Fall 2012

Semantic preserving text tepresentation and its applications in text clustering

Michael Howard

SEMANTIC PRESERVING TEXT REPRESENTATION AND ITS APPLICATIONS IN TEXT CLUSTERING

by

MICHAEL HOWARD

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER SCIENCE

2012

Approved by

Dr. Wei Jiang, Advisor
Dr. Jennifer Leopold
Dr. Donald Wunsch

ABSTRACT

Text mining using the vector space representation has proven to be an valuable tool for classification, prediction, information retrieval and extraction. The nature of text data presents several issues to these tasks, including large dimension and the existence of special polysemous and synonymous words. A variety of techniques have been devised to overcome these shortcomings, including feature selection and word sense disambiguation. Privacy preserving data mining is also an area of emerging interest. Existing techniques for privacy preserving data mining require the use of secure computation protocols, which often incur a greatly increased computational cost. In this paper, a generalization-based method is presented for creating a semantic-preserving vector space which reduces dimension as well as addresses problems with special word types. The SPVSM also allows private text data to be safely represented without degrading cluster accuracy or performance. Further, the result produced is also usable in combination with theoretic based techniques such as latent semantic indexing. The performance of text clustering using the semantic preserving generalization method is evaluated and compared to existing feature selection techniques, and shown to have significant merit from a clustering perspective.

ACKNOWLEDGMENTS

For the advice and patience of the thesis committee: Dr. Wei Jiang, Dr. Jennifer Leopold, and Dr. Donald Wunsch. Faculty members of the Missouri University Of Science And Technology and University Of Central Missouri: Dr. Maggie Cheng, Dr. Chris Merz and Dr. Daniel Tauritz; Dr. Hang Chen, Dr. David Ewing, Dr. Rhonda McKee, Dr. Songlin Tian and Dr. Lianwen Wang. For their fundamental achievements to mathematics, computer science and theory pertaining to this thesis: Archimedes, Alonzo Church, Edsger Dijkstra, Euclid, Leonhard Euler, Carl Friedrich Gauss, Kurt Gödel, Dr. Donald Knuth, Andrey Markov, John von Neumann, Isaac Newton, Charles Émile Picard, Pythagoras, and Alan Turing. Additionally, for their advice, support and tutelage: Ravi Arvapally, Shane Blazek, Blake DeBray, Yousef Elmehdwi, Chin Huang-Stalick, Christopher Hurst, Prakash Kumar, Nian Liu, Brian Pattiz, Bharath Samanthula, Francis Schmitz, John Stephens, Usha Sree Vellala, Quanmin Ye, and the Graduate Studies Office.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

1. INTRODUCTION

Text mining [1] is an area which has experienced rapid growth with the availability of large stores of data, such as archives of books and newspapers, web sites, informal communications such as newsgroups and Internet forums, academic journals, corporate documentation and so on. Many important tasks can be accomplished by organizing and extracting information from text data. One such task is text document clustering, which seeks to identify similarities between documents and place them into groups based on their similarity. A second area of emerging interest is privacy preserving data mining, which adds the requirement that the privacy of information is not compromised in the process of producing a result or model constructed jointly by multiple parties.

Difficulties in document clustering arise from the nature of text data, as well as its applications. When represented in the vector space model[2], text data tends result in a sparse matrix of enormous dimension. The *curse of dimensionality* [3] can force otherwise effective computations to become cumbersome and unworkable, as even a small number of documents could potentially contain tens of thousands of distinct terms. Additional difficulties are created by single words which contain multiple senses (polysemy) and separate words which share meanings (synonymy). Because similarity in the vector space model is based on term frequency, documents containing these types terms will be incorrectly calculated as similar or dissimilar to one another.

Feature selection techniques have been introduced to overcome problems posed by high dimensionality and by polysemous and synonymous words. The goal of feature selection is to identify the most prominent terms or sets of terms in a text collection. By creating a new vector space in which documents are represented by their most relevant terms, the size of the vector space can be reduced. Word sense disambiguation can be incorporated into feature selection to correctly identify polysemous words, and sources of background knowledge can be used to make the feature selection process aware of the shared meanings of synonymous words.

To address unwanted disclosure of sensitive information in a joint computation environment, secure multi-party computation techniques have been devised which allow the calculation of the without sharing private information. These techniques rely on computationally intensive cryptographic primitives and protocols. in [4], a secure protocol is proposed for determining the cosine similarity between two text documents. The experimental section showed that to calculate the similarity between two collections of 1000 documents would require six days.

Furthermore, text data requires that parties agree beforehand on a set of terms to include in the vector space. One solution is to securely compute the set intersection of the term space of both parties. Existing feature transformation techniques based on matrix decomposition cannot be applied here because the resulting features are solely dependent on each parties text collection and cannot be compared. Even after creating the joint vector space, computing clusters using secure multiparty computation has a much higher computational complexity than conventional clustering. The dimensionality problem of text data is exacerbated in the setting of secure multi-party computation. The complexity of secure multi-party computation is usually given in terms of the required number of encryptions, as these are the most costly operations. If the number of communications between parties is based on the dimension of the data, then the typically large feature space found in text collections becomes even more problematic. These issues provide the motivation to represent text collections in a reduced-dimension vector space which protects sensitive information while allowing the relationship between similar documents to be detected.

In this work, a method is presented in which documents words are replaced by related, but less specific terms. The process of correctly identifying words suitable as replacements is guided by a word ontology, which is a collection of words organized in a tree structure by parent-child relationships. Each child word is semantically similar to its parent, but with a more specific definition. At the root of an ontology are general notions, topics and categories, while the leaves of the tree are highly specific concepts. For example, an ontology with the word *food* at the root might see the word *fruit* as a parent of *orange*; the word *orange* as a parent of *mandarin orange*, and so on. Some well known ontologies include Wordnet [5] and Mesh [6], which have already been successfully incorporated in a number of publications on feature selection and word-sense disambiguation. This work aims to use knowledge from an ontology in a way that directly reduces vector space dimension and resolves synonymous words.

The process of replacing words from a document in this manner is called *semantic preserving text generalization*, and the result is a *semantic preserving vector space model* The goal of privacy preservation is achieved because sensitive information is removed gradually from private text collections. Rather being deleted outright from the vector space, highly sensitive terms are instead be replaced by more palatable words. Furthermore, generalization provides a fast and simple way of performing feature selection. The method exhibits the same characteristics of other term selection techniques in that it reduces the feature space without significantly degrading clustering performance, and it is also shown to resolve the problems of synonymous and polysemous words. Furthermore, the SPVSM is able to directly capture the close relationship between sibling words, which enhances the ability of clustering algorithms to organize text collections by topic. The evaluation section shows that the method often increases cluster performance with very little added computation time.

Section 2 presents definitions and related work. Section 3 introduces the SPVSM and its foundational ideas. Sections 3.6, 4 and 3.7 illustrates the feasibility and applicability of the method in text clustering. Section 5 concludes the thesis with some future research directions.

2. RELATED WORK

This section details some of the related work in text clustering, feature selection and word sense disambiguation.

2.1. TEXT CLUSTERING

Cluster analysis is a modeling technique which assigns data observations into groups based on their similarity [7]. Because it requires little to no prior knowledge to perform, it is especially useful for extracting knowledge from unlabeled data. Clustering has performed on text data including documents, words, and word senses using a variety of techniques. Particularly popular are iterative approaches to partitional clustering such as k-means and its popular variants, spherical k-means [8] and bisecting k-means [9]. Using a vector space model, spherical k-means [8] introduces 'concept vectors' for each cluster, which are the the cluster centroids normalized to have a unit vector length, while bisecting k-means [9] begins with one large cluster and repeatedly divides into halves until the desired number of cluster centers are reached. A review by Steinbach [9] claimed that for text data, partitional clustering algorithms like spherical and bisecting k-means are generally more effective than other methods. Another partitional approach is Cutting's [10] Scatter/Gather method, a seed-based algorithm which selects cluster centers randomly, and chooses a small sample of observations to which partitional clustering is applied. [11] clusters text documents using a self-organizing map method. This approach was successfully adapted to text data in Larsen's work [12]. [13] clusters documents using a graph-theory based bipartite matching.

2.2. FEATURE SELECTION

The nature of text data often presents challenges because of high dimension and ambiguous/overlapping word senses. A variety of feature selection methods have been created to address these issues. Parsons' [14] survey about high dimensional clustering techniques makes a distinction between feature selection, which decides on which existing attributes to keep in a model, and feature transformation, which create new attributes using mathematical techniques or aggregation. In the first category, a number of techniques have been devised which use ontologies as external knowledge sources. Most similar to the generalization method presented in this paper are Hotho's [15] three strategies for including Wordnet background knowledge in the vector space model. Hotho's most successful strategy is to replace terms with their related concepts, thus creating *concept vectors* to represent documents. A study [11] comparing the Hotho's concept-vector method against representation through n-grams indicated that Hotho's method resulted in better document classification. Fodeh et al [16] achieves a dramatic vector space reduction by including only highly frequent polysemous and synonymous nouns that exceed a information-gain based threshold after disambiguation, but can also yield a situation where some documents are not represented in the new space at all. Recupero [17] builds new feature vectors by examining the relationship between the most frequent words in a text collection and their distance in an ontology. [18] defined 'Wordnet-enabled k-means' to cluster user preferences for web browsing. The second category consists of matrix decomposition based techniques can reduce potentially thousands of dimensions into a few hundred. Latent semantic indexing [19]uses a matrix decomposition to approximate high dimension data in a new set of axes in a manner which retains the relative distances of the data in its original dimension. Hofmann's [20] probabilistic latent semantic indexing uses expectation maximization to improve the technique and to help overcome problems with polysemous and synonymous words.

2.2.1. Word Sense Disambiguation. Word sense disambiguation is the task of choosing the the appropriate meaning a word based on its usage. WSD can provide gains in information retrieval performance, either directly or through its inclusion in feature selection techniques. Of particular relevance here are the techniques which leverage a source of external knowledge such as Wordnet to perform disambiguation. Fodeh employed an ensemble based approach [21], which also utilizes the Wordnet dictionary. Rather than using the bag-of-words approach to document representation, Hung [22] created extended significance vectors from the gloss definitions of Wordnet concepts instead of using concepts directly and clustered documents using self-organizing maps. Agirre defined a conceptual distance measure with which to disambiguate words. [23] uses the apriori principle to identify frequent Wordnet concepts in documents in conjunction with a hierarchical clustering algorithm and obtained cluster performance comparable to the popular bisecting k-means algorithm. Wang[24] proposed using Wikipedia to disambiguate word senses by measuring the cosine similarity between sentences containing ambiguous words and Wikipedia articles describing polysemous concepts.

2.2.2. Privacy Preserving Computation And Data Anonymization. Specifically relating to text data, Jiang [4] demonstrates how parties can securely compute cosine similarity between individual document vectors, and how the protocol can be extended to entire collections. Some existing efforts relating to privacy preserving clustering include [25], [26] and [27]. Though not specific to text data, each presents partitional clustering achieved by secure multiparty computation protocols. Vaidya's [25] protocol assumes a vertically partitioned data model and describes a k-means clustering method in the distance of observations to the cluster centroids is calculated through Doganay [26] presents another approach based on vertically partitioned data and uses additive secret sharing to perform k means clustering. The method of [27] operates on horizontally partitioned data, and the cluster centers are created by using the concepts of random shares and homomorphic encryption. Another angle to privacy preserving computation is data sanitization, which attempts remove sensitive information from private datasets, thus allowing the data to be shared or published. Techniques in this vein related to text data include [28], [29] and [30]. [28] presents a technique for guaranteeing similarity between document vectors by replacing terms with dummy values. [29] identifies and removes sentences containing potentially sensitive information by using pre-defined contextual definitions. [30] uses an ontology to replace sensitive nouns with their related hypernyms, thus creating a level of uncertainty about the text of the original, unsanitized document. It is this approach provides which the inspiration for the method presented in this paper.

3. METHODOLOGY

This section first gives the formal definition of the vector space model and and similarity measures. An explanation of word ontologies is given, and it is shown how a word ontology is used to perform text generalization. In the next section, the definition of text generalization is stated and the merit of text generalization as a feature selection technique is demonstrated. Details are given about special cases of words which are problematic in the vector space model. It is shown how text generalization overcomes these issues, and a method for creating a semantic preserving vector space model is presented.

3.1. VECTOR SPACE TEXT REPRESENTATION

The vector space model is a common method for representing text data mathematically. Each document is conceptualized as a vector, and the components of each vector are the terms in the document collection. The frequency of words within the each document is counted and a frequency vector is created. Stated formally, suppose that $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ is a set of n documents, and $\mathcal{T} = t_1, t_2, \ldots, t_m$ is a set of terms where each document d contains a subset of $\mathcal{T}$. By counting the number of occurrences of each term in each document, a frequency vector for d is created: $\boldsymbol{d} =< f_{t_1,d_1}, f_{t_2,d_1}, \ldots, f_{t_m,d_1} >$. A *term-document frequency matrix* is created by considering the frequency of every term in every document:

$$\begin{array}{c} \\ \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \vdots \\ \boldsymbol{d}_n \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & \dots & t_m \end{array} \\ \begin{pmatrix} f_{t_1,d_1} & f_{t_2,d_1} & \cdots & f_{t_m,d_1} \\ f_{t_1,d_2} & f_{t_2,d_2} & \cdots & f_{t_m,d_2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{t_1,d_n} & f_{t_2,d_n} & \cdots & f_{t_m,d_n} \end{pmatrix} \end{array}$$

It is this matrix which is referred to as the vector space. Documents which are similar are presumed to share a significant number of the same terms with a high frequency. Thus, most similarity measures in the vector space model are based either on term frequency or weighted term frequency. To compare the similarity between two documents, the angle between their term-frequency vectors can be measured through cosine similarity:

$$sim(\boldsymbol{d}_i, \boldsymbol{d_j}) = \frac{\boldsymbol{d}_i \cdot \boldsymbol{d}_j}{\|\boldsymbol{d}_i\|\|\boldsymbol{d}_j\|}$$

For two related documents, the cosine similarity evaluates close to 1. When the cosine measure evaluated to 0, it indicates that the vectors are perpendicular, and thus unrelated and contain none of the same terms. Using the vector space model and cosine similarity allows for the use of many popular clustering algorithms. Note that cosine similarity is based entirely on term frequencies, and cannot account for semantic relationships between words.

3.2. SEMANTIC PRESERVING TEXT GENERALIZATION

An word ontology can be conceived as a tree structure of terms organized by a lexical relationship called hyponym. The essence of this relationship is that if a word w_1 describes a concept, then its hyponym w_2 is a elaboration of the same concept. At the root of each ontology is a broad category, and at level of the tree the nodes become semantically more narrow, with the leaves containing highly specialized terms. Though a word may have multiple senses, each sense is regarded as a distinct entry in the ontology. An example of a word ontology is shown in Figure 3.1.

A useful operation is to identify the nouns within a text document and replace them with parent words from an ontology. Because semantic meaning of each word becomes more general the closer it is to the root of the tree, this process is called *generalization*, and is the basis of the method presented here. The *level* or *height* of generalization refers to the distance between the original word and its replacement in the tree. That is, if a word is replaced by its immediate parent, then it has been generalized one level. If it is replaced by a word two nodes away in the ontology, it has been generalized two levels, and so on. Text generalization requires a large source of external knowledge, one source of which is available through a large computer dictionary called Wordnet [5].

Wordnet [5] is a database of several hundred thousand words created by a team of computer scientists and linguists at Princeton University (Miller95). Wordnet has the capabilities of both a dictionary and a thesaurus; word definitions are augmented by conceptual and lexical relationships between words. Each word is a member of a synyonym set which contains similar words. These synonym sets are sometimes referred to as concepts. Conceptual relationships between words are realized through a hierarchical tree structure. The most general words appear at the top of the tree, while specific terms appear at the bottom. Wordnet 3.0 contains 44 hierarchies, which are called lexical categories. Each lexical category represents a board class of words, such as . Wordnet contains other important information about words, such as the part of speech, a definition sentence called a *gloss*, as well as several other types of relationships between words. Thus Wordnet provides a mechanism to easily identify nouns, perform term stemming and stopword removal, and can to text generalization operations.

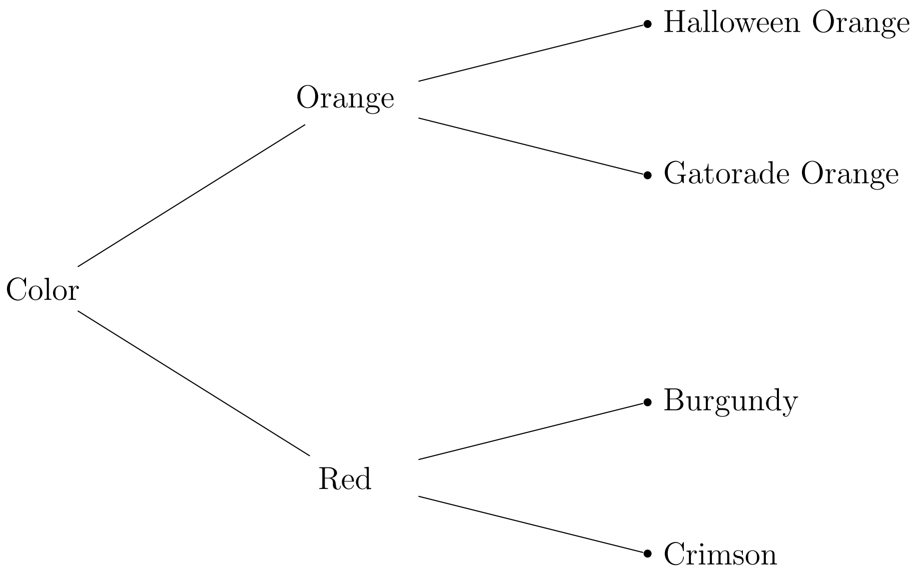

Figure 3.1: A sample ontology for colors

The next section shows how generalization can be used to remove private information from text collections.

3.3. TEXT GENERALIZATION AND PRIVACY PRESERVATION

The following sentence contains sensitive information about a patient's place of residence, symptoms, condition and treatment. If it is to be published, the information must be removed. Generalization allows sensitive information to be removed from a document, while preserving semantic information. The meaning and topic of the sentences are retained, while the private information is removed:

A **Sacramento** resident purchased **marijuana** to relieve **lumbar pain** caused by **liver cancer**.

A **State Capital** resident purchased **drug** to relieve **pain** caused by **carcinoma**.

In the method presented by this thesis, entire documents containing private or sensitive information are generalized uniformly, meaning that all nouns are identified and replaced by This allows text clustering and calculation of document similarity without the use of secure computation protocols, which can potentially save a great deal of computation time when the vector space is used for common text mining tasks. The generalized texts serve as a middle ground between fully disclosing private data, which is highly undesirable; and the use of secure, but computationally expensive protocols for joint computation of text mining results.

The next section shows the merit of text generalization as a feature selection technique by examining existing flaws of the vector space model, and demonstrating how they can be overcome they can be overcome by using the generalization method.

3.4. CHALLENGES OF THE VECTOR SPACE MODEL

The most obvious challenge presented by a vector space representation is the high dimensional nature of text data. A single, relatively short document may still contain hundreds of distinct terms. In a large text collection, the number of unique terms reaches into tens of thousands. Large dimensions become a hindrance to the efficiency of clustering and similarity computations. Consider that the cosine similarity is based on the dot product of vectors, and that the calculation of cluster centers is based on the average values of each vector assigned to a cluster. These operations can become near-unworkable if the matrix contains tens of thousands of terms. Hence the reduction of dimension is a high priority for any feature selection technique.

In addition to issues with dimensionality, certain types of words are problematic to the vector space model. These problems are known as *polysemy* – When a word has more than one meaning, and *synonymy* – when more than one word exists with the same meaning. To illustrate how these words can affect the accuracy of the vector space model, consider the following sentences:

S_1: **Orange** is my favorite color.

S_2: He wore an **orange** colored coat.

S_3: An **orange** is a healthy fruit.

S_4: Yesterday, I purchased a bottle of **citrus sinensis** extract.

S_5: **Apple** farming is popular in Washington state.

It is clear to a human reader that there are two basic topics in these sentences, and that S_1 and S_2 are (somewhat) related, as they are both referring to the word orange in the sense of the color, while S_3 and S_4 are referring to the fruit. When these sentences are placed in a vector space model, some degree of similarity will be incorrectly be detected between S_1, S_2, and S_3 because each contain the word 'orange.' The sense of the word 'orange' is not related to the first two sentences, but the use of this word still results in the sentence being considered similar. Meanwhile, S_4 is not counted as similar to S_3 at all, even though they both refer to the same fruit. These issues are illustrated in the simple term-document frequency matrix:

$$\begin{array}{cc} & \begin{array}{ccccc} Orange & Fruit & Color & Citrus & Apple \end{array} \\ \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{array} & \left(\begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right) \end{array}$$

The word 'orange' is an example of a polysemous word, as it can refer to either a fruit or a color. The presence of polysemous words in a text collection can result in documents being measured as similar when they are actually unrelated. The two synonymous words ‘citrus sinensis’ and ‘orange’ share the same meaning, but are counted as separate terms. The word ‘apple’ is related to the topic of food and fruit, but S_5 is not counted as similar to S_3, S_2 or S_3 because similarity is based only on frequency.

From these examples, some additional shortcomings of the vector space model become apparent. While preprocessing steps like stop-word removal and frequency pruning can remove words with low information content, and can reduce the vector space dimension and remove noise, these methods are unable to detect the presence of synonymous and polysemous words which can result in documents being falsely counted as similar or dissimilar. Consider also that polysemous words raise a privacy-related issue for the generalization method. If the sense of a noun is identified incorrectly, then the replacement word will be unrelated to the topic of the document or sentence. An incorrect substition can affect the accuracy of a similarity measure, or even provide a way to more easily guess the meaning of the original word. For example, in sentence [whatever], if the word ’fruit’ had been substituted in place of ’color,’ a clever reader could easily guess the original meaning of the word. This provides further incentive to use word-sense disambiguation before performing text generalization.

The general task of word sense disambiguation aims to overcome issues relating to polysemous words. If the correct sense of the word can be identified, then by replacing the terms with a common ancestor the relationship between the sentences is correctly detected by the vector space model. A large variety of techniques have been applied for word-sense disambiguation, including considerable number which rely on information disambiguation method presented in [15] is easily adapted here since it is based on the word ontology and hypernym/hyponym relations, The basic idea of the technique in [15] is to identify each possible sense of an ambiguous word, and count the frequency its immediate parent and child words in the document. The sense which maximizes this frequency is chosen as the winner. For example, in S_3 it is seen that the word ‘fruit’ appears in the sentence, and is part of the hypernym tree of ‘orange’. Thus is it this sense which maximizes the frequency of related hypernyms in the sentence, and it is the most appropriate choice. It should be noted that most disambiguation techniques choose a one word sense for an entire document rather than a sentence-by-sentence basis. This is generally appropriate, according to [31] where it is claimed that for most documents only one sense of a polysemous word is likely to appear.

Synonymous words can be addressed by combining terms which share meanings. First, a clarification should be made between two possible treatments of synonymy. In Wordnet, all terms are members of concepts, which are sets of synonymous words. For example, the words ‘school’ and ‘schoolhouse’ are members of the same concept $\mathcal{C}$. Concept membership is used by existing techniques such as [15] and [17] to resolve synonymy, by replacing words by their corresponding concepts. So words which are very closely related synonyms will be correctly consolidated by replacing instances of words by their corresponding concepts. It is important to state that generalization can overcome synonymy simply by replacing both words with a word from the common parent synonym set. However, it is often more efficient to first replace terms with their corresponding synonym set before applying generalization, because ontology based disambiguation techniques such as [15] and [21] work on concepts instead of individual word entries.

In this work however, synonymy is assumed to have a more broad meaning. a distinction is made between true synonyms, which are members of the same concept in the ontology, and immediate sibling words, which are members of distinct concepts sharing a parent concept. This type of relation has been indirectly considered in tree based word sense similarity metrics such as [32] and [33], which have in turn been used in a number of disambiguation techniques [31], but these are aimed at choosing correct senses for polysemous words and will not account for the lack of cosine similarity measured between frequency vectors composed of sibling words.

The following example shows how text generalization can be applied to the vector space model to resolve problems with polysemous and synonymous words. Suppose that the four sentences previously examined are replaced with the following:

S_1: **color** is my favorite color.

S_2: He wore a **color** colored coat.

S_3: An **fruit** is a healthy fruit.

S_4: Yesterday, I purchased a bottle of **fruit** extract.

S_5: **Fruit** farming is popular in Washington state.

The term-document frequency matrix representing these sentences is shown below.

$$\begin{array}{c c} & \begin{array}{cc} Fruit & Color \end{array} \\ \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{array} & \begin{pmatrix} 0 & 2 \\ 0 & 2 \\ 2 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \end{array}$$

S_3 is no longer similar to S_1 and S_2, and S_4 is now similar to S_3.

The polysemy and synonymy issues have both been corrected. The relationship between $S5$ and $S1, S2$, and $S3$ is also made clear.

Further, text generalization can reduce the size of the vector space model. If a text contains a large number of synonymous and polysemous terms, then by generalization these terms are consolidated together. This is seen in the example above, as the number of terms included in the vector space has been reduced from four to two. The number of terms is reduced while retaining the semantic meaning of terms, which leads to the notion of a *semantic preserving vector space model*, in which the words of each document are replaced by their hypernyms.

A final advantage of text generalization is the nature of the resulting vector space. Some feature selection techniques such as [34], [19] and [17] extract entirely new sets of features from the original vector space. While these techniques show a considerable improvement in clustering accuracy, similarity detection, dimension reduction and so on, they do so while modifying the original format of the data. Text generalization still results a matrix of frequencies of terms in documents. This means that further feature selection techniques or modifications can be applied if desired. This format can also help in easily interpreting the cluster results by simply considering the most frequenctly occuring terms assigned in each cluster.

3.5. THE SEMANTIC PRESERVING VECTOR SPACE MODEL

In the SPVSM, document vectors are represented by the frequency of their generalized terms. When documents contain words which are synonyms in the word ontology, the size of the vector space is reduced when generalization is applied. Higher levels of generalization can be expected to yield a greater dimensionality reduction. Additionally, the relationship between synonymous words is not normally detected in the vector space model, but this relationship is accounted for when synonymous terms are generalized to the same parent word. Another strength of the semantic preserving vector space model is that it accounts for both the true synonym relation as well as the close relation between immediate siblings words. This means that the cluster performance will likely be increased by the use of a generalized vector space model.

It was shown by experiment in [16] that nouns are responsible for most of the information content in documents. The experiment demonstrated virtually no difference in document clustering results between using all parts of speech and using nouns exclusively. Selecting nouns exclusively to the vector space provides some immediate relief to the dimensionality issue, hence only nouns are included in the SPVSM. The technique for creating the generalized space can be summarized as follows:

For a document collection D, minimum term frequency m_f, maximum frequency M_f and generalization level g:

1. Terms are added to the feature space only if it exists as a noun in the Wordnet dictionary.
2. Identify Wordnet concepts associated with terms.
3. Perform sense disambiguation
4. Replace concepts by hypernyms, to a level of g where possible.
5. Remove all terms with frequency less than m_f and greater than M_f.

The result of this process is a term-document frequency matrix representing the result of generalizing the text documents. Note that step 2 and 3 can be omitted, and that step 4 can alternatively be placed at the end of the process. The level of generalization is be taken as a parameter in the technique, and some discussion will be given to choosing an appropriate level of generalization given the size of the term space and number of documents in the text.

3.6. EXPERIMENTAL DESIGN

The experimental section of this work has two parts. First, an assessment of the clustering performance on several text datasets. Second, to demonstrate the impact synonymous words, and to survey how the semantic preserving vector space model and other techniques are able to overcome these problems, new datasets were created from the original in which words were replaced by terms from synonymous concepts. After all of the terms of a document are parsed, they are replaced by randomly by a term from a synonymous concept if one exists.

The 20Newsgroups dataset [35] has been frequently used in text processing research since it was first made publicly available in 1999. It contains about 20,000 informal postings (documents) made to Internet newsgroups among 20 manually assigned categories. The distribution of documents is approximately equal between categories. The datasets used here are Religion-Graphics and a four label dataset, Religion-Graphics-Electronics-Motorcycles.

In both experiments, the accuracy of the results are measured by using the Rand index and cluster purity measures. Rand index [1] is an external measure of cluster validity, which means that it examines how closely the cluster assignment produced by the process aligns with ground-truth class labels of the data points [7]. This type of metric is particularly appropriate in this experiment, as the class label assigned to each document is assigned based on the discussion topic of the newsgroup to which the document belongs. More specifically, the Rand index places emphasis on the situation that pairs of points belong to the same group or to different groups. The Rand index for a cluster is calculated by the equation $\mathcal{R} = \frac{TP+TN}{TP+TN+FP+FN}$, where TP is the number of true positives or pairs of points with the same label assigned to the same cluster. True negatives are non-similar points assigned to different clusters, while false positives and false negatives occur when pairs of non-similar points are assigned to same cluster, or when pairs of the same label are assigned to different labeled clusters. False positives and false negatives will result in a lower Rand index score.

Purity [1] is an external measure of accuracy among clusters. Each cluster is assigned the to label which it most frequently contains. The number of points matching the cluster label is compared against the total number of points the cluster contains. When a high value is obtained for cluster purity, it is an indication that data points with the same class label have frequently assigned to the same cluster. Within a cluster C, the purity is defined as $P(C) = \frac{1}{C}\max_{\ell}(|C|_{\ell}$. The overall cluster purity is given by $\mathcal{P} = \sum_{j=1}^{k} \frac{|C_j|}{\mathcal{D}}$

Lastly, the F-measure[1] is a well known measure among statisticians which imposes a stricter penalty on false negatives than false positives. The penalty for false negatives is controlled through the β parameter, where a higher value for β will yield a stronger penalty. The F-measure is calculated by $\mathcal{F}_{\beta} = \frac{(\beta^2+1)PR}{\beta^2 P+R}$.

Measures like the Rand index and F-measure can be adjusted for chance by using the following:

$$\frac{Index - Expected}{Max(Index) - Expected} \tag{3.1}$$

By adjusting the values for chance, the measures are compared against random cluster assignment. The expected value for is the value obtained through randomly assigning documents to clusters. Thus, adjusted measures give insight into the true accuracy of a cluster assignment. Cluster analysis is a modeling technique which assigns data observations into groups based on their similarity. For text clustering, the aim is to place documents which are related to the same topics into the same clusters. Clustering document collections represented by a vector space model is usually performed with a variant of the k-means algorithm. k-means is a partitional clustering algorithm which iteratively assigns a set of observations to k clusters, while attempting to optimize a criteria or condition. Points are initially assigned randomly to prototype clusters, and in each iteration the mean of the attributes of the points assigned to each cluster is calculated. Points are re-assigned to the closest cluster based on the distance between the point and the cluster means. The distance function is also based on the data attributes. After an objective criteria function has been sufficiently optimized, or after a fixed number of iterations, the algorithm terminates.

Spherical k-means [8] is a variant on the k-means algorithm which was developed specifically for clustering text documents. After creating the document-frequency matrix, the weights of the document vectors are normalized to unit length. Spherical k-means introduces 'concept vectors' for each cluster, which are the cluster centroids normalized to have a unit vector length. It is from this normalization that the algorithm draws its name; the vector space can be considered a unit sphere in n-space. The purpose of the normalization is to adjust for the fact that the length of documents may vary greatly across the collection. Documents are assigned to clusters based on their cosine similarity to the concept vectors.

Three of the feature selection methods described in the related work section has been implemented. The cluster metrics were calculated for the result of running the spherical k-means algorithm as a baseline, with Hotho's [15] method; with latent semantic indexing [19]; with Recupero's [17]method; Fodeh's [16] core semantic features; the SPVSM with generalization up to 3 levels; and a combination of generalization together with latent semantic indexing. Hotho's [15] method augments the vector space with information from Wordnet by replacing terms by their related concepts. Recupero [17] builds new feature vectors based on the most frequently appearing Wordnet lexical categories. Latent semantic indexing [19] uses the singular value matrix decomposition to project document vectors into an smaller, but approximately equivalent set of dimensions. Fodeh [16] defines core semantic features as nouns which are both polysemous and synonymous, while exceeding a specified information gain after word sense disambiguation.

For all methods, words were included in the vector space only if they exist as nouns in the Wordnet dictionary. This is comparable to using of a large stop list. Parameters for minimum and maximum term frequencies are employed, and each document representation method is used with minimum frequency values between 1 and 30, and maximum frequency values of 100 and 1000. After parsing documents, the terms are weighted using the TFIDF [2] weighting scheme. For the implementation of clusters with large dimension the Java Matrix Package [36] from the National Institute of Standards was used. All algorithms were implemented using Java. Each method was implemented using Java. The experiments were run in Ubuntu Linux on a Dell Optiplex 755 with 3.00GHz Intel Core 2 Duo E8400 CPU and 6 MB cache.

In some cases there is a slight departure from the methods used in the experimental evaluations described in those papers; for instance, in Recupero's [17] work the bisecting k-means method was used instead of spherical k-means. Also, Recupero's method does not use minimum and maximum frequency thresholds, and so for comparison the same results are repeated in each section of the tables.

3.7. RESULTS

Figures 3.2 and 3.3 show the cosine distance to cluster centroids for the synonymized Religion-Graphics dataset, comparing the baseline against one level of generalization. Tables 3.1 shows the result of evaluating each technique with the metrics described in section 3.6. The first pane is a comparison of overall performance. The second pane shows the result for the second experiment, in which terms in the document collection were randomly replaced by their corresponding synonyms. Because replacement synonyms are randomly chosen in the second experiment, there is some fluctuation in the values obtained for the metrics. While it is expected that overall performance will decrease after the vector space is reduced beyond a certain point, occasionally the random substitution choices will result in higher or lower accuracies. Other variations in the outcome of the experiment include the size of the term space obtained by the feature selection techniques. Also, when the appearances of words across the document collection are split into several synonyms, their individual frequency will be lessened enough so that they may no longer be included in the vector space either due to the frequency thresholds or because of the nature of the feature selection.

Figure 3.4 plots Rand index against vector space dimension for each of the minimum frequency parameters. The lines graphed are levels of generalization, from the baseline up to level three generalization. Table 3.2 displays cluster metrics for the Religion-Graphics data set. Table 3.3 shows the top ten most frequently occuring terms in each cluster. Finally, Table 3.4 shows of the techniques across a variety of parameter settings. Each method requiring WSD uses Hotho's [15] disambiguation by concepts. Generalization and Hotho's concept-replacement are performed after disambiguation, but before thresholds are applied. The methods of Fodeh[16] and

Recupero [17] do not use frequency thresholds, thus the differences in parameters does not apply.

Figure 3.2: Cosine distance to centroids

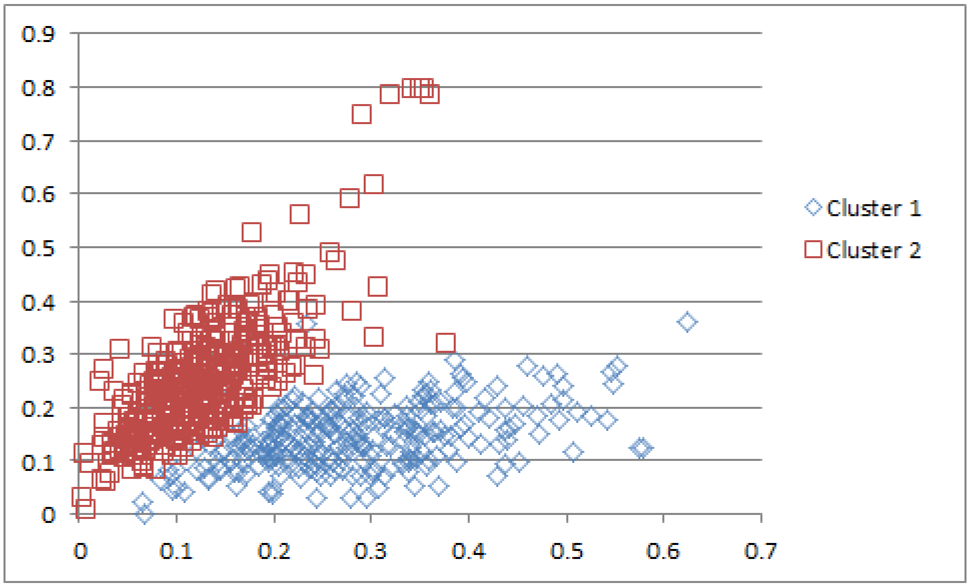

Figure 3.3: Cosine distance to centroids in SPVS model

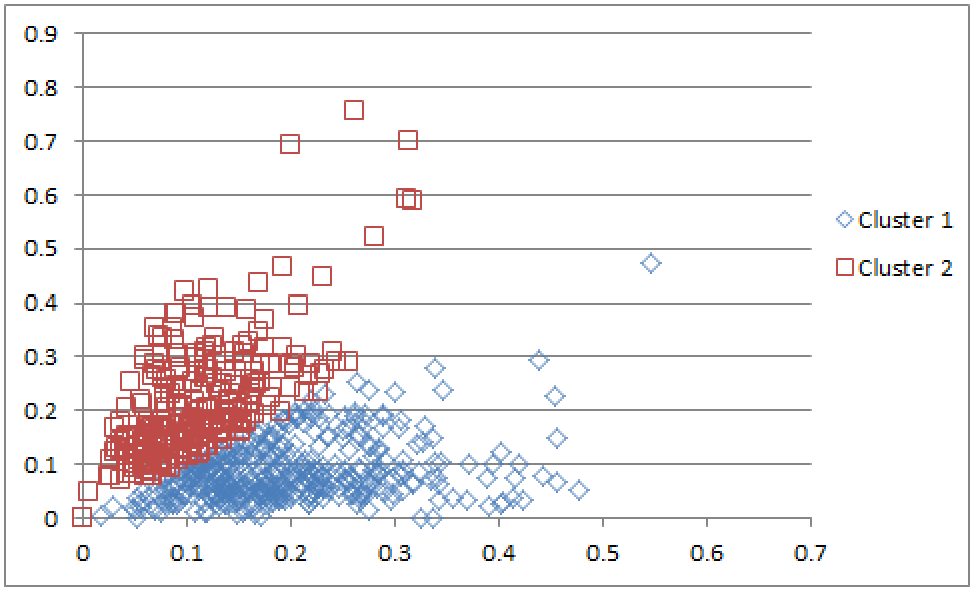

Table 3.1: Cluster metrics for Religion-Baseball-Electronics-Motorcycles data

	Basic Dataset						Synonymized Dataset					
$m_f = 5, M_f = 1000$	Purity	Rand	Adj. R	F	Adj. F	Size	Purity	Rand	Adj. R	F	Adj. F	Size
Baseline	0.681	0.768	0.387	0.706	0.608	2582	0.65	0.74	0.314	0.57	0.427	3113
Hotho	0.609	0.705	0.221	0.653	0.538	2856	0.293	0.575	-0.121	0.344	0.127	3649
Fodeh	0.531	0.66	0.103	0.459	0.280	**96**	0.498	0.655	0.089	0.46	0.281	3123
LSI	0.652	0.75	0.340	0.656	0.542	300	0.641	0.723	0.269	0.68	0.574	300
Recupero	0.552	0.677	0.147	0.413	0.219	471	0.404	0.631	0.026	0.315	0.088	718
Generalization1	0.653	0.766	0.382	0.679	0.572	1856	0.667	0.76	0.366	**0.698**	**0.586**	1720
Generalization2	0.745	0.773	0.401	0.653	0.538	926	0.537	0.658	0.097	0.552	0.404	846
Generalization3	0.445	0.579	-0.110	0.484	0.313	507	0.523	0.628	0.018	0.572	0.430	**279**
Generalization1 w/ LSI	**0.847**	**0.856**	**0.620**	**0.746**	**0.680**	300	**0.671**	**0.766**	**0.382**	0.682	0.576	300
$m_f = 10, M_f = 1000$	Purity	Rand	Adj. R	F	Adj. F	Size	Purity	Rand	Adj. R	F	Adj. F	Size
Baseline	0.674	0.762	0.372	0.706	0.608	1496	0.64	0.756	0.356	0.601	0.469	1506
Hotho	0.6	0.705	0.221	0.654	0.539	1560	0.3	0.576	-0.118	0.34	0.122	1560
Fodeh	0.531	0.66	0.103	0.459	0.280	**96**	0.498	0.655	0.089	0.46	0.281	3123
LSI	0.68	0.765	0.380	**0.715**	**0.620**	300	0.6	0.698	0.203	0.69	0.587	300
Recupero	0.552	0.677	0.147	0.413	0.219	471	0.404	0.631	0.026	0.315	0.088	718
Generalization1	0.651	0.762	0.372	0.625	0.501	1211	0.684	0.762	0.372	0.707	0.610	1047
Generalization2	0.578	0.693	0.190	0.614	0.486	613	0.62	0.718	0.256	0.622	0.497	519
Generalization3	0.451	0.59	-0.081	0.451	0.269	318	0.591	0.695	0.195	0.522	0.364	**257**
Generalization1 w/ LSI	**0.687**	**0.766**	**0.382**	0.705	0.607	300	**0.839**	**0.853**	**0.575**	**0.742**	**0.656**	300
$m_f = 20, M_f = 1000$	Purity	Rand	Adj. R	F	Adj. F	Size	Purity	Rand	Adj. R	F	Adj. F	Size
Baseline	0.665	0.75	0.340	0.731	0.642	751	0.587	0.675	0.142	0.647	0.530	645
Hotho	0.588	0.702	0.213	0.603	0.471	1012	0.287	0.578	-0.113	0.338	0.119	780
Fodeh	0.531	0.66	0.103	0.459	0.280	**96**	0.498	0.655	0.089	0.46	0.281	3123
LSI	0.662	0.745	0.327	0.728	0.638	300	0.629	0.697	0.200	0.617	0.490	300
Recupero	0.552	0.677	0.147	0.413	0.219	471	0.404	0.631	0.0265	0.315	0.088	718
Generalization1	0.851	0.861	0.633	0.744	0.659	694	**0.670**	**0.738**	**0.308**	**0.688**	**0.584**	602
Generalization2	0.53	0.605	-0.042	0.533	0.378	359	0.529	0.632	0.0292	0.614	0.486	329
Generalization3	0.446	0.573	-0.126	0.479	0.306	184	0.482	0.603	-0.047	0.566	0.422	**143**
Generalization1w/LSI	**0.874**	**0.881**	**0.686**	**0.782**	**0.710**	300	0.615	0.717	0.253	0.643	0.525	300

Figure 3.4: Rand Index Versus Vector Space Dimension

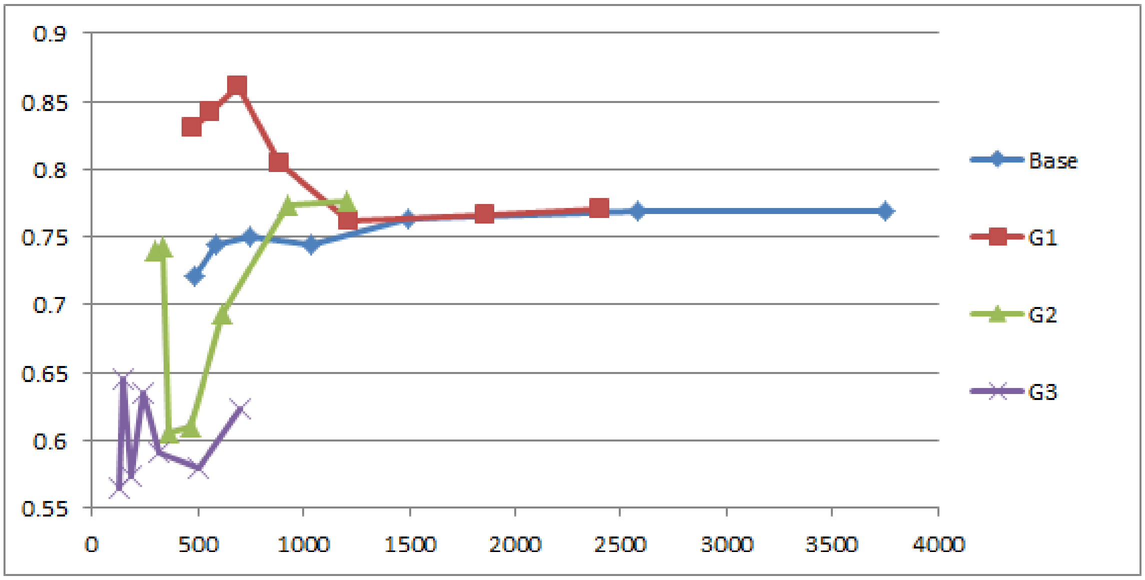

Table 3.2: Cluster metrics for Religion-Graphics data

	Basic Dataset						Synonymized Dataset					
$m_f = 5, M_f = 1000$	Purity	Rand	Adj. R	F	Adj. F	Size	Purity	Rand	Adj. R	F	Adj. F	Size
Baseline	0.945	0.896	0.792	0.887	0.774	1655	0.635	0.536	0.073	0.566	0.133	3113
Hotho	0.671	0.558	0.117	0.637	0.274	1819	.607	.499	0	.500	0.001	3649
Fodeh	0.893	0.809	0.618	0.825	0.650	249	0.706	0.584	0.169	0.635	0.270	2531
LSI	0.946	0.899	0.798	0.890	0.780	**150**	0.773	0.648	0.297	0.664	0.328	**150**
Recupero	0.857	0.755	0.510	0.746	0.492	405	0.874	0.780	0.560	0.770	0.540	717
Generalization1	0.948	0.902	0.804	0.895	0.790	1255	0.951	0.907	0.814	0.899	0.798	1129
Generalization2	0.920	0.853	0.706	0.862	0.724	639	0.926	0.863	0.726	0.871	0.742	545
Generalization3	0.757	0.632	0.265	0.843	0.686	325	0.689	0.570	0.141	0.881	0.762	298
Generalization1 w/ LSI	**.951**	**0.907**	**0.814**	**0.900**	**0.800**	**150**	**0.953**	**0.910**	**0.820**	**0.906**	**0.812**	**150**
$m_f = 10, M_f = 1000$	Purity	Rand	Adj. R	F	Adj. F	Size	Purity	Rand	Adj. R	F	Adj. F	Size
Baseline	**0.954**	**0.913**	**0.826**	0.906	0.812	874	0.64	0.756	0.512	0.601	0.203	1506
Hotho	0.631	0.533	0.0674	0.636	0.272	809	0.608	0.5	0.001	0.5	0.001	875
Fodeh	0.893	0.809	0.6185	0.825	0.650	249	0.706	0.584	0.169	0.635	0.270	2531
LSI	0.953	0.91	0.8202	0.903	0.806	**150**	0.62	0.528	0.057	0.587	0.175	**150**
Recupero	0.857	0.755	0.510	0.746	0.492	405	0.874	0.78	0.560	0.77	0.540	717
Generalization1	0.953	0.91	0.820	**0.948**	**0.896**	766	**0.960**	**0.924**	**0.848**	**0.918**	**0.836**	683
Generalization2	0.909	0.834	0.668	0.857	0.714	380	0.915	0.845	0.690	0.851	0.702	346
Generalization3	0.753	0.627	0.255	0.848	0.696	186	0.67	0.557	0.115	0.897	0.794	177
Generalization1 w/ LSI	0.948	0.902	0.804	0.898	0.796	**150**	0.945	0.896	0.792	0.89	0.780	**150**
$m_f = 20, M_f = 1000$	Purity	Rand	Adj. R	F	Adj. F	Size	Purity	Rand	Adj. R	F	Adj. F	Size
Baseline	0.932	0.874	0.748	0.866	0.732	389	0.587	0.675	0.351	0.647	0.294	645
Hotho	0.882	0.792	0.584	0.804	0.608	295	0.607	0.499	-0.001	0.5	0.001	396
Fodeh	0.893	0.809	0.618	0.825	0.650	249	0.706	0.584	0.169	0.635	0.270	2531
LSI	0.931	0.871	0.742	0.863	0.726	150	0.607	0.5	0.001	0.652	0.304	150
Recupero	0.857	0.755	0.510	0.746	0.492	405	0.874	0.78	0.560	0.77	0.540	717
Generalization1	**0.948**	**0.902**	**0.804**	**0.901**	**0.802**	389	**0.956**	**0.916**	**0.832**	**0.912**	**0.824**	359
Generalization2	0.917	0.847	0.694	0.873	0.746	217	0.928	0.866	0.732	0.866	0.732	198
Generalization3	0.762	0.637	0.275	0.845	0.690	**116**	0.698	0.578	0.157	0.87	0.740	**97**
Generalization1 w/ LSI	0.942	**0.902**	**0.804**	0.9	0.800	150	0.925	0.861	0.722	0.855	0.710	150

Table 3.3: Top Terms For Clusters

No Generalization			
C_1	C_2	C_3	C_4
God	game	battery	can
will	can	clutch	article
Jesus	article	can	bike
Christian	Don	article	Don
can	team	will	will
Lord	year	year	DoD
Bible	time	list	time
people	player	Don	make
Christ	will	concrete	Apr
article	baseball	lead	use
Generalization 1			
God	activity	container	activity
religious_person	time_period	activity	artillery
faculty	container	motor_vehicle	grasping
person	nonfiction	nonfiction	time_period
Jesus	gentleman	person	container
sacred_text	unit	Greg._calendar_month	nonfiction
belief	person	time_period	advantage
container	case	collection	faculty
activity	contestant	kind	happening
content	Gregorian_calendar_month	faculty	database
Generalization 3			
entity	entity	entity	entity
psychological_feature	psy._feature	psy._feature	psy._feature
abstraction	abstraction	living_thing	artifact
living_thing	artifact	abstraction	abstraction
God	attribute	measure	attribute
attribute	living_thing	artifact	living_thing
whole	measure	attribute	measure
measure	whole	whole	whole
artifact	indication	person	social_group
Jesus	idea	idea	idea

Table 3.4: Performance On RBEM Data

Method	Steps	$m_f = 1$	3	5	10	15	20	25	30
Baseline	Cluster	8.69h	28.83m	1.18m	41.16s	26.78s	18.92s	13.76s	10.51s
Hotho[*]	WSD	40.14s	39.11s	38.72s	40.68s	39.07s	39.75s	39.39s	39.08s
	Cluster	10.54h	42.6m	1.47m	46.46s	27.5s	18.47s	13.57s	10.74s
Recupero[†]	WSD	40.14s	39.11s	38.72s	40.68s	39.07s	39.75s	39.39s	39.08s
	Selection	11.2m	11.2m	11.2m	11.2m	11.2m	11.2m	11.2m	11.2m
	Cluster	5.73s	5.73s	5.73s	5.73s	5.73s	5.73s	5.73s	5.73s
Fodeh	WSD	40.14s	39.11s	38.72s	40.68s	39.07s	39.75s	39.39s	39.08s
	Selection	18.5h	18.5h	18.5h	18.5h	18.5h	18.5h	18.5h	18.5h
	Cluster	10.47s	10.47s	10.47s	10.47s	10.47s	10.47s	10.47s	10.47s
LSI	Approximate	13.32m	6.62m	4.07m	2.26m	1.08m	32.70s	19.32s	11.93s
	Cluster	5.49s	5.25s	5.21s	5.42s	5.48s	5.34s	5.40s	5.37s
Gen1	WSD	40.14s	39.11s	38.72s	40.68s	39.07s	39.75s	39.39s	39.08s
	Generalize	1.58s	1.59s	1.67s	1.55s	1.58s	1.61s	1.56s	1.59s
	Cluster	3.05m	1.15m	51.59s	31.31s	21.91s	16.11s	12.66s	10.13s
Gen2	WSD	40.14s	39.11s	38.72s	40.68s	39.07s	39.75s	39.39s	39.08s
	Generalize	3.63s	3.58s	3.54s	3.53s	3.58s	3.45	3.48	3.50
	Cluster	55.17s	31.82s	23.38s	14.43s	10.38s	7.84s	5.87s	4.79s
Gen2	WSD	40.14s	39.11s	38.72s	40.68s	39.07s	39.75s	39.39s	39.08s
	Generalize	5.27s	5.30s	5.26s	5.21s	5.22s	5.23s	5.37s	5.23s
	Cluster	32.17s	15.75s	11.16s	5.66s	3.98s	3.08s	2.51s	2.28s
G1,LSI	WSD	40.14s	39.11s	38.72s	40.68s	39.07s	39.75s	39.39s	39.08s
	Generalize	1.60s	1.62s	2.39s	1.69s	1.68s	1.60s	1.65s	1.65s
	Approximate	6.7m	4.1m	2.88m	1.56m	48s	28s	17s	12s
	Cluster	6.27s	5.29s	5.34	5.35s	5.37s	5.30s	5.39s	5.38s

[*] Hotho's disambiguation is used across each of the methods.

[†] Since this technique does not use minimum frequency thresholds, the result is the same for each value.

4. DISCUSSION

For the RBEM four label dataset shown in Table 3.1, the best values for each of the cluster metrics was obtained using generalization to one level and latent semantic indexing. Even without latent semantic indexing, it is possible to achieve an accurate result with greatly reduced dimensions using the SPVSM representation. A higher minimum frequency seems appropriate for generalization, since as the term space is reduced, the average term frequency may increase. This is reflected by the fact that the best result for generalization occurred with the highest minimum frequency value. In the synonymized data experiment, some of the feature selection techniques are thrown off by the change and actually perform worse than SPK alone. Note however that the combination of generalization and latent semantic indexing is able to cluster the synonymized data at a very effective level.

In the top panel of Table 3.1 where $m_f = 5$ and $M_f = 1000$, by combining the SPVSM with latent semantic indexing the vector space is reduced to 11% of its original size while achieving an increase of 24.3% in purity, 11.1% in Rand index and 5.6% in the F measure. Referring to Table 3.4, it is seen that by first creating the SPVSM, the time required to perform latent semantic indexing is reduced by over a minute for $m_f = 5$. From these results, the SPVSM representation produces a reduced dimension vector space with improved accuracy in a very reasonable time frame. Also noteworthy is the case for $m_f = 20$ and $M_f = 1000$, where generalization by one level boosts cluster purity by 27.9% and Rand index by 14.8%. When using frequency thresholds together with generalization, it also becomes possible to more finely control the size of the vector space by varying the frequency cutoffs and generalization level. When two previously distinct terms are replaced by a common subsumer it is possible that their combined frequency will be above the minimum frequency threshold when neither of the original terms would have been included in the feature space. Observe that the difference vector space dimension between the first level of generalization and the baseline for these parameters is smaller than in the other two panels. This indicates that by generalizing synonymous terms, the frequency of some representative words has been pushed above the minimum threshold.

The binary dataset in Table 3.2 is easier to classify. For most values of the parameters, each of the techniques are able to perform at an effective level. Note that two levels of generalization yields only 217 features, with only a very slight reduction in cluster performance. In Table 3.2, most techniques exhibit a significant drop in performance for the synonymized data. For instance, for $m_f = 20$, the baseline algorithm cluster purity is reduced by 36% and the Rand index score drops by 22%. LSI as well as the techniques of Hotho and Fodeh method show a similar performance loss. The score for Adjusted Rand Index is close to zero for several techniques, which means that the cluster performance is no better than a random assignment. As expected, generalization combines words from sibling concepts into a common ancestor and is able to resolve this type of synonymy reasonably well.

Clustering in the SPVSM is significantly faster than the baseline. Table 3.4 shows the computation time results for an array of parameters on the RBEM dataset. The time required for feature selection and to create the generalized vector space is counted separately from the time spent on performing clustering. Using no minimum frequency, the baseline performance is abysmal and requires several hours to complete. This is reduced to a few minutes by using the SPVSM approach with the same parameter values. The time required to perform text generalization is consistently small across all parameter values, and adds only a few seconds to the computation time.

By viewing the size of the vector space across a variety of parameters, the benefits and limitations of the techniques are placed in perspective. Clearly, the common parameters of the experiment - Frequency thresholds and Wordnet lookup - are responsible for a significant portion of space reduction. A considerable number of terms are discarded by verifying the existence of each word as a noun in the Wordnet dictionary during the parsing stage. There are about 205,000 distinct strings in Wordnet 3.0 [5], with 146,000 nouns according to the statistics provided by the Wordnet manual. From these numbers, a crude estimate is that at least 50% of the terms in a document will be discarded when only nouns are selected.

Figure 3.4 plots Rand index score against the number of terms in the vector space for each level of generalization and minimum frequency, as well as the baseline. It is clear that the chosen parameters can greatly affect the outcome of the results. The minimum and maximum frequency cutoffs should be chosen appropriately based on the number of documents in the collection and the expected number of terms. It is apparent from the results that higher levels of generalization yields an appreciable improvement, provided that the feature space is large enough to leave a substantial number of terms remaining.

The same principle is evident in Figure 3.2 and 3.3. By using two clusters, the distance between observations and cluster centers can easily be visualized. In Figure 3.2 and 3.3, each axis represents the distance to the center of a cluster using the synonymized religion-graphics dataset. The x axis represents the proximity to a cluster containing the majority of documents labelled ‘religion’, while the y axis is the similarity to the ‘graphics’ centroid. While the similarity between the documents and their assigned cluster centers is increased, in some cases the similarity between documents and the opposite cluster is also increased. This means words which are members of the same broad category that are not actually synonymous are being substituted with the same value. This helps to explain the degradation in accuracy at higher generalization levels.

Examining the most frequently occuring terms in each cluster is a useful aid in interpreting the results. Additionally, comparing the top terms across several levels of generalization illustrates the affect which generalization has on the semantic information of the documents. Table 3.3 shows the top 10 terms of each cluster for the 4 label dataset using generalization levels of 0, 1, and 3. For the ungeneralized and 1-generalized data, it is relatively easy to interpret the main topic of each of the cluster based on the top terms. At higher generalization levels, this is more difficult. Recall that in Wordnet, each term is a member of a large ontology called a lexical category. By looking at the top terms in each, it becomes apparent that the distribution of words within lexical categories is heavily skewed, because after three levels of generalization, most of the top terms in each cluster are identical.

Each of these results provide motivation for possible improvements to the model. Thus far, generalization has been applied uniformly to all terms in the vector space. Incorporating factors such as term frequency, lexical category, tree-depth and other information into a non-uniform generalization approach would likely improve the accuracy of the technique. Each term substition can either increase or decrease the cluster accuracy. The distribution of terms in lexical categories indicates that it may be worthwhile to identify the most frequently occuring lexical categories and perform a more limited generalization on those terms, as they could be responsible for the decline in performance at higher generalization levels.

5. CONCLUSIONS

5.1. FUTURE WORK

As seen in Table 3.3, some replacements can actually negatively affect the representation, because certain terms become so frequent among all documents that it is no longer to percieve any difference between the topics contained in the collection. Suppose that $\frac{t_f}{|D|}$ is the number of documents containing a term t divided by the total number of documents in the corpus. A very simple rule for non-uniform generalization would be to consider the difference in this quantity between the original term and its subsumer. If this quantity is greater than a certain threshold - One half, for instance - then the word should not be generalized. Similarly, considering the pairwise cosine similarity between all documents containing base terms (terms to be replaced by hypernyms) and considering if the documents become more similar or less with each generalization. Third, additional Wordnet relationships could be incorporated in the process. This is desirable because related words can appear in different lexical categories. For instance, 'mathematics' and 'mathematician' are in separate categories. By using the 'pertains to' relationship in Wordnet, the relation between such words could be detected. Nonetheless, this deserves a more serious exploration in future work, and this discussion is intended to describe how the issue might be approached. Taking into account that uniform generalization can already enhance text mining performance, it is reasonable to expect that a more careful selection of generalized terms will result in further accuracy improvements.

A second direction for non-uniform generalization is to choose term substitutions based on the privacy preservation requirements. Certain sensitive terms may require high level of generalization, but it was seen in the experiment results that uniformly generalizing more than two levels usually resulted in poor performance. However, uniform generalization may not be necessary if the most sensitive terms are identified programmatically, or specified in advanced. Determining these words and appropriately replacing them while allowing less sensitive words to remain will grant better control between the trade-off of privacy and performance.

5.2. CONCLUSION

This thesis described a method for overcome challenges presented by the vector space representation of text data. A feature selection technique was described which can provide relief to the problems of dimensionality, synonymy and polysemy. It can also easily be used in combination with other techniques, such as advanced word sense disambiguation and latent semantic indexing. The application to privacy preserving data mining can potentially greatly reduce the required computation time to perform text mining tasks such as document clustering. The experimental results provides insight to the worthiness of the semantic preserving vector space model in clustering, and to the effect of performing several levels of generalization. The technique can be improved by more selectively choosing terms for generalization, rather than applying hypernym substitutions uniformly.

BIBLIOGRAPHY

[1] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

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

[3] G Zervas and S.M. Ruger. The curse of dimensionality and document clustering. In *Proceedings of the IEEE Searching for Information: AI and IR Approaches*, 1999.

[4] Wei Jiang, Mummoorthy Murugesan, Chris Clifton, and Luo Si. Similar document detection with limited information disclosure. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 735–743, Washington, DC, USA, 2008. IEEE Computer Society.

[5] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[6] National Library of Medicine Controlled Vocabulary. Medical subject headings.

[7] Rui Xu and Don Wunsch. *Clustering.* Wiley-IEEE Press, 2009.

[8] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Mach. Learn.*, 42(1-2):143–175, January 2001.

[9] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *In KDD Workshop on Text Mining*, 2000.

[10] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, pages 318–329, New York, NY, USA, 1992. ACM.

[11] A. Amine, Z. Elberrichi, M. Simonet, and M. Malki. Wordnet-based and n-grams-based document clustering: A comparative study. In *Broadband Communications, Information Technology Biomedical Applications, 2008 Third International Conference on*, pages 394 –401, nov. 2008.

[12] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 16–22, New York, NY, USA, 1999. ACM.

[13] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graphpartitioning. Technical report, Austin, TX, USA, 2001.

[14] Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, June 2004.

[15] Andreas Hotho, Steffen Staab, and Gerd Stumme. Wordnet improves text document clustering. In *In Proc. of the SIGIR 2003 Semantic Web Workshop*, pages 541–544, 2003.

[16] Samah Fodeh, Bill Punch, and Pang-Ning Tan. On ontology-driven document clustering using core semantic features. *Knowl. Inf. Syst.*, 28(2):395–421, August 2011.

[17] Diego Reforgiato Recupero. A new unsupervised method for document clustering by using wordnet lexical and conceptual relations. *Inf. Retr.*, 10(6):563–579, December 2007.

[18] C. Bouras and V. Tsogkas. Clustering user preferences using w-kmeans. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2011 Seventh International Conference on*, pages 75 –82, 28 2011-dec. 1 2011.

[19] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[20] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[21] Samah Jamal Fodeh, William F Punch, and Pang-Ning Tan. Combining statistics and semantics via ensemble model for document clustering. In *Proceedings of the 2009 ACM symposium on Applied Computing*, SAC '09, pages 1446–1450, New York, NY, USA, 2009. ACM.

[22] Chihli Hung, S. Wermter, and P. Smith. Hybrid neural document clustering using guided self-organization and wordnet. *Intelligent Systems, IEEE*, 19(2):68 – 77, mar-apr 2004.

[23] Eneko Agirre and German Rigau. Word sense disambiguation using conceptual density. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, pages 16–22, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[24] Pu Wang and Carlotta Domeniconi. Building semantic kernels for text classification using wikipedia. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 713–721, New York, NY, USA, 2008. ACM.

[25] Jaideep Vaidya and Chris Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 206–215, New York, NY, USA, 2003. ACM.

[26] Mahir Can Doganay, Thomas B. Pedersen, Yücel Saygin, Erkay Savaş, and Albert Levi. Distributed privacy preserving k-means clustering with additive secret sharing. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*, PAIS '08, pages 3–11, New York, NY, USA, 2008. ACM.

[27] Geetha Jagannathan and Rebecca N. Wright. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 593–599, New York, NY, USA, 2005. ACM.

[28] Mikhail J. Atallah, Craig J. McDonough, Victor Raskin, and Sergei Nirenburg. Natural language processing for information assurance and security: an overview and implementations. In *Proceedings of the 2000 workshop on New security paradigms*, NSPW '00, pages 51–65, New York, NY, USA, 2000. ACM.

[29] Ycel Saygin, Dilek Hakkani-Tr, and Gkhan Tr. Sanitization and anonymization of document repositories. In John Erickson, editor, *Database Technologies: Concepts, Methodologies, Tools, and Applications*, pages 2129–2139. IGI Global, 2009.

[30] Wei Jiang, Mummoorthy Murugesan, Chris Clifton, and Luo Si. t-plausibility: Semantic preserving text sanitization. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 03*, CSE '09, pages 68–75, Washington, DC, USA, 2009. IEEE Computer Society.

[31] Roberto Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, February 2009.

[32] J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33, 1997.

[33] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[34] William B. Cavnar and John M. Trenkle. N-gram-based text categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[35] K. Lang. 20 newsgroups data set. `http://www.ai.mit.edu/people/jrennie/20Newsgroups/`.

[36] J. Hicklin. Jama: A java matrix package. `http://math.nist.gov/javanumerics/jama`, 1999.

VITA

Michael Howard earned a bachelor's degree in Mathematics from the University of Central Missouri in 2008. Upon graduation, he worked as a software engineer at Harris Corporation, developing a sales-and-scheduling application used by the largest media congolomerates in North America. In 2010, he enrolled in the graduate program of Missouri S&T while also working as a programmer at the United States Geological Survey. In December 2012, he received his Master's Degree in Computer Science from Missouri University of Science and Technology.