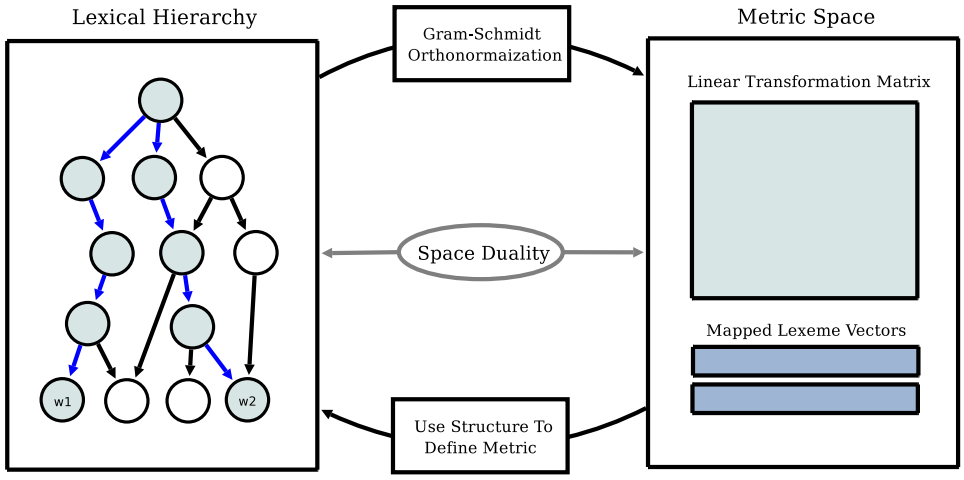

Figure 3.4: Dual spaces for computing effective semantic distance given a lexicon.

3.3.4 Building the Metric Space

Let c be some maximal chain and let q be the length of c. We define the function $m_c : c \to \mathbb{R}$ by

$$m_c(c^i) = \frac{i-1}{q-1}$$

where c^i is the ith element of c, ordered from root to leaf. Next, define the function $f_c : \mathcal{N} \to [0, 1]$ by:

$$f_c(n) = m_c(c^{i_n})$$

where c^{i_n} is the point of intersection of n closest to the leaf (lowest), via some chain containing n, with c.

We illustrate the method with the simple directed acyclic graph G of Figure 3.5. For G, the set of all maximal chains comprises our set of basis chains, while for larger graphs it may be more practical to choose a spanning set of basis chains.

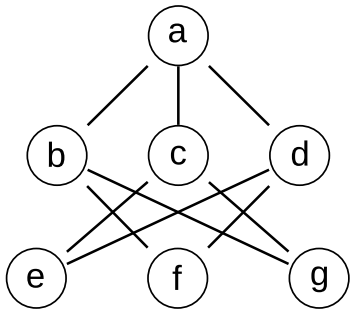

Figure 3.5: A simple directed graph.

For G, these chains are $c_1 = \{e, c, a\}$, $c_2 = \{e, d, a\}$, $c_3 = \{f, b, a\}$, $c_4 = \{f, d, a\}$, $c_5 = \{g, b, a\}$, and $c_6 = \{g, c, a\}$. The value of $f_{c_4}(e)$ is .5, since the earliest point of intersection of any chain containing e with c_4 is halfway down c_4. The complete action of the functions $\{f_{c_k}\}$ on G is shown in Figure 3.6

	f_{c_1}	f_{c_2}	f_{c_3}	f_{c_4}	f_{c_5}	f_{c_6}
a	0	0	0	0	0	0
b	0	0	.5	0	.5	0
c	.5	0	0	0	0	.5
d	0	.5	0	.5	0	0
e	1	1	0	.5	0	.5
f	0	.5	1	1	.5	0
g	.5	0	.5	0	1	1

Figure 3.6: Action of $\{f_{c_k}\}$ on G.

The set of functions $\{f_{c_k}\}$ naturally spans a vector space consisting of all linear combinations of the f_{c_k}'s. We now find an orthonormal set of functions that provides a basis for that vector space, and produce a linear transformation from $\mathcal{N}$ to the corresponding conjugates in the function space with respect to the basis (hence, an inner-product preserving representation of the original elements of G).

In the particular case of G, the Gram-Schmidt algorithm is employed to compute an orthonormal basis $\{w_i\}$ from $\{f_{c_k}\}$. The Gram-Schmidt process takes a finite, linearly independent set of vectors (i.e., a basis) and produces a new set of orthonormal vectors that span the same space. Note that f_6 is in the span of the other functions, so that $\{f_1, f_2, f_3, f_4, f_5\}$ is the linearly independent set of vectors to start from. The Gram-Schmidt algorithm is as follows:

$$u_1 = f_1$$
$$\Rightarrow w_1 = u_1/||u_1||$$
$$u_2 = f_2 - < f_2, w_1 > w_1$$
$$\Rightarrow w_2 = u_2/||u_2||$$
$$u_3 = f_3 - < f_3, w_2 > w_2 - < f_3, w_1 > w_1$$
$$\Rightarrow w_3 = u_3/||u_3||$$
$$u_4 = f_4 - < f_4, w_3 > w_3 - < f_4, w_2 > w_2 - < f_4, w_1 > w_1$$
$$\Rightarrow w_4 = u_4/||u_4||$$
$$u_5 = f_5 - < f_5, w_4 > w_4 - < f_5, w_3 > w_3 - < f_5, w_2 > w_2 - < f_5, w_1 > w_1$$
$$\Rightarrow w_5 = u_5/||u_5||$$

This yields the following matrix form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{||u_5||} \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -< f_5, w_1 > & -< f_5, w_2 > & -< f_5, w_3 > & -< f_5, w_4 > & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{||u_4||} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -< f_4, w_1 > & -< f_4, w_2 > & -< f_4, w_3 > & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{||u_3||} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -< f_3, w_1 > & -< f_3, w_2 > & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{||u_2||} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -< f_2, w_1 > & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} \frac{1}{||u_1||} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$$

The various matrices are referred to as E_1,E_2 ... E_{10}, labeled from right to left. The steps deriving w_i correspond to the elementary operations E_{2i}, E_{2i-1}. We

thus obtain an orthonormal basis $\{w_i\}$ via the matrix $Q = \prod_i E_i$. This gives:

$$Q^{-1} = \begin{bmatrix} 1.2247 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.8165 & 0.9129 & 0.0000 & 0.0000 & 0.0000 \\ 0.2041 & 0.3651 & 1.1511 & 0.0000 & 0.0000 \\ 0.4028 & 1.0042 & 0.4778 & 0.3110 & 0.0000 \\ 0.4082 & -0.0913 & 1.0425 & -0.2351 & 0.4277 \end{bmatrix}$$

A complete explanation of this approach, along with examples, is found in our previous work [22].

3.3.5 Word-to-Word Semantic Distance

The method described allows us to compute effective, quantitative, word-level semantic distance. Figure 3.7 outlines the algorithm, which we use as the foundation for our document-level semantic distance metric.

1. *Select a set of basis chains* $B = \{c_k\}$
2. *Orthonormalize* B *and derive the matrix* $Q = \prod_i E_i$*, where the* E_i*'s are the elementary row operations of the Gram-Schmidt procedure*
3. *Given a new lexeme,* l
   (a) *Derive the vector* $\vec{l}$ *using the function* m_{c_k} *for each basis chain* c_k *(i.e., the coordinates of a chain containing* l *in* B*)*
   (b) *Compute the transformed vector* $\vec{l'} = \vec{l}Q^{-1}$
4. *For any two lexemes,* l_1 *and* l_2
   (a) *Compute* $\vec{l_1'}$ *and* $\vec{l_2'}$ *as per step 3*
   (b) $d(l_1, l_2) = arccos\frac{<\vec{l_1'}, \vec{l_2'}>}{\|\vec{l_1'}\|\|\vec{l_2'}\|}$

Figure 3.7: Word-to-word semantic distance computation by Jensen et al. [22]

3.4 Word-to-Word Preliminary Results

We implemented this approach for word-to-word comparisons and obtained encouraging results [22]. As mentioned in Chapter 2, Budanitsky and Hirst evaluated five WordNet based measures of semantic relatedness[7]. One of their evaluations was performed by comparing distance scores with those determined by humans for 65 pairs of nouns provided by Rubenstein and Goodenough[49]. These pairs of words have been scored by humans according to their semantic relatedness. A group of 51 humans were asked to provide scores from 0.0, denoting that the words were semantically unrelated, to 4.0, a score which represented a relationship of synonymy. The scores of these 51 human participants were then averaged and provided with the 65 noun pairs to comprise a data set. Our evaluation metric for this data set is the correlation of our distance scores with the relatedness measures of the humans. It is common for words to have more than one sense. For example, "crane" may refer to the large construction tool, or to a long-necked bird. For this experiment our linear transformation was generated using maximal chains of the dominant sense of each of the words in the experiment (or mutually triggering senses where it applied, such as in the case of "bird" and "crane"). Table 3.1 shows our distance measures (from 0° to 90° where the smaller the angle of distance, the more conceptually related two words are).

The correlation of our selected metric with the human scores is **-0.802**. The negative sign of the correlation simply recognizes that the human scores were measures of similarity, and ours were measures of distance. We note that this correlation is comparable to, and in some cases more favorable than the correlations of five other semantic measures examined by Budanitsky and Hirst on the same data set (see Table 4.2) [7]. This score was also obtained with very little system tweaking and we believe that it could be improved with additional effort. We also note that our focus is not limited to achieving a perfect correlation with the humans in this experiment. Rather,

w_1	w_2	humans	$d(w_1, w_2)$	w_1	w_2	humans	$d(w_1, w_2)$
cord	smile	0.02	**33.16**	car	journey	1.55	**27.65**
rooster	voyage	0.04	**35.92**	cemetery	mound	1.69	**7.2**
noon	string	0.04	**29.84**	glass	jewel	1.78	**1.65**
fruit	furnace	0.05	**5.56**	magician	oracle	1.82	**5.96**
autograph	shore	0.06	**27.01**	crane	implement	2.37	**0.84**
automobile	wizard	0.11	**25.61**	brother	lad	2.41	**3.74**
mound	stove	0.14	**24.37**	sage	wizard	2.46	**13.38**
grin	implement	0.18	**32.13**	oracle	sage	2.61	**8.82**
asylum	fruit	0.19	**24.69**	bird	crane	2.63	**0.00**
asylum	monk	0.39	**33.05**	bird	cock	2.63	**0.00**
graveyard	madhouse	0.42	**7.56**	food	fruit	2.69	**1.13**
glass	magician	0.44	**24.8**	brother	monk	2.74	**0.00**
boy	rooster	0.44	**27.43**	asylum	madhouse	3.04	**0.21**
cushion	jewel	0.45	**24.41**	furnace	stove	3.11	**1.7**
monk	slave	0.57	**20.57**	magician	wizard	3.21	**0.00**
asylum	cemetery	0.79	**7.48**	hill	mound	3.29	**0.00**
coast	forest	0.85	**17.44**	cord	string	3.41	**0.00**
grin	lad	0.88	**29.32**	glass	tumbler	3.45	**1.21E-6**
shore	woodland	0.90	**12.64**	grin	smile	3.46	**0.00**
monk	oracle	0.91	**19.01**	serf	slave	3.46	**4.02**
boy	sage	0.96	**18.67**	journey	voyage	3.58	**0.00**
automobile	cushion	0.97	**4.18**	autograph	signature	3.59	**1.21E-6**
mound	shore	0.97	**10.34**	coast	shore	3.60	**2.86**
lad	wizard	0.99	**6.85**	forest	woodland	3.65	**1.21E-6**
forest	graveyard	1.00	**12.13**	implement	tool	3.66	**0.4**
food	rooster	1.09	**30.65**	cock	rooster	3.68	**8.54E-7**
cemetery	woodland	1.18	**12.13**	boy	lad	3.82	**0.00**
shore	voyage	1.22	**27.11**	cushion	pillow	3.84	**0.2**
bird	woodland	1.24	**13.34**	cemetery	graveyard	3.88	**8.54E-7**
coast	hill	1.26	**8.07**	automobile	car	3.92	**0.00**
furnace	implement	1.37	**1.72**	midday	noon	3.94	**0.00**
crane	rooster	1.41	**4.72**	gem	jewel	3.94	**0.00**
hill	woodland	1.48	**12.51**				

Table 3.1: 65 noun pairs provided by Rubenstein and Goodenough along with human similarity scores, and our distance measures [49]. The correlation of our scores with the humans was **-0.802**.

Measure	**Correlation w/ humans**
Leacock and Chodorow	.838
Lin	.819
Jensen et al. [22]	**.802**
Hirst and St-Onge	.786
Jiang and Conrath	.781
Resnik	.779

Table 3.2: Our correlation performance in comparison to five other approaches.

we mention it because it lends empirical credibility to our motivation to build upon this method of word-to-word semantic comparison. The significant contributions of

this thesis are demonstrated in the way that this word-to-word semantic distance measure is extended to create a document level semantic distance metric that can be used to semantically compare whole documents.

3.5 Implementation Details

Our implementation of word-to-word semantic distance computation, as previously described, is in C++, using WordNet 3.0 for UNIX. This section describes a few improvements that we made with respect to optimizations and caching, as well as our considerations of runtime and space complexity.

3.5.1 Optimizations and Caching

The algorithm we have presented for computing word-to-word semantic distances (see Figure 3.7), does not, by itself, run fast enough to be used practically on a large scale. However, many parts of the algorithm can be made to run much faster by following a series of precomputational steps, whose results are stored in memory to be accessed when making the actual word-to-word semantic distance computations. First, consider steps 1 and 2 of the algorithm shown in Figure 3.7. Selecting a set of basis chains, and determining the linear transformation that is used to map words into the metric space, need only to be computed once. We thus take these steps, and store Q^{-1} in memory. Next, for each word l in the knowledge structure (in our case WordNet 3.0), we compute the vector $\vec{l}$, described in part 3 a), and index the vectors as key value pairs in memory. We end up with a hash table, diagrammed in Table 3.3.

Later, as we discuss space complexity, it will become clear why this precomputed structure is significantly more advantageous than the precomputed structures that are required for other approaches.

l	$\vec{l}$
football	[2.36, 0.23, ... , 1.23, 0.93]
building	[0.91, 0.75, ... , 3.31, 0.44]
election	[1.03, 1.23, ... , 1.45, 2.30]
⋮	⋮

Table 3.3: Precomputing mapped vectors for words.

3.5.2 Runtime Complexity (Word-to-Word)

Computing chains is possible to do very quickly when the semantic relationships have been indexed (i.e., the graph does not need to be re-traversed every time to identify chains for words). This is the case for WordNet, which we use as the lexical structure in our experiments. Thus, the runtime complexity for computing chains for a given word is $O(b * n)$ where b is the branching factor and n is the number of nodes in the longest chain. In practice, the branching factor for words in the WordNet lexicon, is often very close to one. Furthermore, practice has also shown most WordNet hypernymy chains to have a length of less than 20. Thus computing chains in practice is very fast in our experiments. With chains for given words, we can then compute intersection points, which are used to generate the previously mentioned function values. Computing the lowest intersection for two chains is bounded by $O(n^2)$.

Additionally, after precomputing Table 3.3, semantic distances computations can be performed in constant time ($O(1)$), because the size of the mapped vector for any given word is bounded by a very small number (related to the depth of the hierarchical structure). This is true, first, because word vectors can be looked up in constant time (by using a hash table). Secondly, because we can treat the length of the mapped vectors as a small constant (it does not change as the number of semantic comparisons increases), computing the cosine distance of two vectors also runs in constant time.

3.5.3 Space Complexity (Word-to-Word)

While we acknowledge that our effectiveness is comparable to that of the other WordNet-based approaches (i.e., those from Table 3.2), our distance computation presents distinct advantages with respect to practical usability. Recall that many of the other similarity metrics operate directly on the knowledge structure [7]. This process can be time consuming in a practical application, and would thus need to be precomputed to be used efficiently. Consider the case of precomputing path lengths between words in the WordNet semantic relationship graph. Computing the path lengths of all possible combinations of words would require $24,025,000,000$ different entries (for the $155,000+$ words), equating to approximately 96GB of memory (assuming 32 bit integers are used to represent path lengths). This storage is bounded by $O(n^2)$, where n is the number of words in the structure. A partial visual representation of this is shown in Figure 3.8:

	$word_1$	$word_2$	$\cdots$
$word_1$	0	5	$\cdots$
$word_2$	13	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$

Figure 3.8: Storing path lengths for all combinations of words.

Conversely, our embedding of the lexicon into a metric space allows us to represent words as single vectors. Having vector representations of two words allows us to quickly compute the distance between the words according to our method as described. Thus, only the vector for each word needs to be precomputed, to be used in an efficient, practical manner. Based on the example of $155,000$ words, only $155,000$ vectors need to be precomputed and stored. This equates to approximately 47MB (assuming each vector is able to be represented in less than 300 bytes). This storage is bounded by a much smaller $O(n)$, where n is the number of words in the structure. A partial visual representation of this is shown in Figure 3.9:

$word_1$	[2.36, 0.23, ... , 1.23, 0.93]
$word_2$	[0.91, 0.75, ... , 3.31, 0.44]
$word_3$	[1.03, 1.23, ... , 1.45, 2.30]
⋮	⋮

Figure 3.9: Storing mapped word vectors.

This representation allows us to store word vectors in a reasonable amount of memory. Having the vectors accessible in memory allows us to perform very fast distance computations between two words on the fly, and is thus much more practical for use in real world applications.

Chapter 4

Document Level Distance Metrics

4.1 Objective

Let Equation 4.1 mathematically represent our goal to provide a quantitative distance between two documents.

$$distance(doc_a, doc_b) \in \mathbb{R} \tag{4.1}$$

In this paper we discuss three methods for computing document-level semantic distances according to Equation 4.1.

4.2 Document Representation

In order to extend the word-level semantic distance measure that we have selected, we must formulate documents such that they are conducive to this word-level metric. Essentially we employ a filtered bag-of-words to represent documents, where we select only nouns for representation. We then take the most frequently occurring (top N) nouns to represent the document, and make the assumption that these best reflect the concepts of the document. The knowledge-based structure that we use as the foundation of our word-level semantic distance metric (WordNet 3.0) induces this restriction on us as we only use it to capture semantic relationships for nouns. We believe, however, that nouns are good indicators of topics or concepts that define documents, and that this approach is advantageous because of its reduction of document

dimensionality (particularly where N is small). To extract nouns from our data, we use an English part-of-speech tagger defined by Y. Tsuruoka [58]. This tagger has been shown to achieve 97.1% tag classification accuracy on the Wall Street Journal corpus, and works very well for our noun extraction purposes. This process can be seen in Figure 4.1. When we refer to documents in the remainder of the paper, we mean documents that have been reduced as described here.

Note that choosing the correct value for N is one instance of parameter selection among several throughout our experiments. In the interest of time we generally performed parameter selection by choosing the best performing endpoint value in the parameter space, then greedily exploring the search space around the chosen endpoint. A more complex exploration of the complete parameter space may have resulted in better overall performance.

Chad's interim parliament has changed the electoral law,
banning independents from a November 24 parliamentary poll
wrapping up its transition to multi-party democracy...

Figure 4.1: Reducing document dimensionality by selecting nouns.

4.3 Average Semantic Distance for Documents

To compute an average semantic distance for documents, we take the cross-term average for all terms in the two documents. This is formalized in Equation 4.2.

$$distance_{avg}(doc_a, doc_b) = \frac{\sum_{t_x \in doc_a} \left(\frac{\sum_{t_y \in doc_b} distance(t_x, t_y)}{|doc_b|} \right)}{|doc_a|} \tag{4.2}$$

This average distance can be visualized in Figure 4.2 where the polygons represent documents, and each point in the polygon represents a word in the document.

The line separating the polygons represents the average semantic distance between them. Sometimes this distance is referred to as the "average-link" distance.

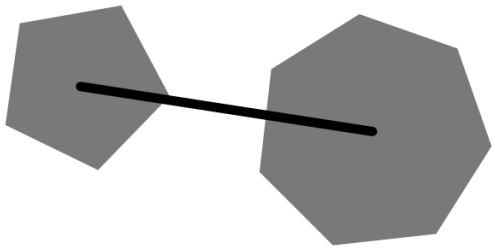

Figure 4.2: Visualizing average semantic distance between documents.

4.4 Hausdorff Semantic Distance for Documents

To maintain consistency with our basic approach at the word-level, driven by *topological* arguments, we adopt the Hausdorff distance, defined as follows.

$$\begin{aligned} \mathrm{H}(d_1, d_2) &= \max_{n \in d_1}(\min_{n' \in d_2}(\mathrm{dist}(n, n'))) \\ \mathrm{H}(d_2, d_1) &= \max_{n' \in d_2}(\min_{n \in d_1}(\mathrm{dist}(n, n'))) \\ \mathrm{Hausdorff}(d_1, d_2) &= \max\{\mathrm{H}(d_1, d_2), \mathrm{H}(d_2, d_1)\} \end{aligned} \tag{4.3}$$

For each noun in a document, the minimum word-level distance to all nouns in the other document is found. Then, the maximum of all of these minimum distances is taken as the distance from the first document to the second one, as depicted in Figure 4.3. This process is repeated reversing the order of the documents, since that measure is clearly not symmetric. Finally, the maximum of the two distances is the Hausdorff distance between the documents.

Figure 4.4 shows the pseudo code to determine document level Hausdorff distance between two documents.

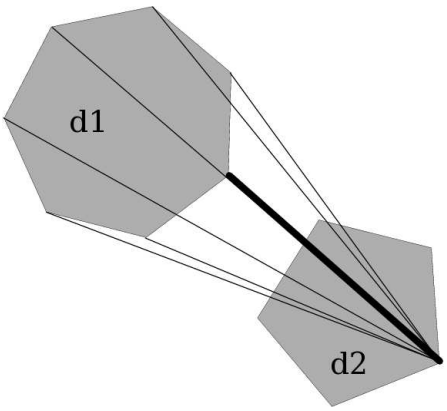

Figure 4.3: Visualizing a Hausdorff distance from document d_1 to d_2.

```
def hausdorff_dist_aux(doc_a, doc_b):
    max_ab = -Inf
    for t_x in doc_a:
        min_ab = Inf

        for t_y in doc_b:
            dist = d(t_x, t_y)
            if dist < min_ab:
                min_ab = dist

        if min_ab > max_ab:
            max_ab = min_ab

    return max_ab

def hausdorff_dist(doc_a, doc_b):
    return max(hausdorff_dist_aux(doc_a, doc_b),
               hausdorff_dist_aux(doc_b, doc_a))
```

Figure 4.4: Pseudo code for determining a Hausdorff distance between documents.

4.5 Document Distance Through Aligned Word Clusters

Typical approaches to determining a suitable distance measure between documents tend to make the implicit assumption that documents have a single semantic topic. What if that assumption is violated? Consider the following contrived example of two

documents, each comprised of two semantic topics, described by word clusters (i.e., sets of words that have a small word-level semantic distance from each other): $d_1 = \{$ ["house", "apartment"], ["book"] $\}$ and $d_2 = \{$ ["condo", "mansion"], ["basketball"] $\}$. Using the Hausdorff distance, we might find that the distance between these two documents is measured by the distance between "book" and "mansion" (i.e, their word-to-word distance is the maximum of all the minimum word-to-word distances for the two documents). Then, the documents would appear rather distant from each other. On the other hand, if we somehow align the word clusters and compute the distance between them, we would find that the distance between the first clusters in d_1 and d_2 is rather small, and hence, the documents could be deemed rather close. In this section, we describe a method to handle the clustering of documents with multiple semantic topics.

Forming Word Clusters

Word clusters are formed in a naive, agglomerative approach. To allow for consistent document alignment, each document contains a fixed number of word clusters which contain zero or more words.[1] Each word cluster is seeded with a random word from the document representation, whereupon the remaining words from the document representation are iteratively added to the most similar word cluster. A particular word's similarity to a given word cluster is defined as the minimum word-level semantic distance with the words in the cluster.

Clearly, word clusters can be formed with more complex (and possibly effective) clustering techniques, however the above approach has been chosen in the interest of runtime, as it is comparatively fast in practice.

[1]Note that, in the limit, word clusters could consist of a single word each. Word clusters reduce the complexity of computing the document level distance.

Document Level Distance with Word Clusters

With documents that are represented by word clusters, we compute the document-level distance, as depicted in Figure 4.5, by: 1) finding the best alignment of word clusters; and 2) summing the distances of the aligned word clusters. The process of finding the best alignment of word clusters consists of beginning with the first word cluster in d_1, and computing a Hausdorff distance between that cluster, and all the remaining, available word clusters in d_2. The distance between word clusters is computed by treating each word cluster as a small document, then computing the Hausdorff distance between the two word clusters. A word cluster in d_2 is identified as unavailable when it has already been aligned with another word cluster in d_1. This process is repeated for all word clusters in d_1, resulting in a complete alignment of d_1 and d_2.

The distances between the thus aligned word clusters are then summed up to form a document-level distance. The summation results in a normalized value because all documents contain the same number of word clusters, and the Hausdorff distance between word clusters is normalized to [0.0, 90.0] (degrees).

Advantages and Drawbacks of Word Cluster Alignment

One advantage of this method is that it runs slightly faster than the normal symmetric Hausdorff distance. While it is bounded by the same $O(n \times m)$ (where n and m represent the number of words in d_1, d_2 respectively), the preceding multiplicative constants are smaller because the number of cluster comparisons decreases as clusters are aligned and become unavailable. Another advantage of this approach is that it attempts to handle documents with multiple, and potentially differing, semantic topics. This approach, however, is not as theoretically satisfying as the Hausdorff distance, in the context of a topological mapping (the basis for our word-level semantic

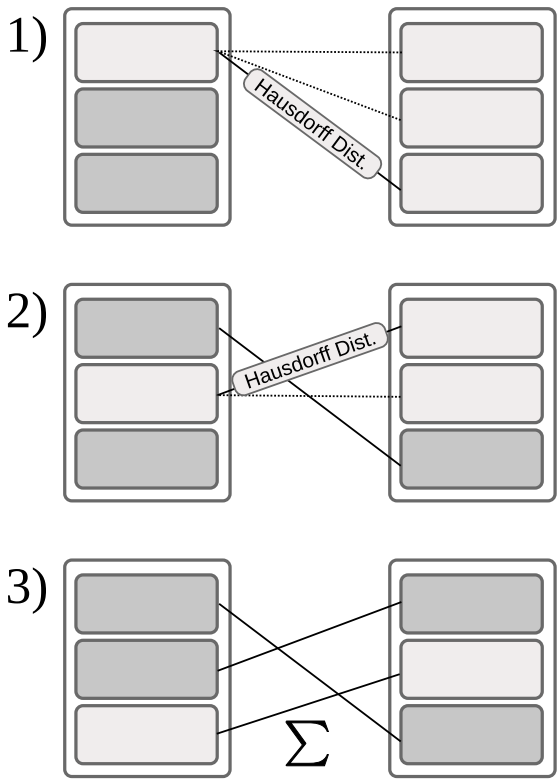

Figure 4.5: Aligning word clusters. Once all clusters have been aligned, the distances are summed to form a document level distance metric.

distance metric). A challenge it presents is determining the best number of word clusters in documents.

4.6 Discussion

We asses the effectiveness of our semantic distance by examining performance metrics designed to model relevancy and conceptual relatedness as observed by humans. Of the three document-level semantic distance metrics we have introduced here, we found that in a large majority of our experiments the Hausdorff semantic distance metric outperformed the average semantic distance and the document-level distance that utilizes aligned word clusters. Taking this into consideration, we present a series of

experiments in which we used each of these distance metrics. The results that we present are those that were achieved using the most effective document-level distance metric, which was the Hausdorff semantic distance metric unless otherwise stated.

Chapter 5

Analysis of Integration with Search Indexing

A popular approach to document retrieval is to use an inverted index [68]. An inverted index maps words to documents in a structure similar to that of Figure 5.1. More complex approaches which allow for relevancy ranking are often used in combination with an inverted index, such as the Vector Space Model (VSM) [53]. This chapter describes our attempt at extending the VSM indexing method by incorporating a level of semantic analysis.

...
`computer`: $[doc_5, doc_{22}, doc_{27}]$
`football`: $[doc_3, doc_{12}]$
`politics`: $[doc_1, doc_7, doc_5]$
...

Figure 5.1: An example inverted index.

5.1 Expanding a Search Index

One of the limitations of the VSM indexing approach is its sensitivity to semantics. Particularly, if documents contain similar concepts, but are expressed with different vocabularies, they will not be strongly associated. However, because of its structure, retrieval using the VSM indexing approach is very fast. We attempt to improve the VSM indexing approach by identifying for each document, *additional* terms that are semantically similar to the terms in the document, and then including those terms in the index. The degree to which the additional terms are associated with the document

depends on their VSM term frequency-inverse document frequency (TF-IDF) scores, and a weighted factor that is proportional to the semantic distance between the term and the document. Thus, speed of retrieval can remain high, *with* the consideration of higher-order semantic information.

5.1.1 Identifying Semantic Terms

We enumerate a set of semantically similar terms to each word in a document by: 1) identifying the location of the word in the WordNet structure; 2) adding terms that are either siblings or descendants of the word, in a breadth first manner; and 3) stopping when a predefined, maximum number of terms has been reached, or when all the descendants of the word have been added. If a sibling or descendant term already exists in the document, it is not added to the set. The predefined maximum number of terms that we use for this experiment is 20 as we empirically found this to yield the best results. This process is visualized in Figure 5.2, where the word of interest (identified by 1) is located, and terms identified by 2-8 are used to form the set of semantically similar words.

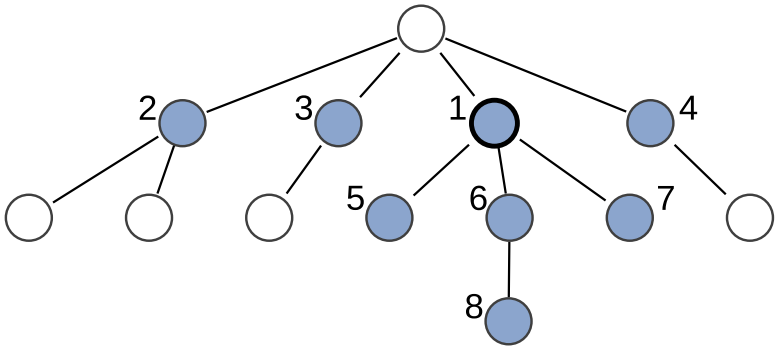

Figure 5.2: Identifying semantic terms for indexing.

5.1.2 Integration with Vector Space Model

With a set of semantically similar terms for each word, w, in a document, d, we now must identify to which degree each new, additional semantic term, t, should be

associated with the document. We first apply a weight, $\omega_{sem}(t, d)$, to each additional semantic term t relative to document d, as follows.

$$\omega_{sem}(t, d) = (1 - \frac{H(t, d)}{90})^2$$

This process produces weight values between 0 and 1, with values closer to 1 denoting terms that are semantically close to the document. The weighting diminishes exponentially as the distance between the term and the document increases. Finally, the new index term's weight, $\omega(t, d)$, is the product of $w_{sem}(t, d)$ and the original TF-IDF weight $\omega_{TFIDF}(w, d)$ of the word in d that it was derived from, namely:

$$\omega(t, d) = \omega_{sem}(t, d) \times \omega_{TFIDF}(w, d)$$

In this experiment, we use the C++ port of the very popular and open source Lucene information retrieval library [43]. Lucene supports traditional VSM retrieval by performing standard TF-IDF term indexing. In order to assess the value of our proposed technique, we compare the performance of retrieving documents using Lucene with its built-in indexing mechanism (baseline), with the performance of retrieving documents using Lucene with the addition of the new terms and their associated weights as input.

5.2 Data Set

We measure our performance on a series of queries and relevancy judgments provided by the NIST sponsored Text REtrieval Conference [39]. Particularly, we use the Aquaint data set from TREC Disks 4 and 5 [59]. The Aquaint data set is comprised of approximately 1,033,000 English, plain-text documents (primarily news articles from the New York Times and Xinhua news services).

5.2.1 Queries and Relevancy Judgments

The Aquaint data set is annotated with 50 queries, with associated relevancy judgments for each query. The queries are derived from topics, each with a title, description and narrative. One of the topics from the Aquaint data set is shown in Figure 5.3.

```
<top>

<num> Number: 303
<title> Hubble Telescope Achievements

<desc> Description:
Identify positive accomplishments of the Hubble telescope since it
was launched in 1991.

<narr> Narrative:
Documents are relevant that show the Hubble telescope has produced
new data, better quality data than previously available, data that
has increased human knowledge of the universe, or data that has led
to disproving previously existing theories or hypotheses.  Documents
limited to the shortcomings of the telescope would be irrelevant.
Details of repairs or modifications to the telescope without
reference to positive achievements would not be relevant.

</top>
```

Figure 5.3: An Aquaint data set topic.

In our experiments, and consistent with the TREC experiment requirements, we use the title of each topic as a keyword query [59]. Information in the description and narrative provide additional insight to the human ranking process that occurred when the data set was formed, but are not available to the systems in our experiments. For each query, the Aquaint data set provides a relevancy judgment that associates documents with the query. Our experimental results are compared to these relevancy judgments to compute evaluation metrics.

5.3 Results and Discussion

We evaluate our performance on common metrics in the fields of information retrieval, specifically:

- **Precision** - the number of relevant retrieved documents, divided by the number of retrieved documents. The specialized cases of precision that we measure include:
  - **Mean Average Precision** - the mean of the precision scores after each relevant document in the result set is retrieved.
  - **Precision after five documents** - The precision score that considers the first five documents in the result set.
  - **Precision after ten documents** - The precision score that considers the first ten documents in the result set.
- **Recall** - the number of relevant retrieved documents divided by the total number of relevant documents.

The results that we present here are averages for all 50 queries. We compare the results of our method (Semantic Expansion Index) with the results of the standard VSM TF-IDF indexing approach provided by Lucene (Baseline). These results are included in Table 5.1.

Metric	Baseline	Semantic Expansion Index	Improvement
Mean Average Precision	0.0854	0.0615	-27.9859%
Precision after 5 docs	0.3080	0.3080	0.0000%
Precision after 10 docs	0.2780	0.2780	0.0000%
Recall	0.4373	0.1725	-60.5534%

Table 5.1: Metric scores for semantic index expansion.

Our motivation in adding a layer of semantic analysis to the TF-IDF indexing approach was to associate documents that contained similar concepts but were

expressed with different vocabularies. In examining the performance of our method, it is important to note that of the three observed metrics in our experiments, the metric that is arguably the most important in user facing search engines, is precision after five documents, as this measure accounts for the relevancy of the results that are first viewed, and where there is the greatest expectation for relevancy. In our experiment, our precision after five documents is no worse than the baseline. It is still no worse after ten documents, but finally drops off in the long tail of precision after n documents. Consequently, this loss in precision in the long tail results in a much lower recall.

After investigating the causes for our drop in precision in the long tail, we were able to identify some contributing factors. The most significant challenge was the result of the nature of the semantic relationships used to perform semantic evaluation. Specifically, consider the query from Figure 5.3: "Hubble Telescope Achievements". Note as well that the narrative specifies that documents limited to the shortcomings of the telescope would be irrelevant, which clarification, however, is not available to the system for analysis. In this case we found that "success", a synonym of "achievement", is a direct sibling of "failure", under "occurrence" in the WordNet hypernymy/hyponymy tree. Consequently, we found that in this case the term "achievement" was also actually associated with documents that only mentioned failures of the Hubble Telescope in addition to those that mentioned Hubble Telescope successes. The term"failure" and "success" are semantically "close" to each other according to our hypernymy tree.

We recognize that there are several alternative methods for implementing term indexing in IR, however, given our performance relative to the baseline (no improvement), we did not find it necessary to compare our approach with other methods. We should state however, that although our performance in this experiment is not extraordinary, our results are necessarily obviously poor either. They could be due to

bias in the formulation of the Aquaint data set, which may favor lexical and keyword relationships rather than conceptual relationships. Our results are certainly different than the baseline in the long tail, however, it is possible that this difference is caused by an approach that may yield better results with a data set that favored conceptual relationships between words, particularly hypernymy.

Chapter 6

Analysis of Search Result Re-ranking and Query Expansion

6.1 Post Processing Procedures

In addition to using a semantic distance measure to alter search indexes, we also present methods for re-ranking search results and expanding keyword queries. These two methods define alternative uses of a semantic distance measure in improving common information retrieval measures. To re-rank search results, we use the document level distance metrics that have been described to determine the closeness of documents to a particular query for which the documents were retrieved. For query expansion, we identify additional query terms by extracting words from documents found in a preliminary result set that are determined to be semantically similar to the original query terms. When we discuss improving search results (both with search result re-ranking, and query expansion) we do so in the context of retrieving documents from a document collection that are determined to be relevant to user's keyword queries.

These approaches are post-processing steps in that they perform some function after an initial result set is obtained by a document retrieval system. In this chapter we explain the details of both of these approaches, as well as the results that these methods obtained in practical experiments.

6.2 Search Result Re-ranking

A purely semantic algorithm for retrieving documents that are relevant to a particular keyword query can be simply defined given a document level semantic distance measure. By representing a keyword query as a document we can take the closest documents in the collection to the query and return them as the result set. This approach is very impractical with a document set of any consequential size. It could potentially take a very long time to form a result set using this method. So, to reduce the runtime, we propose a much faster solution which effectively re-ranks search results determined by a much faster IR technique.

6.2.1 Overview

To overcome the large amount of runtime required to formulate a result set using the method described in Equation 6.1 we first create an initial result by using a common information retrieval model. Our approach to re-ranking results can build on top of any IR model that can retrieve relevant documents for user generated keyword queries. For the process we describe in this section, and for the results which follow, we use a Vector Space Model (VSM) to generate result sets because it is a very common method and can generate result sets very quickly. Then, using an initial

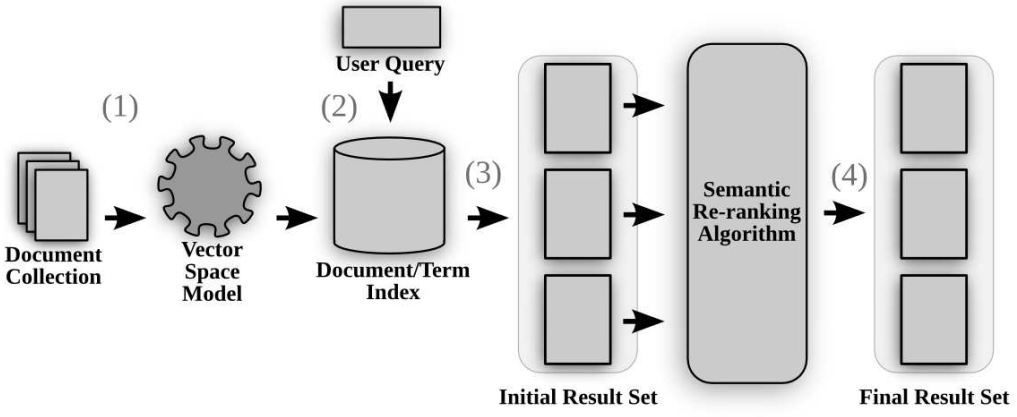

Figure 6.1: Re-ranking search results using semantic distance.

result set that has been generated by a vector space IR model, for a particular query, we re-rank that set of documents by measuring each document's semantic distance from the query, and ordering by distance in an ascending manner. We present the re-ordered result set as the final results. This process is shown in Figure 6.1.

6.2.2 Semantic Re-ranking Algorithm

The actual method for re-ranking retains the theoretically pleasing, full semantic measure between documents, but operates much faster because of the reduced size of the initial result set in comparison to a full document collection. Using an initial result set, we produce a new result set for a query by re-ranking documents in the result set such that the documents are ascendingly ordered by their semantic distance to to the query. This method supports any of the document-level semantic distances described in Chapter 4, however we present results that rely on the use of a Hausdorff semantic distance as that method outperformed the other types of document level distances.

6.2.3 Data and Experimental Results

We conducted an experiment of the semantic re-ranking that we have described, using the same data from Chapter 5. This Aquaint text collection consists of 1,033,461 plain text, English documents taken from the New York Times, the Associated Press, and the Xinhua News Agency newswires. 50 prepared keyword queries received relevancy judgments that act as our gold standard.

Recalling the process shown in Figure 6.1 we first index all of the documents in the collection by using an off-the-shelf, vector space model implementation called Lucene (as mentioned in the last chapter). From the index that is created with Lucene, for the terms in the collection, we can generate 50 initial result sets, each corresponding to a given keyword query from the Aquaint data set. By querying the

index to obtain the result sets, we can then compute baseline scores for the following metrics:

- Mean Average Precision
- Precision after 5 Documents
- Precision after 10 Documents

We report scores for these metrics as averages for all 50 queries. We do not report recall because it is irrelevant in this context, as we do not change the set documents that is retrieved, only the order in which the documents are ranked. The results are shown in Table 6.1.

Metric	**Baseline**	**With Re-ranking**	**Improvement**
Mean Average Precision	0.0854	**0.0872**	**+2.1077%**
Precision after 5 docs	0.3080	**0.3200**	**+3.8961%**
Precision after 10 docs	0.2780	**0.2860**	**+2.8777%**

Table 6.1: Precision metrics for result set re-ranking.

6.2.4 Discussion

We report that by using our semantic re-ranking algorithm, which uses a symmetric Hausdorff document distance, we improved all three observed precision metrics. It is also notable that this process can occur quickly when using a reduced document representation (using a topN cutoff that has been described), and when the original result set is small. For each of the 50 queries that we examined in this experiment, the result set size was 1000 total documents. We used a *topN* value of 20, which was chosen empirically.

As we have previously mentioned, due to computational complexity, it is impractical (with currently available computational resources) to implement a search engine that uses a pure semantic document distance method as previously described.

However, the results (see Table 6.1) suggest that measuring relevancy with the semantic document distance we have described can be effective.

Finall, we note that we improve precision most significantly after five documents, which is encouraging given the importance of this metric with respect to user facing search engines.

6.3 Semantic Query Expansion

Another way to improve document retrieval is through query expansion. When we discuss query expansion in this chapter, we simply mean the process of adding additional terms to a user keyword query. This definition is simple, and it allows us to describe a method that can easily be used to augment several different types of keyword information retrieval systems. Formally, we describe this with Equation 6.1 (where || represents concatenation).

$$q' = (q \;||\; semantically\ similar\ terms) \tag{6.1}$$

We formulate a new query q', which is a result of concatenating our original keyword query q (which can have one or more keyword terms), with new terms that have been identified as semantically similar. The definition of query expansion in our case does not take into account weightings of particular query terms, ordering of terms, or any boolean logical operations. For our purpose, query expansion is the process of adding additional terms to an existing query, such that the new query contains a superset of keyword terms, and can be issued to a keyword IR system (in our case a Vector Space Model IR system). In this section we describe the process of using semantic distance to choose new, additional query terms, and the results of appending these query terms to original queries.

6.3.1 Method for Expanding Queries

In this experiment, our retrieval system looks similar to that of the last system, with some modifications. We again take an existing user keyword query and issue it to an existing IR system. We use the same baseline Vector Space Model IR system as described in the previous section to build upon. By issuing a user keyword query to this IR system we formulate an initial result set R (see Equation 6.2).

$$R = VSM(q, D) \tag{6.2}$$

In this experiment, as with the last, we again attempt to leverage information that exists in this initial result set, to improve the final results. We do this by identifying keyword terms that exist in the result set R, that are semantically similar to the initial keyword query terms q, but that do not occur in the initial query. We expand the query q to form the new query q' as shown in Equation 6.3.

$$q' = (q \mid\mid extractSemanticTerms(\sigma_{top\ \delta}(R))) \tag{6.3}$$

where $extractSemanticTerms$ is a selection function that extracts the most semantically similar terms from the result subset $\sigma_{top\ \delta}$ which is made up of the top δ number of result documents from R, according to a standard VSM result set ranking. The most semantically similar terms are those with the smallest Hausdorff document-level semantic distance from the query. Here we treat each word as a document containing a single word, and the query as a document comprised of the individual query terms. Given this new query q', we are able to define a new result set R' by issuing our expanded query q' to the same VSM IR system (Equation 6.4).

$$R' = VSM(q', D) \tag{6.4}$$

Empirically we determined that using the top 10 documents from the initial result set R, and appending 50 new terms to q, to form q', yielded the best results. Figure 6.2 diagrams this process.

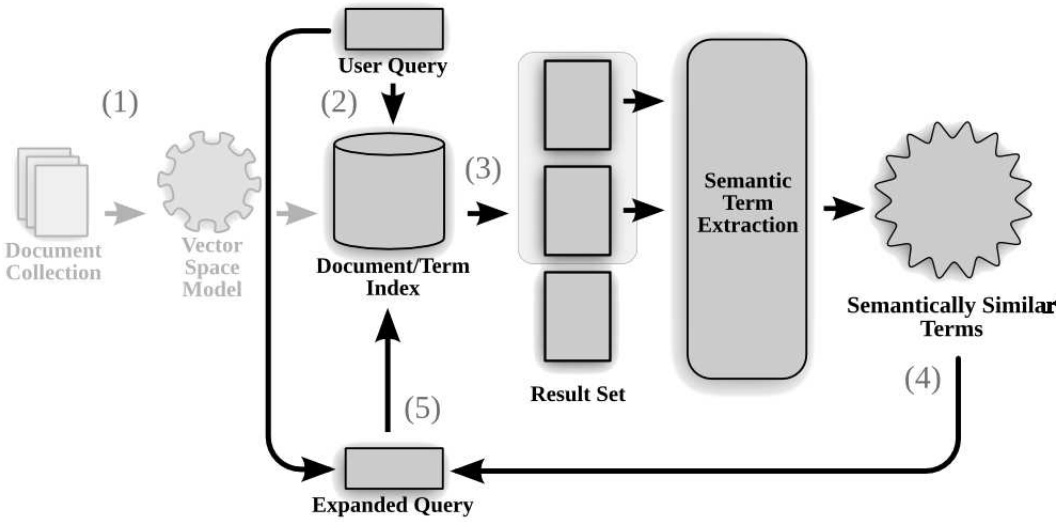

Figure 6.2: Expanding keyword queries using semantic distance.

6.3.2 Results

Using our method for query expansion, we were able to obtain significant improvements in average precision, and recall. The results are shown in Table 6.2

Metric	Baseline	Semantic Expansion	Improvement
Mean Average Precision	0.0854	**0.1023**	**+19.7892%**
Precision after 5 docs	0.3080	**0.3120**	**+1.2987%**
Precision after 10 docs	**0.2780**	0.2680	-3.5971%
Recall	0.4373	**0.4734**	**+8.2552%**

Table 6.2: Metric scores for semantic query expansion.

6.4 Query Expansion with LDA

We are pleased with the results that we achieved using our semantic query expansion method. In the process of our research of expanding queries, we also developed an

approach using Latent Dirichlet Allocation(LDA) [5]. For this particular problem, the method we developed with LDA performed quite well, and even better than our semantic approach. A short description of our approach using LDA is included here. Our approach using LDA is a corpus-based approach to identifying conceptual relatedness on a word-level, and thus differs greatly from the underlying methodology of our knowledge-based semantic approach. However, we include a brief overview of it in this thesis because it substantiates our claim that an analysis of word semantics can indeed improve the performance of information retrieval, and because of the significance of the results.

LDA has been used in IR experiments, but it is still a very new idea [61, 11]. Croft et al. uses LDA for document summarization and other pre-processing tasks[61]. We propose a method that uses LDA as part of a post-processing step to generate relevant keywords that can be used to expand a user query. Daume et al. use LDA in similar post-processing way, but require that relevancy judgments be provided for every query[11]. We operate in an unsupervised space where no relevancy judgments are necessary.

The query expansion system using LDA is very similar to the system we have described that uses our semantic distance to determine words to append to the existing query. Instead of using our semantic distance, however, we process the top 10 documents from our initial result set with LDA, parametrized with the assumption of one existing topic. For a more detailed explanation of LDA and its parameters, we refer the reader to [4]. The LDA process with these inputs probabilistically determines the terms that best indicate the topic of the 10 documents. These terms are ranked according to probability, where we take the 50 most probable terms, and append them to our initial query to create an expanded query. This process is shown in Figure 6.3

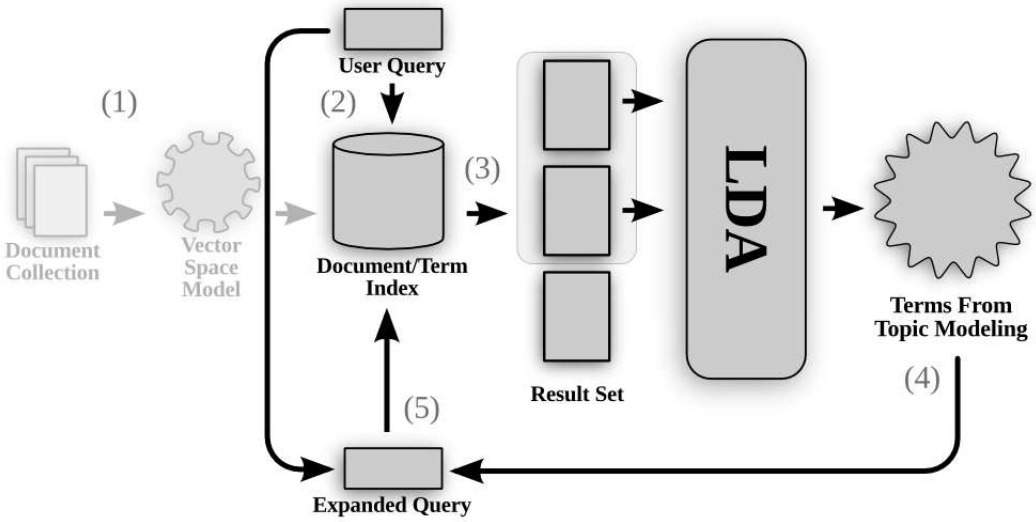

Figure 6.3: Expanding user keyword queries using LDA.

The improvements that this process yielded were also very significant, making substantial improvements for all of our observed metrics. The results are shown in Table 6.3

Metric	Baseline	LDA Expansion	Improvement
Mean Average Precision	0.0854	**0.1464**	**+71.4286%**
Precision after 5 docs	0.3080	**0.3640**	**+18.1818%**
Precision after 10 docs	0.2780	**0.3300**	**+18.7050%**
Recall	0.4373	**0.5376**	**+22.9361%**

Table 6.3: Metric scores for query expansion using LDA.

In our investigation of why our method with LDA outperformed our semantic distance method of query expansion, we found that the differences were related to the differences between corpus-based and knowledge-based approaches. Recall from Chapter 2 that corpus-based methods derive similarity measures from lexical and statistical information, and infer implicit semantic relationships within a text corpus. Knowledge-based approaches rely on explicitly defined semantic relationships, and specifically in our experiments, hypernymy. Now consider the query "Hubble Telescope Achievements". We noticed that our method with LDA chose "data" as a term to add to the initial query. By observing occurrences of "data" with terms from the

original query, LDA inferred a semantic relationship between them. However, "data" does not have a strong, explicit hypernymy relationship to any of the terms in the original query, and was thus not chosen by our semantic approach. Other situations like this resulted in LDA associating additional terms of this implicit type of relationship, which yielded better results overall, with the Aquaint data set. Although it has been the case in this particular experiment, the breadth of our research does not allow us to assert that corpus-based approaches, such as LDA, will always yield better results than knowledge based approaches.

6.5 Conclusion

We have shown that using our document-level distance metric to re-rank query results yielded higher levels of precision for the 50 queries of the very large, and real world Aquaint data set. We have also shown that the use of our semantic distance metric in expanding user keyword queries also improves precision and recall.

Chapter 7

Document Clustering using Semantic Distance

Document clustering is a common task that has become useful in a wide number of applications. This chapter proposes the use of our purely semantic distance metric in two classical distance based clustering algorithms.

7.1 Document Clustering with Distance Metrics

Traditionally, text documents are encoded with VSM. Once documents have been thus encoded, distance measures such as Jaccard index (for binary-valued vectors), Euclidean distance, cosine distance and Pearson similarity, are well-defined and can be used by distance-based clustering algorithms.

The standard taxonomy of distance-based clustering techniques splits methods into two general classes. Partitional methods, such as the well-known k-means algorithm, produce a single k-subset partition through iterative re-assignment of items to clusters, based on some distance to cluster centroids until all assignments remain unchanged (e.g., [35, 25, 10, 13, 31, 56, 27]). Hierarchical methods, such as hierarchical agglomerative clustering (HAC), produce a series of nested partitions, each representing a possible clustering, through iterative proximity-based merging (e.g., [24, 21, 63, 15]).

We propose the use of our document-level semantic distance with the k-means algorithm, and with hierarchical agglomerative clustering. Although distance met-

rics typically used with these clustering algorithms, such as cosine similarity and Euclidean distance, are effective to some degree, they lack the ability to cluster documents into *conceptually* similar groups. Using the Reuters-21578 data set, we show that the use of our semantic distance in these algorithms produces better clusters than cosine similarity or Euclidean distance, according to several cluster quality metrics.

7.2 Semantic k-Means

As stated above, the k-means algorithm is commonly used for document clustering. This algorithm is both effective and fast, with a linear runtime upper bound. The algorithm is as follows in Figure 7.1.

1. *Choose k centroids at random*
2. *Assign all documents to their closest centroid (using some distance metric)*
3. *Reposition all centroids to the center of the documents that have been assigned to them*
4. *Go back to step 2 until centroids do not change position*

Figure 7.1: The k-means algorithm.

We use our Hausdorff document-level semantic distance metric in step 2, to yield a semantically enabled k-means algorithm.

7.3 Experimental Results (k-means)

We report on experiments that compare the use of the cosine distance versus our semantic distance in the k-means clustering algorithm. We also report on further experiments where word clusters are computed prior to clustering. Our test data consists of a reduced version of the Reuters-21578 data set containing 200 documents from 5 classes. First, we turn our attention to the question of which cluster quality metrics to use.

7.3.1 Cluster Quality Metrics

One of the main challenges of research in data clustering is the assessment of the quality of the clusters resulting from some clustering process (e.g., [64, 54]). Here we choose a mixture of internal and external quality metrics as follows.

External Metrics

External metrics refer to those that can be computed with a comparison to actual classification labels. The data set used in our experiment (Reuters-21578) does contain classification labels for documents. A comparison of those labels with the clusters generated from the clustering algorithms allows for various external measures.

We selected the following external quality metrics: F-measure, Average Entropy and Adjusted Rand Index. Let $C_1, \ldots, C_n$ be the correct target clusters (as per the available labeling) and $HC_1, \ldots, HC_m$ be the computed clusters (as per the clustering algorithm). Then:

$$P(i,j) = \frac{|C_i \cap HC_j|}{|HC_j|}$$
$$R(i,j) = \frac{|C_i \cap HC_j|}{|C_i|}$$
$$F(i) = \max_{j=1}^{m} \frac{2.P(i,j).R(i,j)}{P(i,j)+R(i,j)}$$
$$F = \sum_{i=1}^{n} \frac{|C_i|}{N} F(i)$$

$$Entropy(HC_i) = \sum_{C_i \in C} -p(C_i|HC_i) log(p(C_i|HC_i))$$
$$AvgEntropy(HC) = \sum_{i=1}^{m} \frac{|HC_i|}{\sum_{j=1}^{m} |HC_j|} Entropy(HC_i)$$

Now let a be the number of pairs of documents in the same class in $C = C_1 \cup \ldots \cup C_n$ and the same cluster in $HC = HC_1 \cup \ldots \cup HC_m$. Let b be the number of pairs of documents in the same class in C but not in the same cluster in HC. Let c be the number of pairs of documents in the same cluster in HC but not in the same

class in C. Finally, let d be the number of pairs of documents in different classes and different clusters. The Rand Index (RI) is defined as (see [20]):

$$RI = \frac{a+b}{a+b+c+d} \tag{7.1}$$

The Adjusted Rand Index (ARI) is an extension of the Rand Index that addresses the fact that the expected value of the Rand Index for two random partitions does not take a constant value. In the Adjusted Rand Index Let n_{ij} be the number of objects that are in both class u_i and cluster v_j. Let n_i and n_j be the number of objects in class u_i and cluster v_j respectively. The Adjusted Rand Index is defined as (see [20]):

$$ARI = \frac{\sum_{i,j}\binom{n_{ij}}{2} - [\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}] - [\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}]/\binom{n}{2}} \tag{7.2}$$

These quality metrics essentially determine how effective the clustering algorithm is at reconstructing the grouping assigned by the original class labels. High values for F-measure and Adjusted Rand Index, and low values for Average Entropy, indicate a better clustering with respect to class labels. These measures are useful when a clustering is desired, which strongly correlates with underlying classifications.

Internal Metrics

Internal metrics refer to those that can be computed without class labels. Internal metrics are computed solely from the composition of a cluster, with no recourse to external information.

The internal metric we selected deals primarily with cluster divergence. Essentially, cluster divergence describes the difference in probability distributions for words between clusters. The internal divergence metric used here is based on the Kullback-Leibler (KL) divergence defined as follows, where c_1 and c_2 are clusters and

$V_{c_1c_2}$ is the vocabulary (i.e., set of nouns that occur in either c_1, c_2 or both).

$$KL(c_1, c_2) = \sum_{x_i \in V_{c_1c_2}} P(x_i|c_1) log \frac{P(x_i|c_1)}{P(x_i|c_2)} \tag{7.3}$$

For each cluster, we calculate self divergence by randomly splitting the cluster in half, and computing the KL divergence between the resulting two sub-clusters. This process is repeated 5 times, and the average is reported. The total self divergence of the clustering is then the sum of the averages for each cluster. When total self divergence is low, a clustering manifests the narrowness of its distribution for words, which indicates that documents with similar word distributions have been successfully grouped together. When total self divergence is high, a clustering shows that it contains clusters of documents with greatly differing word distributions, which may not be desirable.

7.3.2 Preliminary Experiments

A couple of preliminary experiments were necessary to fix the value of N in our top N approach to document representation, and to determine which of cosine and Euclidean distance should serve as the baseline to compare against our semantic distance. In both cases, the four cluster quality metrics were used.

To compare cosine and Euclidean distances, we ran the k-means algorithm with each one for $3 \leq k \leq 20$. The results showed that both metrics performed comparably, with the cosine distance metric slightly outperforming the Euclidean distance. We therefore use the cosine distance metric when comparing against our semantic k-means algorithm.

To determine a reasonable value for N, we ran the k-means algorithm with both the cosine and the Hausdorff distances for $1 \leq k \leq 20$, and varied N from 1 to 30. The results suggested that a value of $N = 19$ was optimal.

7.3.3 Semantic vs. Cosine k-Means

Using the best runs from k-means with a cosine distance metric, and the best runs from our semantic k-means algorithm, we compare their performance over the four cluster quality metrics for values of k ranging from 3 to 20. The results are shown in Figures 7.2-7.5.

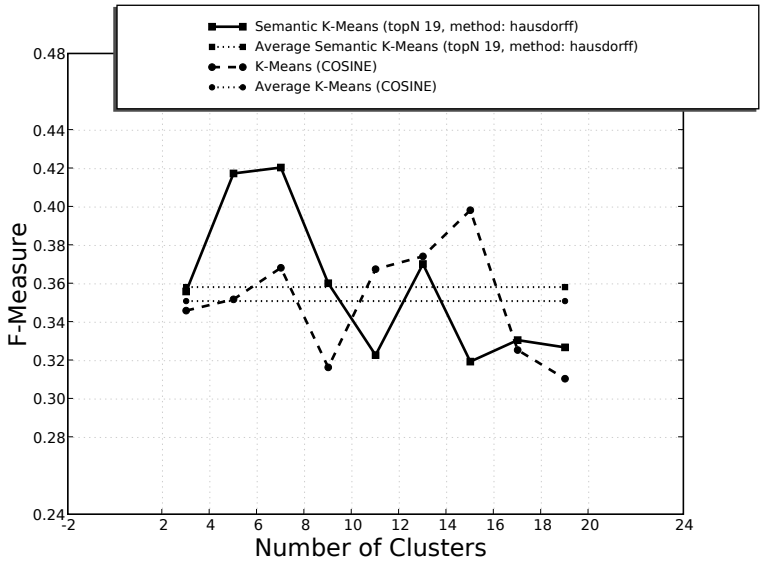

Figure 7.2: Semantic vs. Cosine: F-measure.

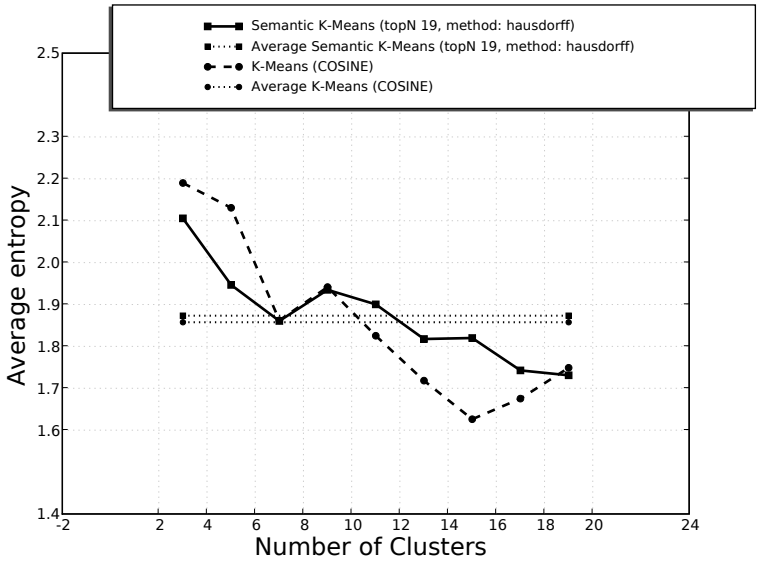

Figure 7.3: Semantic vs. Cosine: Entropy.

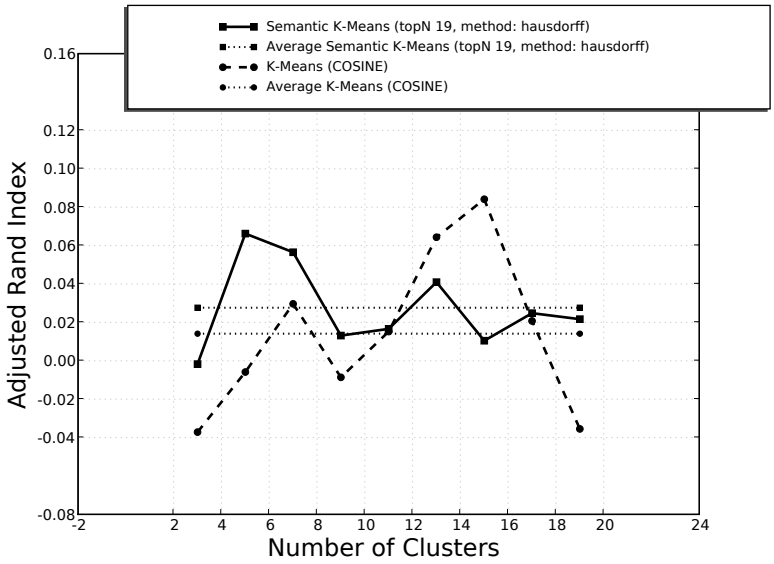

Figure 7.4: Semantic vs. Cosine: Adjusted Rand Index.

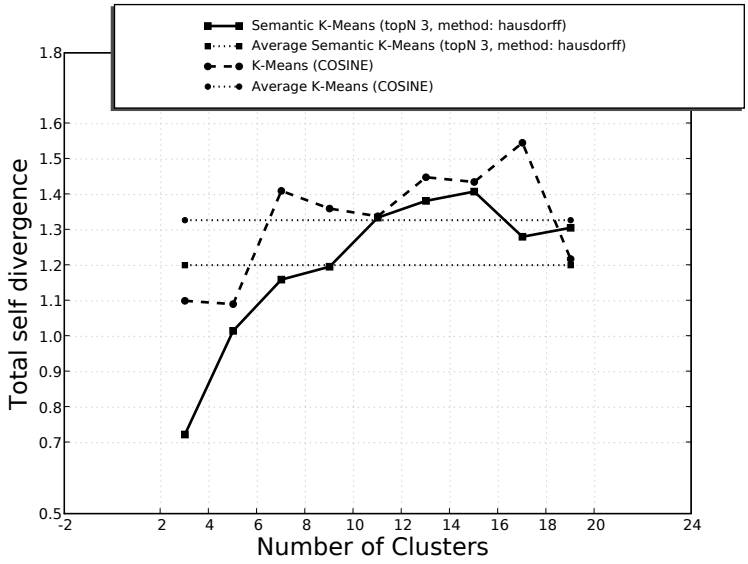

Figure 7.5: Semantic vs. Cosine: Self Divergence.

The following conclusions in favor of semantic k-means can be drawn from these graphs.

- On average, cluster quality is marginally superior when the semantic distance is used.

- External cluster quality, as measured by self-divergence, is consistently superior when the semantic distance is used, up to $k = 18$.
- For internal cluster quality, the curves appear to cross twice, with semantic distance giving rise to higher quality for $k \leq 10$ and $k \geq 18$, and lower quality between these values.
- Interestingly, for $k = 5$, which corresponds to the number of underlying classes, the semantic distance gives clusters of significantly superior quality, especially in terms of internal quality. This could indicate that the underlying class labels represent semantic topics of some description.

7.3.4 Aligned Word Clusters

In addition to the clustering that we performed using our Hausdorff document-level semantic distance metric, we also obtained significant results using our document-level distance metric that utilizes aligned word clusters. We use the aligned word clusters metric in an attempt to capture similarity between documents that contained multiple topics. As this metric functions in the same way as our Hausdorff distance metric, we can trivially substitute it into our semantic distance k-means algorithm.

Using $N = 3$, we re-run the comparison between the best runs from k-means with a cosine distance metric and the best runs from our extended (multiple topics) semantic distance k-means algorithm. Again, we compare performance over the four cluster quality metrics for values of k ranging from 3 to 20. The results are shown in Figures 7.6-7.9.

Conclusions similar to the ones discussed when using the standard Hausdorff distance can be drawn here about the improved cluster quality obtained with word cluster alignment.

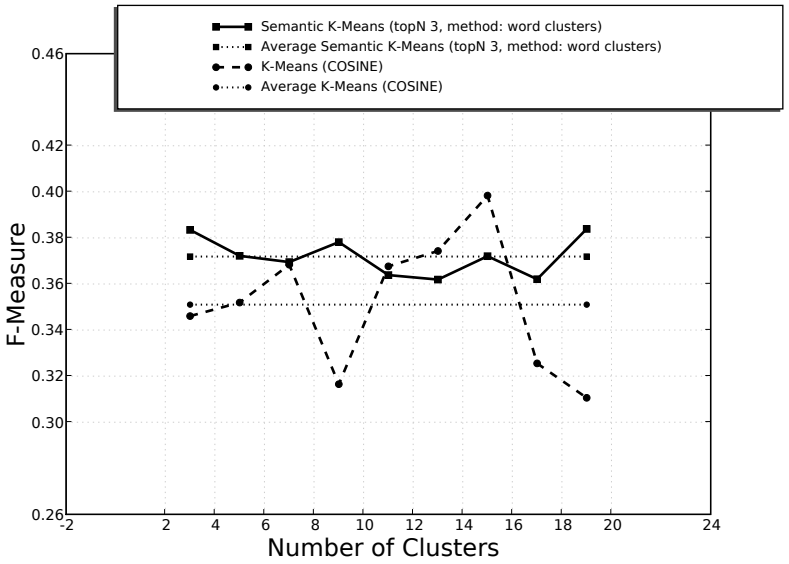

Figure 7.6: Multiple Semantic vs. Cosine: F-measure.

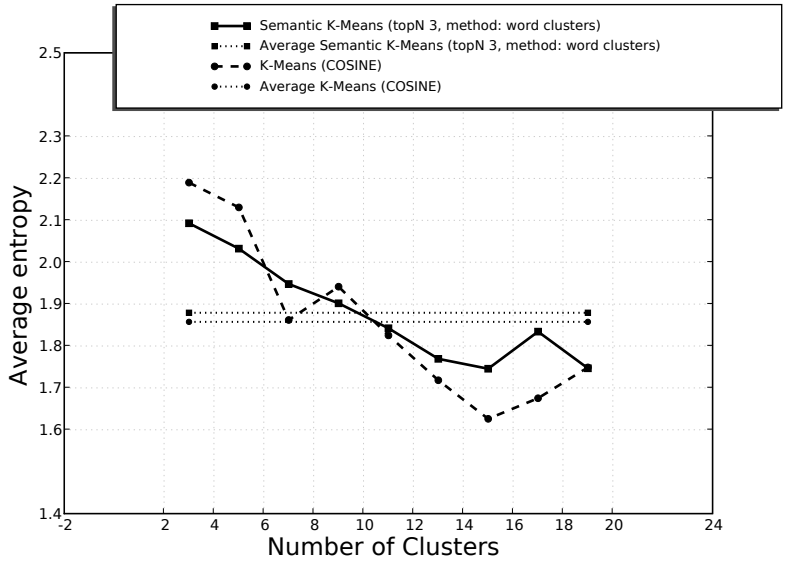

Figure 7.7: Multiple Semantic vs. Cosine: Entropy.

Comparison with Jiang-Conrath Metric

In a further experiment, we tested the fitness of our foundational word-level distance metric as the basis for our document-level distance metric, by comparing it with the same document-level distance metric that uses a different word-level semantic distance. We do this by substituting a word-level semantic distance metric defined by

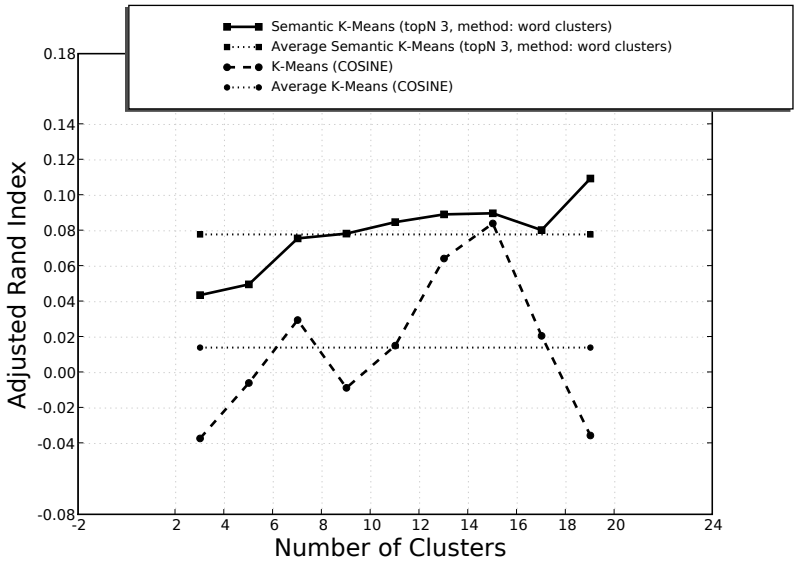

Figure 7.8: Multiple Semantic vs. Cosine: Adjusted Rand Index.

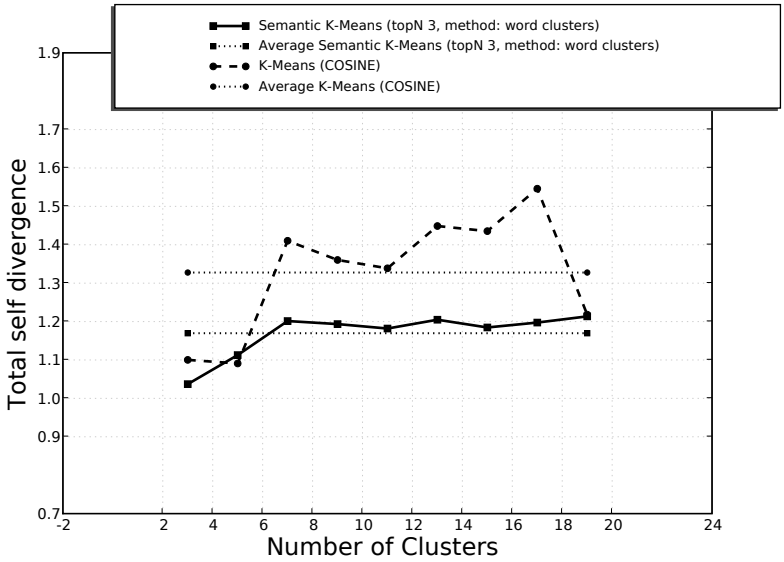

Figure 7.9: Multiple Semantic vs. Cosine: Self Divergence.

Jiang and Conrath, which also relies on WordNet as a knowledge structure [23]. We compare clusterings on the same Reuters-21578 data set as in previous experiments, and with the Hausdorff document-level semantic distance metric. The difference is the substitution of our word-level distance metric with the Jiang-Conrath word-level semantic distance metric, as our new basis for comparison. Figures 7.10-7.13 show

that our word-level distance metric produces generally better clusterings than the Jiang-Conrath approach, according to our stated evaluation metrics. The metric for which we do not excel is total self divergence, and after a brief investigation, it is unclear as to why this is the case. In many of our other experiments, success with respect to this metric was correlated to success with the other metrics we observe, however in our comparison with the Jiang-Conrath distance, we do not find this to be the case.

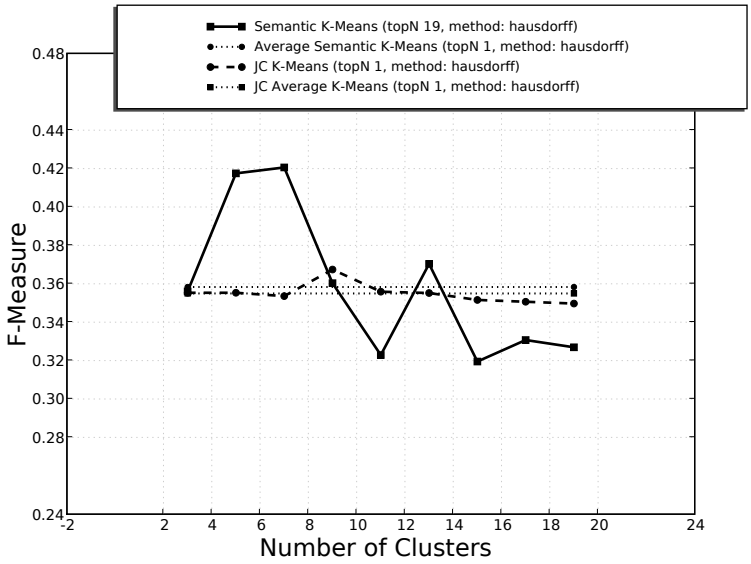

Figure 7.10: Semantic vs. Jiang-Conrath: F-measure.

7.4 Semantic Hierarchical Agglomerative Clustering

Another very common document clustering algorithm is the hierarchical agglomerative clustering algorithm. This algorithm is also distance based, and works as described in Figure 7.14. This approach presents a "bottom-up" formation of clusters that result in a cluster hierarchy.

We use our Hausdorff document-level semantic distance metric in step 2, to yield a semantically enabled HAC algorithm. Specifically, we denote cluster distance according to both the following approaches:

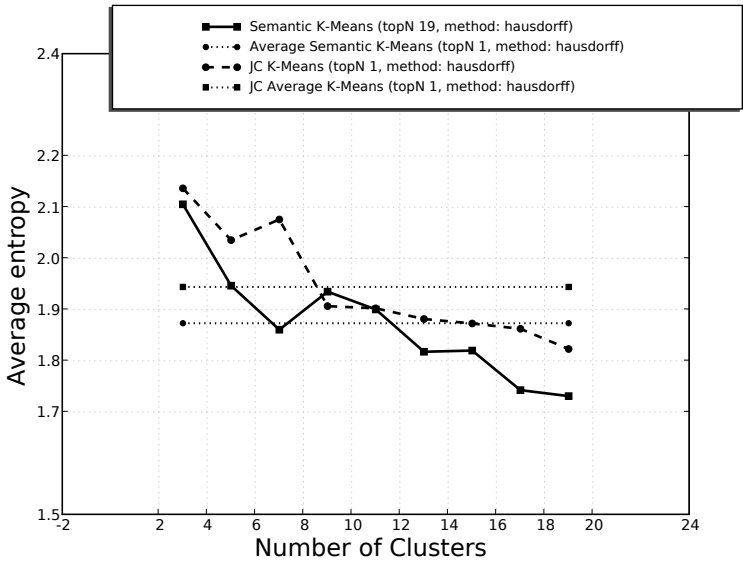

Figure 7.11: Semantic vs. Jiang-Conrath: Entropy.

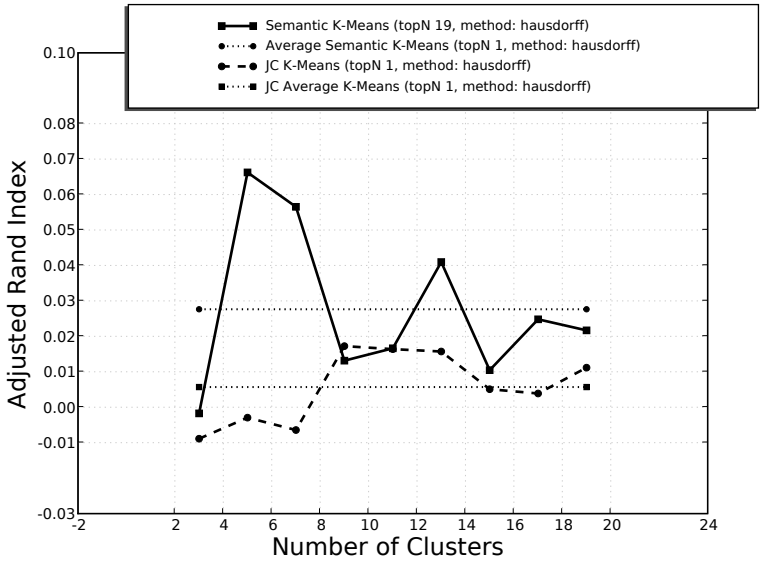

Figure 7.12: Semantic vs. Jiang-Conrath: Adjusted Rand Index.

- **Complete Link** - The maximum distance between the documents in clusters. Formally, for clusters A and B, we use:

$$max\{distance(a, b) : a \in A, b \in B\}$$

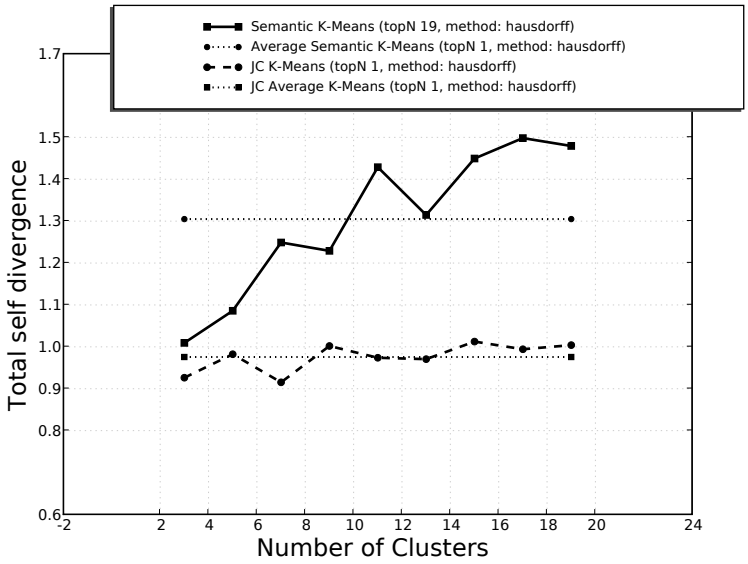

Figure 7.13: Semantic vs. Jiang-Conrath: Self Divergence.

1. *Put each document in its own cluster*
2. *Combine the closest two clusters (using some distance metric)*
3. *Repeat step 2 until all elements are in a single cluster*

Figure 7.14: The HAC algorithm.

- **Single Link** - The minimum distance between the documents in clusters. Formally, for clusters A and B, we use:

$$min\{distance(a, b) : a \in A, b \in B\}$$

7.5 Experimental Results (HAC)

We report on experiments that compare the use of cosine and Euclidean distances versus our semantic distance in the HAC algorithm. Our test data in this experiment consists of the same reduced version of the Reuters-21578 that we used in our k-means experiment. Our results consist of a comparison of the same cluster quality metrics across different numbers of clusters. We examine both cosine and Euclidean

distances, and both single and complete link clustering to increase the coverage of our comparison with our semantic distance.

7.5.1 Complete Link

Figures 7.15 through 7.18 show results using complete link clustering. The use of our semantic distance in this experiment yielded generally better results as we were able to obtain better scores on three of our four observed metrics. It is unclear why we do not also outperform cosine and Euclidean distances with respect to Entropy, however our results for this metric, although slightly worse, are very comparable.

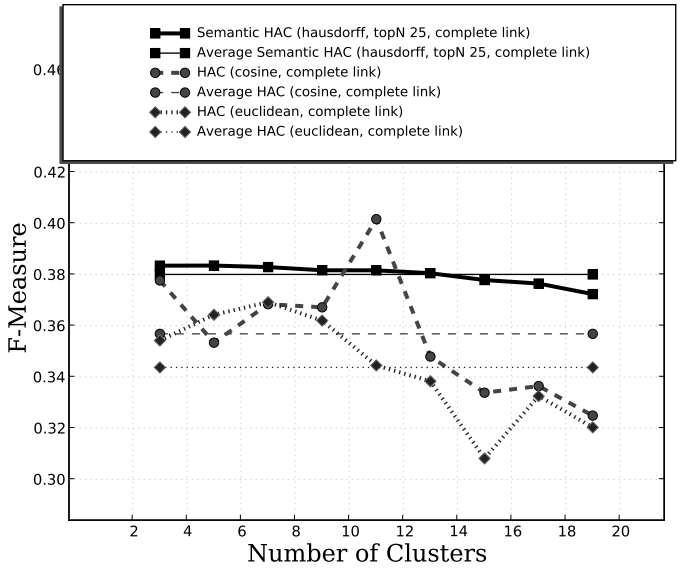

Figure 7.15: F-measure (Complete Link HAC)

7.5.2 Single Link

Figures 7.19 through 7.22 show results using single link clustering. Here again we were able to produce generally better results.

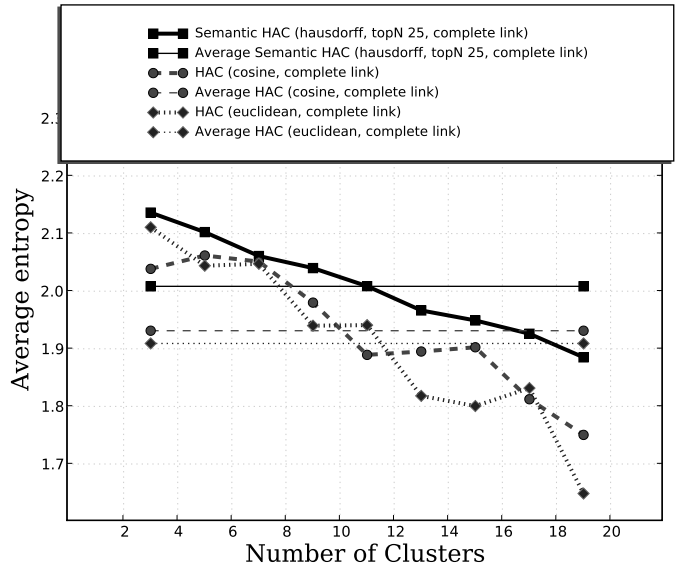

Figure 7.16: Entropy (Complete Link HAC)

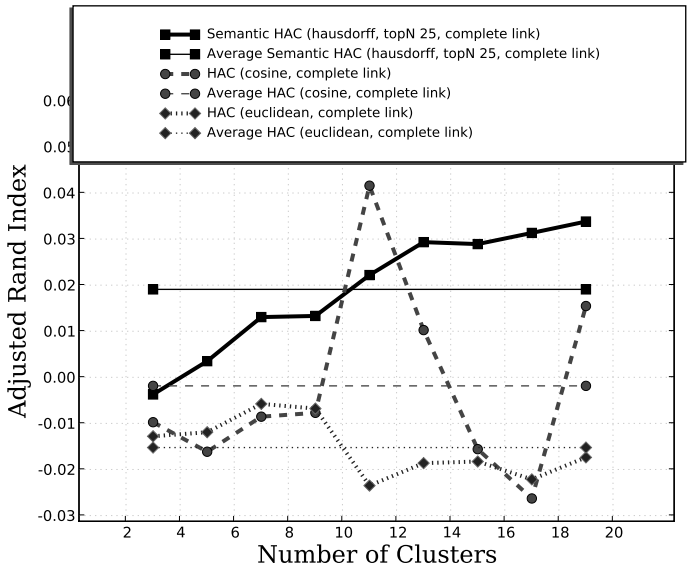

Figure 7.17: Adjusted Rand Index (Complete Link HAC)

7.6 Discussion

We have shown that using our proposed document-level semantic distance to cluster 200 documents from the Reuters-21578 data set, we are able to improve cluster quality as measured by F-measure, average entropy, adjusted rand index and total

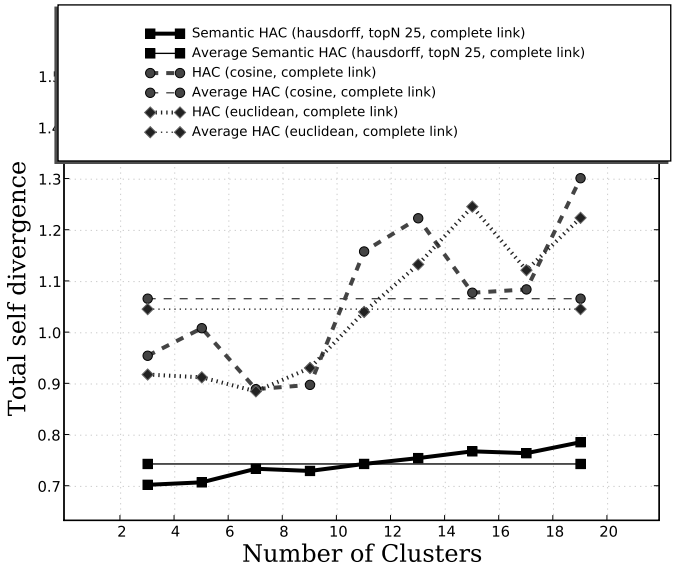

Figure 7.18: Total Self Divergence (Complete Link HAC)

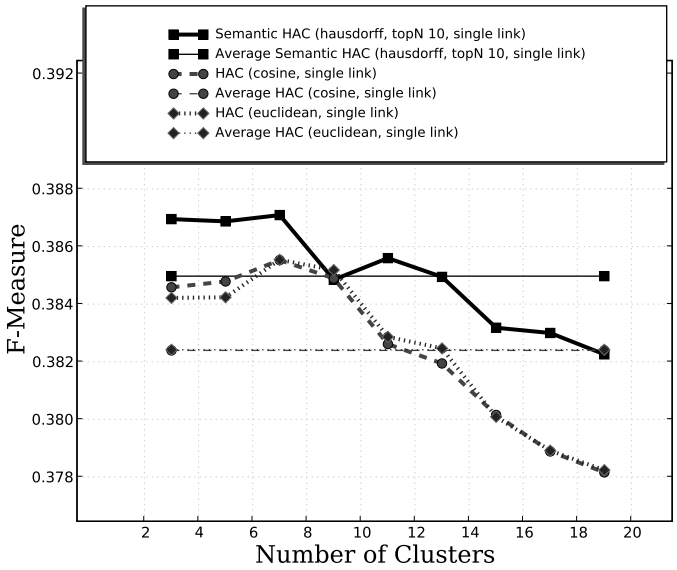

Figure 7.19: F-measure (Single Link HAC)

self divergence, over cosine and Euclidean distances when using either the k-means clustering algorithm, or hierarchical agglomerative clustering.

We have also shown how we can extend the measure to account for documents with multiple semantic topics and showed again that a similar gain may be obtained in terms of cluster quality. Given the ability to appropriately choose parameters, we

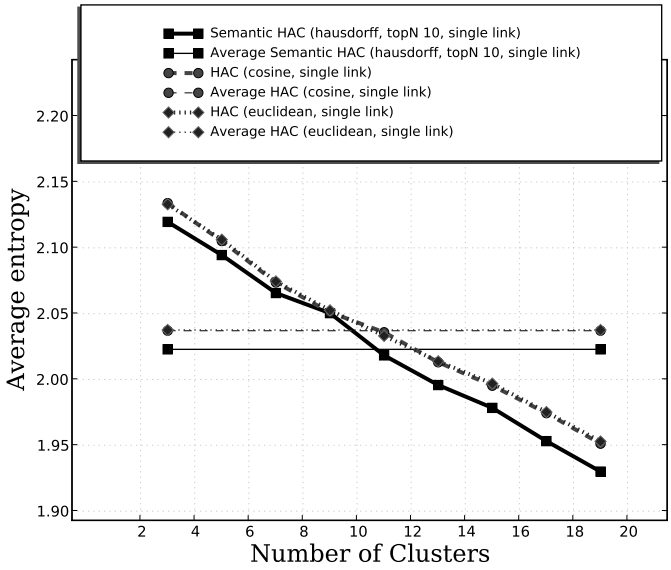

Figure 7.20: Entropy (Single Link HAC)

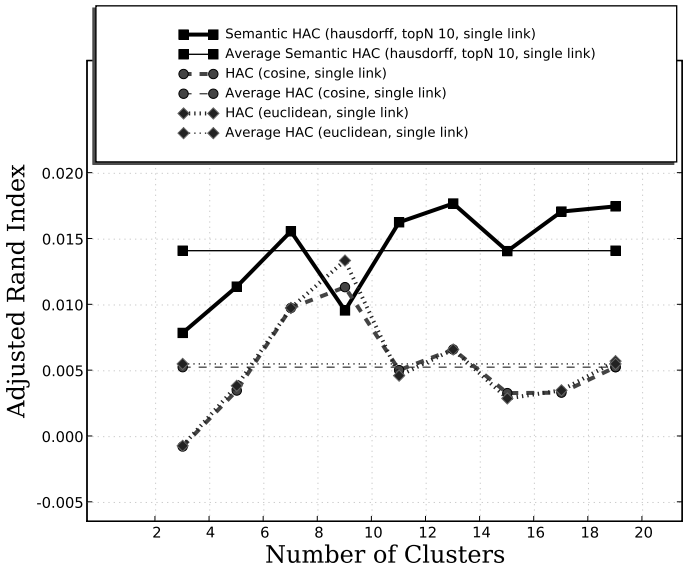

Figure 7.21: Adjusted Rand Index (Single Link HAC)

conclude that the semantic Hausdorff distance outperforms, overall, both cosine distance and Euclidean distance, when used in the k-means or hierarchical agglomerative clustering algorithms, on the data set of our examination.

We further propose that our semantic distance will generalize, in distance based clustering algorithms, to other collections of natural language text, provided

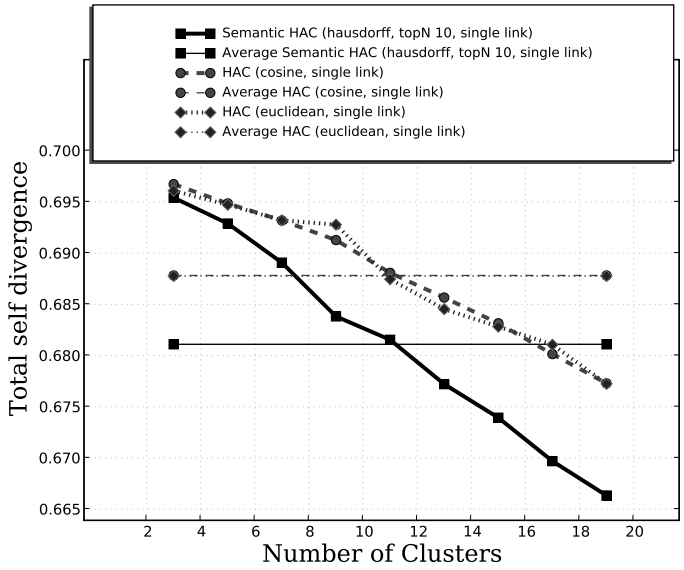

Figure 7.22: Total Self Divergence (Single Link HAC)

that the knowledge base structure used to determine word-level semantic distances accurately describes semantic relationships of words in the text.

Chapter 8

Conclusion and Future Work

8.1 Summary of Contributions

The most significant contribution of this thesis is our design, implementation, and use of a novel, effective, and theoretically sound document-level semantic distance metric. This distance metric makes possible document-level quantitative analysis for normally qualitative word semantics. We built our document-level distance metric from our implementation of an effective word-level distance metric, and successfully showed how it could be used to improve results for real world problems related to document retrieval and clustering.

Additionally, we define a secondary method for expanding user keyword queries using Latent Dirichlet Allocation (LDA). While this method differs in its approach from our knowledge-based, semantic distance approach, it is nonetheless a significant contribution as it was found to be both novel (to the best of our knowledge) and highly effective in improving precision and recall.

8.2 Challenges and Achievements

One significant challenge was that of identifying and using the correct semantic relationships for the problems that we addressed. For example, as mentioned in the case of search indexing, the use of hypernymy/hyponymy resulted in associating terms with documents that were not determined to be relevant according to our Aquaint

data set, but that were identified as semantically similar. This was one example, of many, that could be cited where relationships of hypernymy were clearly evident, but not extremely useful in identifying topical similarity or term relevance in the expected sense. Another challenge of our approach is its execution runtime requirements. While we were able to identify several improving techniques with respect to runtime, our approach is still slower than alternative approaches, such as many corpus based methods.

Despite these challenges, however, we were able to make significant achievements. Using our document-level semantic distance metric, we were able to obtain improved information retrieval measures for both precision and recall by successfully 1) re-ranking search results, and 2) by expanding keyword queries. We were also able to improve scores for F-measure, average entropy, adjusted rand index, and total self divergence when applying our document-level semantic distance metric to document clustering. These improvements were had when examining large, real world problems and data sets.

8.3 Future Work

Interesting research related to our semantic distance metric remains for future work. Specific items of future work include:

- Utilizing semantic distance for text classification.
- Applying similar experiments to those described in this thesis, to data sets where hypernymy/hyponymy are more evident, such as query-by-example data sets. One could imagine a keyword query system that allowed users to search for images by providing keywords that describe the concepts in the image. A search for "buildings", or similar example type concepts, may be a setting where our hypernymy/hyponymy WordNet semantic distance would be very well suited.

- Exploring alternate knowledge structures, such as the Wikipedia category hierarchy, as an alternative to WordNet.
- Forming clusters, using our distance metric, for the purpose of initializing probability distributions for words and clusters that can be used to seed the Expectation Maximization clustering method.

Bibliography

[1] The longman defining vocabulary, March 2007. `http://home.earthlink.net/ neilbawd/longman.txt`.

[2] M. Baziz, M. Boughanem, and N. Aussenac-Gilles. A conceptual indexing approach for the trec robust task. In *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, 2005.

[3] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. In *Journal of machine Learning Research 3*, 2003.

[4] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] F. Blei, M. Ester, and X. Xu. Frequent term-based text clustering. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, pages 436–442, 2002.

[6] T. Bogers and A. van den Bosch. Authoritative re-ranking of search results. *Advances in Information Retrieval*, 3936/2006:519–522, 2006.

[7] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic distance. *Computational Linguistics*, 32(1):13–47, 2006.

[8] S. Bttcher, C.L.A. Clarke, and P.C.K.Yeung. Index pruning and result reranking: Effects on ad-hoc retrieval and named page finding. In *The Fifteenth Text REtrieval Conference (TREC 2006) Notebook*, page 237, 2006.

[9] B. Choudhary and P. Bhattacharyya. Text clustering using semantics. In *Proceedings of the 11th International World Wide Web Conference*, 2002.

[10] D.R. Cutting, D.R. Karger, J.O. Pedersen, and J.W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th International Conference on Research and Development in Information Retrieval*, pages 318–329, 1992.

[11] Hal Daumé III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, Sydney, Australia, 2006.

[12] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 1990.

[13] B. Everitt. *Cluster Analysis*. John Wiley & Sons, Inc., 1993.

[14] D. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[15] B. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent itemsets. In *Proceedings of the SIAM International Conference on Data Mining*, pages 59–70, 2003.

[16] J.H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40:11–61, 1989.

[17] P.V. Henstock, D.J. Pack, Y.-S. Lee, and C.J. Weinstein. Toward an improved concept-based information retrieval system. In *Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval*, pages 384–385, 2001.

[18] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, pages 50–54, 1999.

[19] A. Hotho, S. Staab, and A. Maedche. Ontology-based text clustering. In *Proceedings of the IJCAI Workshop on Text Learning: Beyond Supervision*, 2001.

[20] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[21] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., 1988.

[22] D. Jensen, C. Giraud-Carrier, and N. Davis. A method for computing lexical semantic distance using linear functionals. In *Journal of Web Semantics*, 2007. DOI:http://dx.doi.org/10.1016/j.websem.2007.11.001.

[23] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy, 1997.

[24] S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 2:241–254, 1967.

[25] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Inc., 1990.

[26] S.-B. Kim, H.-C. Seo, and H.-C. Rim. Information retrieval using word senses: Root sense tagging approach. In *Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval*, pages 258–265, 2004.

[27] J. Kogan, M. Teboulle, and C. Nicholas. The entropic geometric means algorithm: An approach to building small clusters for large text datasets. In *Proceedings of the ICDM Workshop on Clustering Large Data Sets*, pages 63–71, 2003.

[28] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[29] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[30] K. Lang. News weeder: Learning to filter netnews. In *Proceedings of the 12th International Conference of Machine Learning*, pages 331–339, 1995.

[31] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1999.

[32] L. Lebart and M. Rajman. *Handbook of Natural Language Processing Computing Similarity*. Marcel Dekker, Inc., 2000.

[33] A. Leouski and W. Croft. An evaluation of techniques for clustering search results. Technical Report IR-76, Department of Computer Science, University of Massachusetts, Amherst, 1996.

[34] K. Lerman. Document clustering in reduced dimension vector space. Unpublished, 1999.

[35] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[36] R.F. Mihalcea and S.I. Mihalcea. Word semantics for information retrieval: Moving one step closer to the semantic web. In *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence*, pages 280–287, 2001.

[37] G.A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[38] Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *Research and Development in Information Retrieval*, pages 206–214, 1998.

[39] The National Institute of Standards and Technology (NIST) and U.S. Department of Defense, `http://trec.nist.gov/`. *The Text REtrieval Conference (TREC)*.

[40] A. Natsev, A. Haubold, J. Tešić, L. Xie, and R. Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 991–1000, New York, NY, USA, 2007. ACM.

[41] Wordnet Senses Paolo. Text categorization and information retrieval using.

[42] V. Prince and M. Lafourcade. Mixing semantic networks and conceptual vectors: The case of hyperonymy. *In Proceedings of the 2nd IEEE International Conference on Cognitive Informatics*, pages 121–128, 2003.

[43] Apache Lucene Project. Lucene — an open source information retrieval library, 2007. `http://lucene.apache.org/java/docs/index.html`.

[44] Yonggang Qiu and Hans-Peter Frei. Concept-based query expansion. In *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 160–169, Pittsburgh, US, 1993.

[45] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989.

[46] P. Resnik. *Selection and Information: A Class-based Approach to Lexical Relationships*. PhD thesis, University of Pennsylvania, 1993.

[47] 1 Rohini and 1 Varma. A novel approach for re-ranking of search results using collaborative filtering. In *International Conference on Computing: Theory and Applications (ICCTA'07)*, pages 491–496, 2007.

[48] P. Rosso, E. Ferretti, D. Jiménez, and V. Vidal. Text categorization and information retrieval using wordnet senses. In *Proceedings of the 2nd International Conference of the Global WordNet Association*, pages 299–304, 2004.

[49] H. Rubenstein and J. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627633, 1965.

[50] D.E. Rumelhart and D. Zipser. Feature discovery by competitive learning. *Cognitive Science*, 9:75–112, 1985.

[51] H.M.T. Saarikoski. 2t: two-term indexing of documents using syntactic and semantic constraints. In *Proceedings. Sixteenth International Workshop on Database and Expert Systems Applications*, pages 1025–1028, 2005.

[52] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[53] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[54] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd International Conference on Research and Development in Information Retrieval*, pages 208–215, 2000.

[55] W. Song and S.C. Park. A novel document clustering model based on latent semantic analysis. In *Proceedings of the 3rd International Conference on Semantics, Knowledge and Grid*, pages 539–542, 2007.

[56] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *Proceedings of the KDD Workshop on Text Mining*, 2000.

[57] A. Termier, M-C. Rousset, and M. Sebag. Combining statistics and semantics for word and document clustering. In *Proceedings of the IJCAI Workshop on Ontology Learning*, 2001.

[58] Y. Tsuruoka and J. Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *HLT/EMNLP*, pages 467–474, 2005. `http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/postagger/`.

[59] E. M. Voorhees. The trec robust retrieval track. *SIGIR Forum*, 39(1):11–20, 2005.

[60] Y. Wang and J. Hodges. Document clustering with semantic analysis. In *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006.

[61] Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA, 2006. ACM.

[62] M. Wen and X. Huang. A multilevel searching and re-ranking framework for information retrieval. In *2006 IEEE International Conference on Granular Computing*, pages 619–622, 2006.

[63] P' Willett. Recent trends in hierarchic document clustering: A critical review. *Information Processing and Management*, 24(5):577–597, 1988.

[64] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.

[65] O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st International Conference on Research and Development in Information Retrieval*, pages 46–54, 1998.

[66] J. Zhang, Z. Ghahramani, and Y. Yang. A probabilistic model for online document clustering with application to novelty detection. In *Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, 2004.

[67] Z. Zhuang and S. Cucerzan. Re-ranking search results using query logs. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 860–861, New York, NY, USA, 2006. ACM.

[68] Justin Zobel and Alistair Moffat. Inverted files for text search engines. *ACM Comput. Surv.*, 38(2):6, 2006.