COMPARATIVE ANALYSIS OF MEASURES OF SIMILARITY AND SEMANTIC RELATEDNESS FOR TEXT CLASSIFICATION

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science & Engineering

By:
Shirin Chandna
(800832016)

Under the supervision of:
Ms. Shalini Batra
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

July 2010

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Comparative Analysis of Measures of Similarity and Semantic Relatedness for Text Classification"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Ms. Shalini Batra* and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Shirin Chandna.

(Shirin Chandna)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Ms. Shalini Batra)
Assistant Professor
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by:

(Dr. Rajesh Bhatia)
Head
Computer Science & Engineering, Department
Thapar University
Patiala

(Dr. R.K. Sharma) 2/7/10
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

I would like to start by thanking GOD ALMIGHTY for providing me with faith, self confidence, ability and strength to complete this thesis.

I would like to place on record my deepest gratitude to Ms. Shalini Batra, Assistant Professor, Thapar University, Patiala for her valuable advice and guidance in carrying out the thesis. I appreciate her unlimited help and patience with her students. I will always remain indebted to her for the moral support, encouragement and the infectious zeal and enthusiasm for work that she imbibed upon me.

With great pleasure and acknowledgement, I extend my profound thanks to Dr. Rajesh Bhatia, Assistant Professor and Head (CSED), Thapar University, Patiala who has been a constant source of inspiration for me throughout this work.

I am also thankful to all the staff members of the Department for their full cooperation and help.

Thanks are also due to my friends Sugandha, Rashmi, Sushila, Tikka and Yogesh for boosting me with their constant encouragement and confidence and bearing me, day and night throughout the thesis work.

Lastly, I am thankful to my parents for their blessings and moral support without which this thesis would not have been possible.

Shirin Chandna

ABSTRACT

Due to the increased availability of documents in digital form, the automated categorization (or classification) of texts into predefined categories has witnessed a large interest among researchers in the last few years. In the research community, the dominant approach to the text classification problem is based on Machine Learning techniques, in which an inductive process automatically builds a classifier by learning, from a set of preclassified documents.

The advantages of this approach over the knowledge engineering approach (consisting in the manual definition of a classifier by domain experts) are effectiveness, considerable savings in terms of expert labor power, and straightforward portability to different domains.

In this thesis, different techniques like Latent Semantic Indexing (LSI) and measures of semantic relatedness and similarity for text classification are discussed.

Latent Semantic Indexing is based upon the assumption that there is an underlying semantic structure in textual data, and that the relationship between terms and documents can be re-described in this semantic structure form. The key idea of Latent Semantic Indexing (LSI) is to map documents on to a vector space of reduced dimensionality, called the latent semantic space. This mapping is done using a technique called Singular Value Decomposition (SVD).

Semantic relatedness measures quantify the degree in which some words or concepts are related, considering not only similarity but any possible semantic relationship among them. In this thesis, various semantic relatedness measures that use the WordNet as their knowledge source and others MSRs like NGD and NCD which make use of the Web as their knowledge base are computed. These semantic measures are tested and their correlation with human judgement is checked.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
INTRODUCTION

Considering the vast amounts of textual information available on the internet today, retrieving information relevant to a user's specific information need from large collections of documents such as the World-Wide-Web, scientific literature, medical data, news repositories, internal databases of companies, etc. has become a great challenge as well as a pressing need. Quite coincidentally, this has led to an equally explosive interest in accurate methods to filter, categorize and retrieve information relevant to the end consumer. Of special emphasis in such systems is the need to reduce the burden on the end consumer.

Modern search engines such as Google and Yahoo have made efficient access to relevant information possible through keyword based search on the web. The basic idea is to preindex the entire Web and quickly retrieve a ranked list of documents that match the user's key words based on some weighted ranking functions.

The Information Retrieval (IR) research community has addressed the same challenge of efficient information access from large collections through the formal problem of ad hoc retrieval defined by the Text Retrieval Conference (TREC). Ad hoc retrieval, also sometimes called query based retrieval, is considered a core research problem in IR and can be defined as the problem of retrieving a ranked list of documents relevant to a query from a large collection of documents. The TREC research community has developed certain standardized test beds and objective evaluation metrics to measure the quality of retrieval algorithms [1, 2] for this task.

1.1 An Overview of Classification

The notion of classification is very general and has many applications within and beyond information retrieval (IR). For instance, in computer vision, a classifier may be used to divide images into classes such as landscape, portrait, and neither. In the context of information retrieval, it has applications such as:

- The automatic detection of spam pages (which then are not included in the search engine index).
- *Sentiment detection* or the automatic classification of a movie or product detection review as positive or negative. An example application is a user searching for negative reviews before buying a camera to make sure it has no undesirable features or quality problems.
- *Personal email sorting*: A user may have folders like talk announcements, sorting electronic bills, email from family and friends, and so on, and may want a classifier to classify each incoming email and automatically move it to the appropriate folder. The most common case of this application is a spam folder that holds all suspected spam messages.
- *Topic-specific or vertical search*: Vertical search engines restrict searches to a particular topic. For example, the query computer science on a vertical search engine for the topic China will return a list of Chinese computer science departments with higher precision and recall than the query computer science China on a general purpose search engine. This is because the vertical search engine does not include web pages in its index that contain the term china in a different sense (e.g., referring to a hard white ceramic), but does include relevant pages even if they do not explicitly mention the term China.
- Several of the preprocessing steps necessary for indexing: detecting a document's encoding (ASCII, Unicode UTF-8 etc), word segmentation (Is the white space between two letters a word boundary or not?); true casing; and identifying the language of a document.

In Machine Learning, the classification task is commonly referred to as supervised learning. In supervised learning there is a specified set of classes, and example objects are labelled with the appropriate class. The goal is to generalize (or form class descriptions) from the training objects, that will enable novel objects to be identified as belonging to one of the classes. The success of classification learning is heavily dependent on the quality of the data provided for training; a learner has only the input to learn from. If the

data is inadequate or irrelevant then the concept descriptions will reflect this and misclassification will result when they are applied to new data.

The process of classification is composed of two steps which are:

- *Training*: Supervised learning of a training set of data to create a model.
- *Testing*: Classifying the data according to that model.

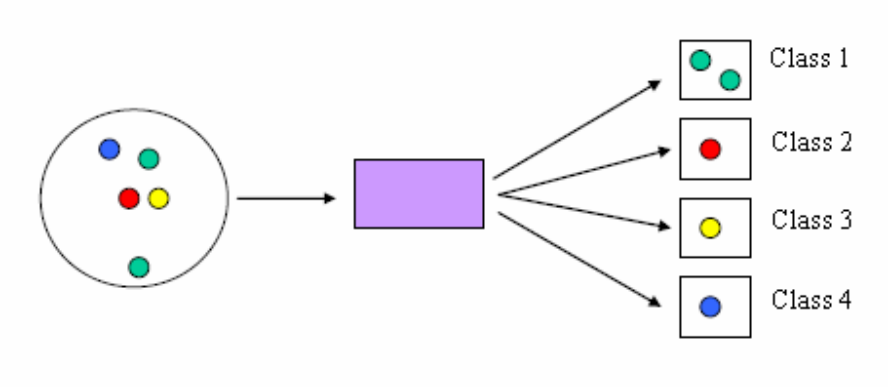

Figure 1.1: A classification process

Some well-known classification algorithms include Bayesian Classification (based on Bayes Theorem), decision trees, neural networks and back propagation (based on neural networks), k-nearest neighbor classifiers (based on learning by analogy), and genetic algorithm.

1.1.1 Text Classification

During the last twenty years the number of text documents in digital form has grown enormously in size. As a consequence, it is of great practical importance to be able to automatically organize and classify documents. Research into text classification aims to partition unstructured sets of documents into groups that describe the contents of the documents. There are two main variants of text classification: text clustering and text categorization. The former is concerned with finding a latent group structure in the set of documents, while the latter (also known as text classification) can be seen as the task of structuring the repository of documents according to a group structure that is known in advance.

Text classification/categorization is the task of assigning an electronic document to one or more categories, based on its contents. Text classification tasks can be divided into two sorts: *supervised document classification* where some external mechanism (such as human feedback) provides information on the correct classification for documents, and *unsupervised document classification*, where the classification must be done entirely without reference to external information. There is also a *semi-supervised document classification*, where parts of the documents are labeled by the external mechanism.

The goal of text categorization is to classify documents into a fixed number of predefined categories. Each document can be in multiple, exactly one or no category at all. Using machine learning, the objective is to learn classifiers from examples which perform the category assignments automatically.

The steps followed in text categorization are shown in Figure 1.2. The first step is to transform documents, which typically are strings of characters into a representation suitable for the learning algorithm and the classification task. Then, word stemming is performed. Information Retrieval research suggests that word stems work well as representation units and that their ordering in a document is of minor importance in many tasks. This leads to an attribute-value representation of text. Each word w_i corresponds to a feature, with the number of times word w_i occur in the document as its value. To avoid unnecessarily large feature vectors, words are considered as features only if they occur in the training data atleast 3 times and if they are not stop words (‘like’, ‘and’, ‘or’ etc.).

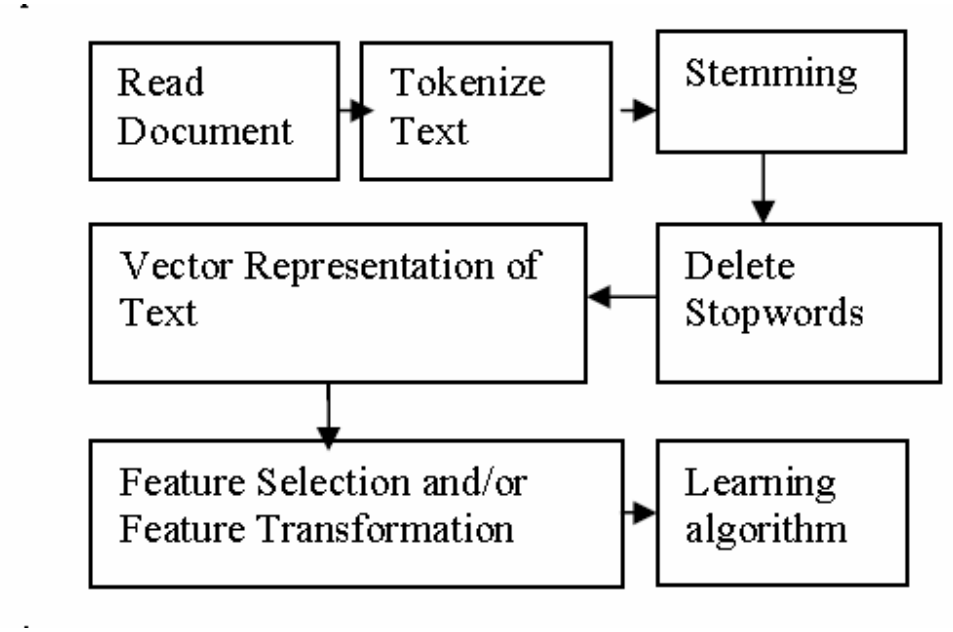

Fig 1.2 Text Classification process [3]

Document classification appears in many applications, including e-mail filtering, mail routing, spam filtering, news monitoring, selective dissemination of information to information consumers, automated indexing of scientific articles, automated population of hierarchical catalogues of Web resources, identification of document genre, authorship attribution, survey coding and so on. Automated text categorization is attractive because manually organizing text document bases can be too expensive, or unfeasible given the time constraints of the application or the number of documents involved.

1.2 Classification Algorithms

The goal of classification is to build a set of models that can correctly predict the class of the different objects. The input to these methods is a set of objects (i.e., training data), the classes which these objects belong to (i.e. dependent variables), and a set of variables describing different characteristics of the objects (i.e. independent variables). Once such a predictive model is built, it can be used to predict the class of the objects for which class information is not known a priori. The key advantage of supervised learning methods over unsupervised methods (for example, clustering) is that by having an explicit knowledge of the classes the different objects belong to, these algorithms can perform an effective feature selection if that leads to better prediction accuracy.

The following is a brief overview of some classification algorithms that have been used in data mining and machine learning area.

1.2.1 k-Nearest Neighbor (KNN) Algorithm

KNN classifier is an instance-based learning algorithm that is based on a distance function for pairs of observations, such as the Euclidean distance or Cosine. In this classification paradigm, k nearest neighbors of a training data are computed first. Then the similarities of one sample from testing data to the k nearest neighbors are aggregated according to the class of the neighbors, and the testing sample is assigned to the most similar class. One of advantages of KNN is that it is well suited for multi-modal classes as its classification decision is based on a small neighborhood of similar objects (i.e., the major class). So, even if the target class is multi-modal (i.e., consists of objects whose independent variables have different characteristics for different subsets), it can still lead to good accuracy. A major drawback of the similarity measure used in KNN is that it uses all features equally in computing similarities. This can lead to poor similarity measures and classification errors, when only a small subset of the features is useful for classification.

1.2.2 Naive Bayesian (NB) Algorithm

NB algorithm[4] has been widely used for the purpose of document classification. The basic idea is to use the joint probabilities of words and categories to estimate the probabilities of categories, given a document. NB algorithm computes the posterior probability that the document belongs to different classes and assigns it to the class with the highest posterior probability. The posterior probability of class is computed using Bayes rule and the testing sample is assigned to the class with the highest posterior probability. The naive part of NB algorithm is the assumption of word independence that the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category. There are two versions of NB algorithm. One is the multi-variate Bernoulli event model [5] that only takes into

account the presence or absence of a particular term, so it doesn't capture the number of occurrence of each word. The other model is the multinomial model that captures the word frequency information in documents.

1.2.3 Support Vector Machines (SVMs)

Support vector machines (SVMs) [6] are a set of supervised learning methods used for classification and regression. In simple words, given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

1.2.4 Latent Semantic Indexing

Latent Semantic Indexing (LSI) [6] is an information retrieval method that organizes information into a semantic structure that takes advantage of some of the implicit higher-order associations of words with text objects. The resulting structure reflects the major associative patterns in the data while ignoring some of the smaller variations that may be due to idiosyncrasies in the word usage of individual documents. This permits retrieval based on the the "latent" semantic content of the documents rather than just on keyword matches.

Latent Semantic Indexing [7] takes advantage of the implicit higher-order structure of the association of terms with articles to create a multi-dimensional semantic structure of the information. This approach models documents as bag-of-words and arranges them in a

Term-Document Matrix. Through the pattern of co-occurrences of words, LSI is able to infer the structure of relationships between articles and words. A Singular-value decomposition (SVD) [8] of the term by article association matrix is computed producing a reduced dimensionality matrix containing the best K orthogonal factors to approximate the original matrix as the model of "semantic" space for the collection. This semantic space reflects the major associative patterns in the data while ignoring some of smaller variations that may be due to idiosyncrasies in the word usage of individual documents. In this way, LSI produces a representation of the underlying latent semantic structure of the information. Retrieving information in LSI overcomes some of the problems of keyword matching by retrieval based on the higher level semantic structure rather than just the surface level word choice. Research on LSI [9] shows that retrieval of relevant documents is significantly improved over keyword matching.

The steps that LSI performs when indexing a document [10] are as follows:

1) *Linearisation*: All the markup tags (i.e. code) from a page are removed so that all its content is represented as a series of characters.
2) *Tokenization*: Formatting such as punctuation, capitalisation and markup etc. are removed.
3) *Filtration*: A stop list is applied to remove commonly used words from the document. This leaves with only the content words.
4) *Stemming*: The remaining content words are then ‘stemmed’, i.e. the remaining terms are reduced to common word roots (e.g. ‘techno’ for ‘technology’, ‘technologies’, ‘technological’).
5) *Weighting*: Weighting is the process of determining how important a term is in a document. ’Term weighting’ means the importance given to terms or words that appear in a document.

Latent Semantic Indexing is a statistical technique that leverages word co-occurrence from large unlabeled corpus of texts. Although it can be successfully used for many purposes, its coverage is still restricted to the corpus used as input, and it needs costly pre-processing tasks. Therefore, other methods like measures of semantic relatedness are considered as alternatives to text classification.

1.3 Measures of Semantic Relatedness

Semantic relatedness measures quantify the degree in which some words or concepts are related, considering not only similarity but any possible semantic relationship among them. Many applications, in Natural Language Processing (NLP) and other fields, benefit from calculating measures to determine numerically how semantically related two words are. Semantic measures can also be defined between lexically expressed word senses, or between whole texts.[11]

There are at least two kinds of similarity: *Attributional similarity* is correspondence between attributes and *Relational similarity* is correspondence between relations (Medin, Goldstone, and Gentner 1990). When two words have a high degree of attributional similarity, they are called *synonyms*. When two word *pairs* have a high degree of relational similarity, they are *analogous*.[12]

Three main kinds of measures are considered about this topic: *semantic similarity*, *semantic relatedness* and *semantic distance*. Unfortunately they have not been interpreted always in the same way by different authors. These are as follows:

1. *Semantic similarity*: It is usually defined by considering lexical relations of *synonymy* (e.g. *car, automobile*) and *hypernymy* (the meaning of a word is encompassed by another more general term, as in *car, vehicle*).

2. *Semantic relatedness*: It covers any kind of lexical or functional association, so it is a more general concept than *semantic similarity*. Dissimilar entities may still be related by many possible relationships, such as *meronymy* (or "part of" relation, as in *finger, hand*), *antonymy* (opposite meanings, as *hot, cold*), or any kind of functional relationship or frequent association (for example, *penguin, Antarctica*, that are not linked by any lexical relation).

3. *Semantic distance*: It is the inverse of *semantic relatedness*. The more two terms are semantically related, the more semantically close they are. [11]

1.3.1 Calculation of Semantic Relatedness

Typically, automated systems assign a score of semantic relatedness to a given pair of words (target words) calculated from a relatedness measure. The absolute score is usually

irrelevant on its own. For example, a relatedness score of 0.7 between a and b, in a possible range of 0 to 1, does not imply that a and b are more related than the average word pair. However, given that the semantic relatedness of c and d is 0.6, the system can conclude that a and b are more related than c and d. Thus even though the absolute score given by a relatedness measure is not of much significance, it is important that the measure give a higher score to word pairs which humans think are more related and comparatively lower scores to word pairs that are less related.[13]

Semantic Similarity represents a special case of semantic relatedness: for example, cars and gasoline would seem to be more closely related than, say, cars and bicycles, but the latter pair are certainly more similar [14]. Semantic similarity measures are necessary for various applications in natural language processing such as word-sense disambiguation, language modeling, synonym extraction, and automatic thesauri extraction. Manually compiled taxonomies such as WordNet and large text corpora have been used in previous works on semantic similarity.

A new method which uses page counts and text snippets extracted from result pages of web searches to measure semantic relatedness between words is proposed. They achieve a high correlation measure of 0.83 on the Charles-Miller benchmark dataset. In yet another method, the semantic relatedness using the Normalized Google Distance (NGD), in which Google is used to determine how closely, related two words are on the basis of their frequency of occurring together in web documents.

1.4 Thesis Outline

The thesis is divided into six chapters:

Chapter 1: The introduction to the work is provided.

Chapter 2: Literature review section gives a brief survey of research going on in the area of text classification and various methods used for it.

Chapter 3: After going through the literature review, the problem statement has been identified and defined in this chapter.

Chapter 4: The general concepts used in the thesis work are introduced. This chapter gives a description of the techniques used for semantic categorization of text.

Chapter 5: In this chapter, experimental results are achieved by implementing techniques like LSI and various measures of semantic relatedness. The illustrations captured during the work done have also been discussed briefly.

Chapter 6: Focuses on the conclusion of the work presented in this thesis and ideas for further research have been proposed.

Chapter 2
LITERATURE REVIEW

Text categorization is the assignment of text content to predefined categories. As the volume of text content grows continuously, text categorization, acts as a way to organize the text content. Supervised text categorization has a learning (or training) component where pre-defined category labels are manually assigned to a set of documents that will become the basis for subsequent automated categorization, whereas text categorization systems performing unsupervised training (or learning) automatically detect clusters or other common themes in the data that identify topics or labels without manual labelling of the data.

In the late ‘80s the most popular approach to text categorization, at least in the “operational” (i.e., real-world applications) community, was a knowledge engineering (KE), but in 90s this approach is slowly being replaced by the machine learning (ML) paradigm, according to which a general inductive process automatically builds an automatic text classifier by learning, from a set of pre-classified documents. Supervised and unsupervised learning have been the focus of critical research in the areas of machine learning and artificial intelligence. In the literature, these two streams flow independently of each other, despite their close conceptual and practical connections. The automated categorization (or classification) of data into predefined categories has seen a significant rising interest in the last ten years, due to the increased availability of documents in digital form and the ensuing need to organize them. Fabrizio Sebatiani [3], in his survey discusses the main approaches to text categorization that fall within the machine learning paradigm.

Thorsten Joachims [15] proposed Support Vector Machines (SVMs) for text categorization. The theoretical analysis conclude that SVMs acknowledge the particular properties of text : (a) high dimensional feature spaces , (b) few irrelevant features (dense concept vectors) and (c) sparse instance vectors.

The experimental results [16] show that SVMs consistently achieve good performance on text categorization tasks, outperforming existing methods substantially and significantly.

With their ability to generalize in high dimensional feature spaces, SVMs eliminate the need of feature selection, making the application of text classification considerably easier. Furthermore, SVMs don't require any parameter tuning, since they find good parameter settings automatically. All this makes SVMs a very promising and easy to use method for learning text classifiers from examples as compared to others like decision trees, k nearest neighbours algorithm etc..

Marcus et al proposed Latent semantic indexing (LSI) [17] as an Information retrieval technique that takes synonymy and polysemy into account as a means to locate concepts and is based on the Vector Space Model (VSM) approach. This technique compresses document vectors into vectors of a lower-dimensional space whose dimensions are obtained as combinations of the original dimensions by looking at their patterns of co occurrence. In practice, LSI infers the dependence among the original terms from a corpus and "wires" this dependence into the newly obtained, independent dimensions. It takes a large matrix of term-document association data and constructs a "semantic" space wherein terms and documents that are closely associated are placed near one another. The function mapping original vectors into new vectors is obtained by applying Singular value decomposition (SVD) [8] to the matrix formed by the original document vectors. Singular-value decomposition allows the arrangement of the space to reflect the major associative patterns in the data, and ignore the smaller, less important influences. As a result, terms that did not actually appear in a document may still end up close to the document, if that is consistent with the major patterns of association in the data. Position in the space then serves as the new kind of semantic indexing, and retrieval proceeds by using the terms in a query to identify a point in the space, and documents in its neighbourhood are returned to the user.

SVD represents both terms and documents as vectors in a space of choosable dimensionality, and the dot product or cosine between points in the space gives their similarity. In Text Classification, this technique is applied by deriving the mapping function from the training set and then applying it to training and test documents alike.

Generalized Vector Space Models (GVSM) [19] extend the standard Vector Space Model (VSM) by embedding additional types of information, besides terms, in the

representation of documents. An interesting type of information that can be used in such models is semantic information from word thesauri like WordNet.

WordNet [20] is a lexical database in which nouns, verbs, adjectives, and adverbs are each organized into networks of synonym sets (*synsets*) such that each represent one underlying lexical concept and are interlinked with a variety of relations. (A polysemous word will appear in one synset for each of its senses). In their work, George Tsatsaronis and Vicky Panagiotopoulou [19] present a new GVSM model that exploits WordNet's semantic information. The model is based on a new measure of semantic relatedness between terms. Experimental study conducted in three TREC collections reveals that semantic information can boost text retrieval performance by considering the semantic relatedness between the query and document terms. This method computes semantic relatedness between terms and takes into account relation weights, and senses' depth, exploiting all the semantic information a thesaurus can offer.

Existing measures of semantic relatedness rely either on ontologies and semantic networks or just raw text. Budanitsky, Budanitsky and Hirst [14] and Patwardhan et al. did an extensive survey and comparison of the various WordNet-based measures. They selected five measures that use WordNet as their knowledge source: Hirst and St-Onge's [21], Leacock and Chodorow's [22] , Jiang and Conrath's [23] , Lin's [24], and Resnik's [25]. The first is claimed as a measure of semantic relatedness because it uses all noun relations in WordNet; the others are claimed only as measures of similarity because they use only the hyponymy relation.

Lin [24] defined the similarity between two concepts as the information that is in common to both concepts and the information contained in each individual concept.

However, due to the very limited size of Word Net as a knowledge base and the absence of well known named entities (e.g., Harry Potter) in Word Net, researchers have started to look for more comprehensive knowledge bases.

The advent of Wikipedia in 2001 has fulfilled the need for a more comprehensive knowledge base. Many techniques that use Wikipedia to compute semantic relatedness have been developed in the recent years. Among others, Strube and Ponzetto [26] have used Wikipedia to determine semantic relatedness. Their results outperform those

obtained using Word Net, hence showing the effectiveness of Wikipedia in determining the similarity between two words.

Matsuo et. al [27] used a similar approach to measure the similarity between words and apply their method in a graph-based word clustering algorithm. Given a taxonomy of concepts, a straightforward method to calculate similarity between two words (concepts) is to find the length of the shortest path connecting the two words in the taxonomy. If a word is polysemous then multiple paths might exist between the two words. In such cases, only the shortest path between any two senses of the words is considered for calculating similarity.

Bollegala et al. [27] have proposed to use page counts and text snippets extracted from result pages of web searches to measure semantic relatedness between words. They achieve a high correlation measure of 0.83 on the Charles-Miller benchmark dataset [28]. Sahami and Heilman) have used a similar measure. Cilibrasi et al. have proposed to compute the semantic relatedness using the Normalized Google Distance (NGD), in which they used Google™ to determine how closely related two words are on the basis of their frequency of occurring together in web documents. The Cilibrasi and Vitanyi's Normalized Google Distance [29] (NGD) uses the relative frequency whereupon two terms appear on the Web within the same documents. NGD is well-founded on information distance and Kolmogorov complexity theories, and it does not preclude any kind of relationship between compared words.

From the past few years, much research is going on in the area of text classification with large focus on techniques like Latent Semantic Indexing (LSI) etc. and more recently, using various measures of semantic relatedness for the classification of text.

Some groups have proposed to use the description of words present in dictionaries and techniques such as LSI to compute semantic relatedness.

Most of traditional methods to compute semantic measures exploit particular lexical resources: corpus, dictionaries, or well structured taxonomies such as WordNet. Some of them explore path lengths among nodes in taxonomies. Others exploit glosses (textual descriptions of concepts) in dictionaries, while some groups rely on annotated corpora to compute information content. The limited dictionary of the lexical database like WordNet

and search engines like Wikipedia has forced the researchers to think about the World Wide Web as an effective alternate database.

Chapter 3
PROBLEM STATEMENT

3.1 Problem Definition

The task of classifying textual data is of prime importance considering the huge amount of information available today. Traditional machine learning programs use a training corpus of often hand-labeled data to classify new test examples. Training sets are sometimes extremely small, due to the difficult and tedious nature of labelling, and decisions can therefore be difficult to make with high confidence.

The simplest approach to represent the semantics is to treat the text as an unordered bag of words, where the words themselves (possibly stemmed) become features of the textual object. The sheer ease of this approach makes it a suitable candidate for many information retrieval tasks such as search and text categorization. However, this simple model can only be reasonably used when texts are fairly long, and performs sub-optimally on short texts. Furthermore, it is not able to resolve the two main problems of natural language processing (NLP), *polysemy* and *synonymy.*

During the literature review, it was analyzed that Latent Semantic Indexing (LSI) is a purely statistical technique, which leverages word co-occurrence information from a large unlabeled corpus of text. LSI does not use any explicit human-organized knowledge; rather, it “learns" its representation by applying Singular Value Decomposition (SVD) to the words-by-documents co-occurrence matrix. LSI is essentially a dimensionality reduction technique that identifies a number of most prominent dimensions in the data, which are assumed to correspond to “latent concepts". Meanings of words and documents are then represented in the space defined by these concepts.

LSI can be implemented for five applications which are: finding the nearest neighbour of a term (i.e. term which is closest to given term in a corpora), matrix comparison, one to many comparison (i.e. comparing a target text against a set of texts) and finding the coherence among sentences.

Methods of similarity and semantic relatedness can also be used for text classification. The more similar or related two words are, the more likely they are found to be together. These measures can either use WordNet as their source of knowledge or the entire Web as their knowledge base. In both the cases, their values can be compared against the standard Miller Charles values which are based on human judgement.

3.2 Methodology

The step-by-step methodology to be followed for text classification using statistical techniques like Latent Semantic Indexing (LSI) and various measures of semantic relatedness is:

- Study of LSI technique for information retrieval, implement the algorithm and obtain LSI based word or passage vectors, similarities between words and words, words and passages, and passages and passages.
- Calculate similarity values using measures which use the lexical database WordNet as a knowledge base.
- Calculate values of similarity and semantic relatedness using Measures of Semantic Relatedness such as Normalized Google Distance (NGD), Normalized Compression Distance (NCD) which make use of the Web as a source of knowledge.
- Compare the obtained results with the standard similarity values of the Miller-Charles word pairs.

Chapter 4

TEXT CLASSIFICATION TECHNIQUES

Text classification (or text categorization) is the technique of automatic assignment of documents to a fixed number of predefined categories. Each document can be in multiple, exactly one or no category at all. In the approach that uses the Machine Learning techniques for text classification, a general inductive process automatically builds an automatic text classifier by learning, from a set of previously classified documents, the characteristics of the categories of interest.

In text classification, a description $d \in X$ of a document, where X is the document space; and a document fixed set of classes C = {c1, c2,…,cJ} is given. Classes are also called categories or labels. Typically, the document space X is some type of high-dimensional space, and the classes are human defined for the needs of an application. Also, a training set D <d, c>, where <d, c>$\in$ X × C of labeled documents is given.

Using a learning method or learning algorithm, we then wish to learn a classifier or a classification function γ that maps documents to classes:

$$\gamma: X \rightarrow C \text{…….(4.1)}$$

This type of learning is called supervised learning because a supervisor (the learning human who defines the classes and labels training documents) serves as a teacher directing the learning process. The supervised learning method is denoted by Γ and

Γ(D) = γ. The learning method Γ takes the training set D as input and returns the learned classification function γ. [30]

4.1 Types of Classifier Learning Algorithms

Some of the widely used linear classifiers like linear Naive Bayes Classifier, Support Vector Machines(SVMs) etc. have been discussed below:

4.1.1 Naïve Bayes Classifier

The Naïve Bayes is a probabilistic generative training method that learns a model of the distribution P(x,y) from the training sample S. Naïve Bayes is “naïve” in that it assumes

conditional independence between all feature values in a feature vector. Naïve Bayes nevertheless produces fairly accurate classification rules in many cases. The probability of a document d being in class c is computed as given by Equation 4.2:

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c) \quad \text{.....(4.2)}$$

where $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c. $P(t_k|c)$ is the measure of how much evidence t_k contributes that c is the correct class. P(c) is the prior probability of a document occurring in class c.

NB's main strength is its efficiency: Training and classification can be accomplished with one pass over the data. Because it combines efficiency with good accuracy it is often used as a baseline in text classification research. It is often the method of choice if a very large amount of training data is available and there is more to be gained from training on a lot of data than using a better classifier on a smaller training set.

4.1.2 Support Vector Machines (SVMs) Classifier

Improving classifier effectiveness has been an area of intensive machine learning research over the last two decades, and this work has led to a new generation of classifiers, such as support vector machines.

An SVM is a kind of large-margin classifier. It is a vector-space–based machine-learning method where the goal is to find a decision boundary between two classes that is maximally far from any point in the training data. SVM performs classification by constructing an *N*-dimensional hyperplane that optimally separates the data into two categories. SVM models are closely related to neural networks. Using a kernel function, SVM's are an alternative training method for polynomial, radial basis function and multi-layer perceptron classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training.

In the terms of SVM literature, a predictor variable is called an *attribute*, and a transformed attribute that is used to define the hyperplane is called a *feature*. The task of choosing the most suitable representation is known as *feature selection.* A set of features

that describes one case (i.e., a row of predictor values) is called a *vector*. So the goal of SVM modeling is to find the optimal hyperplane that separates clusters of vector in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyperplane are the *support vectors*. Figure 4.1 below presents an overview of the SVM process.

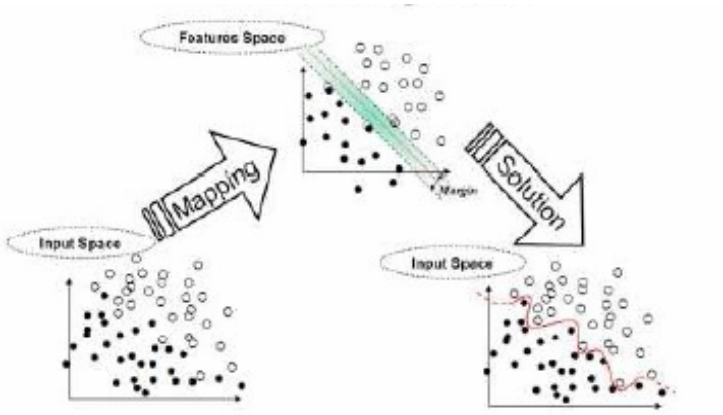

Figure 4.1: SVM algorithm[31]

The SVM in particular defines the criterion to be looking for a decision surface that is maximally far away from any data point. Figure 4.2 shows the margin and support vectors for a sample problem. Other data points play no part in determining the decision surface that is chosen.

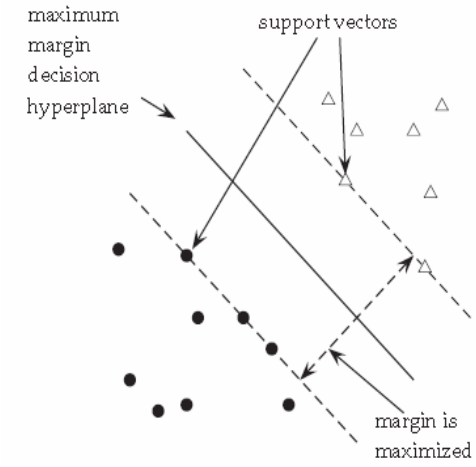

Figure 4.2: The support vectors are the five points right up against the margin of the classifier [15]

Maximizing the margin seems good because points near the decision surface represent very uncertain classification decisions; there is almost a 50% chance of the classifier deciding either way. A classifier with a large margin makes no low-certainty classification decisions. This gives us a classification safety margin: a slight error in measurement or a slight document variation will not cause a misclassification.

4.2 Techniques for Text Classification

Text documents cannot be directly interpreted by a classifier or by a classifier building algorithm. Because of this, an indexing procedure that maps a text dj into a compact representation of its content needs to be uniformly applied to training, validation and test documents.

The fact that certain words often co-occur and are similar to each other can also be used for text classification, and this could be discovered from large collections of text in domain. To achieve classification of text, techniques like Latent Semantic Indexing (LSI) [7], measures of semantic relatedness and similarity which make use of this fact can also be used.

The main advantage of using LSI is that it is useful in solving the problems of *polysemy* (one word can have different meanings) and *synonymy* (different words are used to describe the same concept), which can make classification tasks more difficult. It is based upon the assumption that there is an underlying semantic structure in textual data, and that the relationship between terms and documents can be re-described in this semantic structure form. Categorization systems based on the LSI do not rely on auxiliary structures (thesauri, dictionaries, etc.) and are independent of the native language being categorized.

Similarly, semantic relatedness measures quantify the degree in which some words or concepts are related, considering not only similarity but any possible semantic relationship among them. There are certain semantic measures that use WordNet as their knowledge base to find the similarity, while others like Normalized Google Distance

(NGD), Normalized Compression Distance (NCD) make use of the web as knowledge source.

To get an idea of the performance of a computational measure, Miller Charles word pairs are selected and their similarity values based on human judgment are compared wit those obtained using these measures. The more similar two words are, the higher their probability of being together. These approaches are discussed in brief below.

4.2.1 Approach using Latent Semantic Indexing for text classification

Latent Semantic Indexing (LSI) [32] is a machine-learning model that induces representations of the meaning of words by analyzing the relation between words and passages in large bodies of text. In the context of text classification, Latent Semantic Indexing (LSI), is used to index, analyze, and categorize text documents. It analyzes how terms are spread over the documents of a text corpus and creates a search space with document vectors: similar documents are located near each other in this space and unrelated documents far apart of each other. It is used to analyze the linguistic information. LSI has been used in applied settings with a high degree of success in areas like automatic essay grading and automatic tutoring to improve summarization skills in children. As a model, LSI's most impressive achievements have been in human language acquisition simulations and in modeling of high-level comprehension phenomena like metaphor understanding, causal inferences and judgments of similarity.

LSI was originally developed in the context of information retrieval as a way of overcoming problems with polysemy and synonymy that occurred with vector space model (VSM) [18] approaches. The method used by LSI to capture the essential semantic information is dimension reduction, selecting the most important dimensions from a co occurrence matrix decomposed using Singular Value Decomposition. It is an information retrieval technique that locates linguistic topics in a set of documents [7]. LSI is applied to compute the linguistic similarity between source artifacts (e.g. packages, classes or methods) and cluster them according to their similarity. This clustering partitions the system into linguistic topics that represent groups of documents using similar vocabulary. It has been shown in [32] that LSI addresses the synonyms very well. With simple corpus

training, LSI managed to answer correctly 64% of the synonyms questions in the Test of English as a Foreign Language, better than the average student. There is a wide range of applications of LSI, such as automatic assignment of reviewers to submitted conference papers [33], cross-language search engines, spell checkers and many more. In the field of software engineering LSI has been successfully applied to categorized source files [34] and open-source projects [35], recover links between external documentation and source code [36]. LSI produces measures of word-word, word-passage and passage-passage relations that are well correlated with several human cognitive phenomena involving association or semantic similarity. LSI allows to closely approximate human judgments of meaning similarity between words and to objectively predict the consequences of overall word-based similarity between passages, estimates of which often figure prominently in research on discourse processing. Furthermore LSI has proved useful in psychology to simulate language understanding of the human brain, including processes such as the language acquisition of children.

4.2.1.1 Term-Document Matrix

Like other information retrieval (IR) techniques, Latent Semantic Indexing (LSI) is based on the vector space model (VSM) approach. This approach models documents as bag-of-words and arranges them in a Term-Document Matrix A, such that $a_{i,j}$ equals the number of times term t_i occurs in document d_j. As discussed earlier, LSI can overcome problems with synonymy and polysemy that used to occur in prior vectorial approaches, and thus improve the basic vector space model by replacing the original term-document matrix with an approximation. This is done using singular value decomposition (SVD), a principal components analysis (PCA) technique originally used in signal processing to reduce noise while preserving the original signal. The measure of similarity computed in the reduced dimensional space is usually, but not always, the cosine between vectors.

Assuming that the original term-document matrix is noisy (the synonymy and polysemy), the approximation is interpreted as a noise reduced and thus a better model of the text corpus.

For example, a typical search engine covers a text corpus with millions of web pages,

containing some ten thousands of terms, which is reduced to a vector space with 200-500 dimensions only. In Software Analysis, the number of documents is much smaller and the text corpus is reduced to 20-50 dimensions.

The entire corpus can be represented as a term-document matrix D. D consists of document vectors d_i. The document vectors are the columns of this matrix. Document vector d_i contains m generalized term.

Since the terms might not be of the same type (i.e. one term may be a term count, one may be some relative quantity) it no longer makes sense to normalize across terms. So, it is assumed that the columns of D are not normalized document vectors, because they are considered as generalized terms. Earlier the document vectors were considered individually. The question of how the documents in the corpus relate to each other is not considered, except implicitly with respect to the query and in consideration of the IDF (Inverse Document Frequency). The idea is to get a better representation of the documents that corresponds to the structure of the corpus, since a better representation may help us do better retrieval.

Inverse document frequency (IDF) considers the whole document corpus. IDF looks at the distribution of the terms in the whole corpus and shrinks term frequencies for terms that occur a lot in the corpus. IDF thus uses the overall corpus characteristics to alter our document representation.

Figure 4.3 schematically represents the LSI process. The document collection is modeled as a vector space. Each document is represented by the vector of its term occurrences, where terms are words appearing in the document. The term-document-matrix A is a sparse matrix and represents the document vectors on the rows. This matrix is of size $n \times m$, where m is the number of documents and n the total number of terms over all documents. Each entry $a_{i,j}$ is the frequency of term t_i in document d_j . A geometric interpretation of the term-document-matrix is a set of document vectors occupying a vector space spanned by the terms. The similarity between documents is typically defined as the cosine or inner product between the corresponding vectors. Two documents are considered similar if their corresponding vectors point in the same direction.

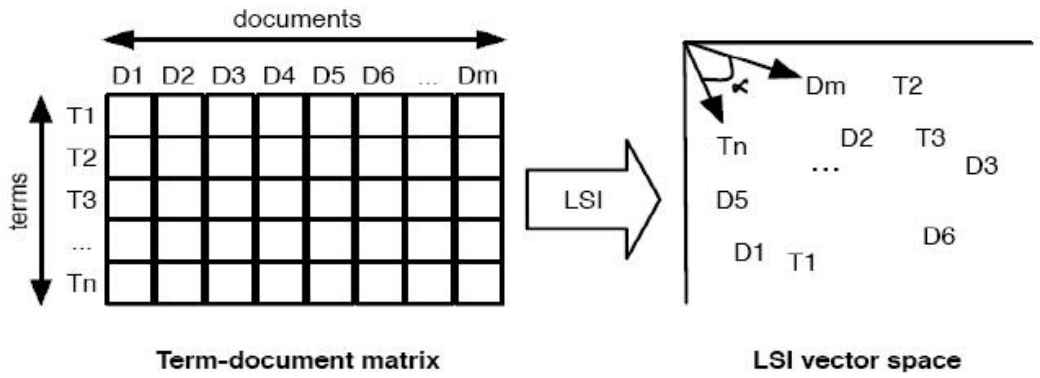

Figure 4.3: LSI takes as input a set of documents and the terms occurrences, and returns as output a vector space containing all the terms and all the documents. The similarity between two items (i.e., terms or documents) is given by the angle between their corresponding vectors [37].

4.2.1.2 Data Normalization

All words are not equally significant as some words are more likely to discriminate documents than others. Words with a high frequency are considered to be too common and those with low frequency too rare, and therefore both of them are not contributing significantly to the content of a document. Words with medium frequency have the highest ability to discriminate content. Thus all words above an upper and all words below a lower threshold are excluded from the term list.

The removal of high frequency words is usually done excluding a list of common words, called stopwords. As the distribution of word frequencies follows the power law, this removal reduces the size of a text corpus by about 30 to 50 percent. In case of an English text corpus, the SMART stopword list is well-established: it contains about 500 common non-discriminative words like the, of, to, a, is [38]. The removal of low frequency words is usually done by excluding all words that occur in only one document.

Furthermore the same term may appear in different grammatical inflections. Most words are composed of a stem, which bears the meaning of the word, and a suffix that bears grammatical information. If two words have the same stem then they refer to the same

concept and should be indexed as one term. The process of removing the grammatics is called stemming, and a standard approach is to have a list of suffixes and to remove the longest possible one. For example, train, trained and training are all reduced the common stem train. In case of an English text corpus the Porter Stemming Algorithm is well established.

Research in information retrieval (IR) has shown that normalization leads to more effective retrieval than if the raw term frequencies were used [39].

4.2.3 Term Frequency and Weighting

The goal of a weighting function is to balance out very rare and very common terms. Typically a combination of two weighting schemes is applied, a local weighting and a global weighting. The first puts a term occurrence in relation to its document, and the latter in relation to the whole text corpus. IR systems assign weights to terms by considering:

1. Local information from individual documents
2. Global information from collection of documents

In addition, systems that assign weights to link graph information to properly account for the degree of connectivity between documents. This weighting scheme is given by the equation:

$$\text{Term Weight} = \quad w_i = tf_i * log\left(\frac{D}{df_i}\right) \quad \text{..... (4.3)}$$

where:

- tf_i = term frequency (term counts) or number of times a term i occurs in a document.
- df_i = document frequency or number of documents containing term i
- D = number of documents in the database

Local Weights

Equation 4.3 shows that w_i increases with tf_i. This makes the model vulnerable to term repetition. Given a query q,

1. For documents of equal lengths, those with more instances of q are favoured during retrieval.
2. For documents of different lengths, long documents are favoured during retrieval since these tend to contain more instances of q.

Global Weights

In Equation 4.3, the log(D/dfi) term is known as the inverse document frequency (IDFi) – a measure of the shear volume of information associated to a term i within a set of documents. Inspecting the dfi/D ratio, this is the probability of retrieving from D a document containing term i. In equation, we simply invert this probability and take its log. The result is then premultiplied by tfi. Several modifications to the equation have been proposed and more general form of this equation is given below:

$$x_{i,j} = local(t_i, d_j) \times global(t_i) \quad \text{.....(4.4)}$$

Equation 4.6 shows that wi decreases as dfi increases. For example, if in a 1000-document database only 10 documents contain a particular term, the IDF for this term is log(1000/10) = 2. However, if only one document contains that term, then IDF is log(1000/1) = 3. Thus, terms which appear in too many documents (e.g. stopwords, very frequent terms) receive a low weight, while uncommon terms which appear in few documents receive a high weight. This makes sense since too common terms are not very useful for distinguishing a relevant document from a non-relevant one. Terms with acceptable weights are those that are not too common or too rare; i.e. their term vectors are not too far or too close to the query vector.

When applied on textual data LSI achieves best results with the entropy weighting. Nevertheless we present here the tf-idf -weighting as it is more popular in both information retrieval and software analysis.

4.2.4 Singular Value Decomposition

LSI starts with an input as term-document-matrix, weighted by a weighting function to balance out very rare and very common terms. SVD is used to break down the vector space model into less dimensions. This algorithm preserves as much information as

possible about the relative distances between the document vectors, while collapsing them into a much smaller set of dimensions.

SVD decomposes matrix A into its singular values and its singular vectors, and yields– when truncated at the k largest singular values – an approximation A` of A with rank k. Furthermore, not only the low-rank term-document matrix A` can be computed but also a term-term matrix and a document-document matrix. Thus, LSI allows us to compute term-document, term-term and document-document similarities.

$$A = U \times S \times V^T$$

Singular value decomposition transforms the matrix A into three matrices, using Eigenvalue decomposition. This yields three matrices: $U \times S \times V^T$. The two outer matrices contain the singular vectors: U is the term matrix, it contains the left singular vectors and each of its row corresponds to a term. The same goes for V, the document matrix, which contains the right singular vectors and whose rows correspond to documents. The middle matrix S is a diagonal matrix with the singular values, which are a kind of eigenvalue, of A in descending order.

When multiplying all three matrices together to reconstruct the original matrix, the singular values act as weights of the singular vectors. A singular vector with a large corresponding singular vectors has a large impact on the reconstruction, while a small value indicates a singular vector with almost no impact on the result. Thus small values and their vectors may be discarded without affecting the result noticeably. Keeping the k largest singular values only yields a low-rank approximation of A, which is the best rank k approximation of A under the least-square-error criterion:

$$A`_{nxm} = U_{nxk} \times S_{kxk} \times V^T_{kxm}$$

As the rank is the number of linear-independent rows and columns of a matrix, the vector space spanned by A` is of dimension k only and much less complex than the initial space. Applying SVD on natural data (such as signals, images or text documents) yields a distribution of singular values that follows the power law: a few large values, and a long tail with very small values. When used for information retrieval, k is typically about 200-500, while n and m may go into millions. That is why a text corpus with millions of documents can be approximated with such a low-ranked matrix. When used to analyze software on the other hand, k is typically about 20−50 with vocabulary and documents in

the range of thousands only, and since A` is the best approximation of A under the least square- error criterion, the similarity between documents is preserved, while in the same time mapping semantically related terms on one axis of the reduced vector space and thus taking into account synonymy and polysemy. In other words, the initial term-document matrix A is a table with term occurrences and by breaking it down to much less dimension the latent meaning must appear in A` since there is now much less space to encode the same information. Meaningless occurrence data is transformed into meaningful concept information.

4.2.5 Term and Documents Similarity

To show the SVD factors geometrically, the rows of the matrices are taken as coordinates of points representing the documents and terms vector dimensional space. The nearer one points to the other, if they are more similar documents or terms.

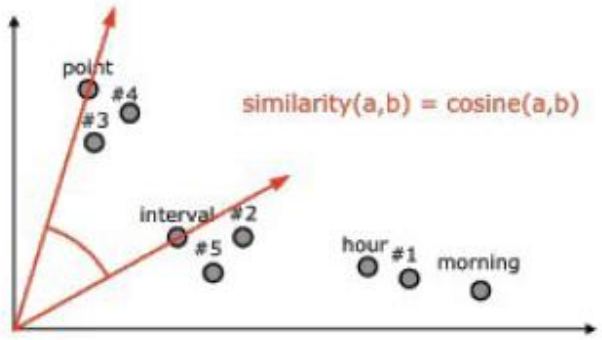

Fig 4.4: On the left: An LSI-Space with terms and documents, similar elements are placed near each other. [37]

Similarity is typically defined as the cosine between the corresponding vectors:

$$sim(d_i, d_j) = \cos(v_i, v_j)$$

Computing the similarity between document di and dj is done taking the cosine between the i-th and j-th row of the matrix.

The resulting cosine value, similarity values range from 1 to 0: 1 for similar vectors with the same direction and to 0 for dissimilar, orthogonal vectors. Theoretically cosine values can go all the way to −1, but because there are no negative term occurrences, similarity values never goes below to zero.

Though the use of LSI for text classification has advantages like its independence of the use different languages, and its significant degree of inherent noise immunity with regard to errors in the documents being processed, but it suffers from certain problems like:

- The SVD algorithm is $O(N^2k^3)$, where N is the number of terms plus documents, and k is the number of dimensions in the concept space. Typically, k will be small, ranging anywhere from 50 to 350. However N grows rapidly as the number of terms and the number of documents increase. This makes the SVD algorithm unfeasible for a large, dynamic collection.
- Determining the optimal number of dimensions in the concept space to obtain the best results for information retrieval.
- Another problem with LSI is that the concept space is not understandable by humans. This makes LSI difficult to debug and difficult to understand.

Considering these problems associated with LSI, alternative methods for text classification are explored. These include finding similarity between word pairs. The values obtained are then compared with the standard similarity values of the Miller Charles word pairs.

4.3 Measures of semantic relatedness and similarity

Budanitsky and Hirst [14] describe *semantic relatedness* as follows:

Recent research on the topic in computational linguistics has emphasized the perspective of semantic relatedness of two lexemes in a lexical resource, or its inverse, semantic distance. It's important to note that semantic relatedness is a more general concept than similarity; similar entities are usually assumed to be related by virtue of their likeness (bank–trust company), but dissimilar entities may also be semantically related by lexical relationships such as *meronymy* (car–wheel) and *antonymy* (hot–cold), or just by any kind of functional relationship or frequent *association* (pencil–paper, penguin–Antarctica).

Determining the semantic relatedness between two words refers to computing a statistical measure of similarity between those words. Semantic Measures play an important role in text classification. Measures of Semantic Relatedness (MSRs) are statistical methods for extracting word associations from text corpora. Semantic relatedness measures quantify the degree in which some words or concepts are related, considering not only similarity

but any possible semantic relationship among them. Many semantic measures have been proposed in the past to compute degrees of relatedness among words, texts or concepts.

4.3.1 Desirable features of semantic relatedness measures:

As discussed in [11], the characteristics that are desirable for a semantic measure are:-

1. *Domain independence*: Nowadays, an increasing amount of online ontologies and semantic data is available on the Web, thus, enabling a new generation of semantic applications. In developing that kind of domain independent applications, one has to deal with this increasing heterogeneity, without establishing in advance the ontologies to be accessed.

2. *Universality*: Semantic measures, in the highly dynamic context of the Web, must be flexible and general enough to be used independently of their final purpose, and without relying on specific lexical resources or knowledge representation languages.

3. *Maximum* coverage: It is assumed that, in the context of web applications with no predefined domain, *maximum coverage* of possible interpretations of the words must be warranted.

4.4 Classification based on source of knowledge utilized

Semantic measures can also be defined between lexically expressed word senses, or between whole texts. Manually compiled taxonomies such as WordNet and large text can be used for finding similarity between words. Another method can be, regarding the Web as a live corpus and using it to find semantic relatedness measure between words.

4.4.1 Using WordNet as a knowledge base

All approaches to measuring semantic relatedness that use a lexical resource construe the resource, in one way or another, as a network or directed graph, and then base the measure of relatedness on properties of paths in this graph. Most of the methods use WordNet, a broad coverage lexical network of English words. The lexical database WordNet is particularly well suited for similarity measures, since it organizes nouns and verbs into hierarchies of *is–a* relations. Nouns, verbs, adjectives, and adverbs are each

organized into networks of synonym sets (*synsets*) that each represent one underlying lexical concept and are interlinked with a variety of relations.

WordNet is based on the following definitions and notations:

- The *length* of the shortest path in WordNet from synset ci to synset cj (measured in edges or nodes) is denoted by len(ci, cj).
- The *depth* of a node is the length of the path to it from the global root, i.e., depth(ci = len(root, ci).
- The *lowest super-ordinate* (or most specific common subsumer) of c1 and c2 is denoted by lso(c1, c2).
- Given any formula *rel(c1, c2)* for semantic relatedness between two concepts c1 and c2, the relatedness rel(w1,w2) between two words w1 and w2 is :

$$\mathrm{rel}(w_1, w_2) = \max_{c_1 \in s(w_1), c_2 \in s(w_2)} [\mathrm{rel}(c_1, c_2)]$$

where $s(w_i)$ is "the set of concepts in the taxonomy that are senses of word w_i" . That is, the relatedness of two words is equal to that of the most-related pair of concepts that they denote.

Given taxonomy of concepts, a straightforward method to calculate similarity between two words (concepts) is to find the length of the shortest path connecting the two words in the taxonomy. If a word is polysemous then multiple paths might exist between the two words. In such cases, only the shortest path between any two senses of the words is considered for calculating similarity. A problem that is frequently acknowledged with this approach is that it relies on the notion that all links in the taxonomy represent a uniform distance.

WordNet::Similarity [40] makes it possible to measure the semantic similarity or relatedness between a pair of concepts (or word senses). It provides six measures of similarity, and three measures of relatedness, all of which are based on the lexical database WordNet. These measures are implemented as Perl modules which take as input two concepts, and return a numeric value that represents the degree to which they are similar or related.

Three similarity measures are based on path lengths between *concepts*: *lch* (Leacock & Chodrow) [22], *wup* (Wu &Palmer) [41], and *path*. The lch measure finds the shortest

path between two concepts, and scales that value by the maximum path length in the is–a hierarchy in which they occur. wup finds the path length to the root node from the least common subsumer (LCS) of the two concepts, which is the most specific concept they share as an ancestor. This value is scaled by the sum of the path lengths from the individual concepts to the root. The measure path is equal to the inverse of the shortest path length between two concepts.

The three remaining similarity measures are based on *information content*, which is a corpus–based measure of the specificity a concept. These measures include *jcn* (Jiang & Conrath) [23], *lin* (Lin) [24], *res* (Resnik) [25]. The *lin* and *jcn* measures augment the information content of the LCS of two concepts with the sum of the information content of the individual concepts. The *lin* measure scales the information content of the LCS by this sum, while jcn subtracts the information content of the LCS from this sum (and then takes the inverse to convert it from a distance to a similarity measure).

4.4.2 Using Wikipedia as a knowledge base

The advent of Wikipedia in 2001 has fulfilled the need for a more comprehensive knowledge base. Many techniques that use Wikipedia to compute semantic relatedness have been developed in the recent years. Some recent research efforts have focused on using Wikipedia to improve coverage with respect to traditional thesauri-based methods. Nowadays Wikipedia is rapidly growing in size, and it is not difficult to find new terms and named entities on it. Some classic measures are adapted to use Wikipedia instead of WordNet as knowledge source, showing promising results. A further step in using Wikipedia is found in [9]. They propose a method to represent the meaning of texts or words as weighted vectors of Wikipedia-based concepts, using machine learning techniques.

Wikipedia can also be used to represent the meaning of texts or words as weighted vectors of Wikipedia-based concepts, using machine learning techniques.

Although they have clear benefits, Wikipedia is still not comparable with the whole Web in the task of discovering and evaluation of implicit relationships because all the words can still not be found in Wikipedia.

4.4.3 Measures based on Web

Semantic relatedness between the words is also defined using the Web. Rudi L and et al. [12] developed the method that defines the relatedness between the words via Google Similarity Distance. They use the World Wide Web as the database, and Google as the search engine. Some of these measures are NGD, NCD etc.

Normalized Google Distance (NGD)

Every text corpus or particular user combined with a frequency extractor defines its own relative frequencies of words and pharses usage. In the World-Wide-Web and Google setting there are millions of users and text corpora, each with its own distribution. The Google distribution is universal for all the individual web users distributions. The number of web pages currently indexed by Google is very large. Every common search term occurs in millions of web pages. This number is so vast, and the number of web authors generating web pages is so enormous (and can be assumed to be a truly representative very large sample from humankind), that the probabilities of Google search terms, conceived as the frequencies of page counts returned by Google divided by the number of pages indexed by Google, approximate the actual relative frequencies of those search terms as actually used in society.

Google distance is a measure of semantic interrelatedness derived from the number of hits returned by the Google search engine for a given set of keywords. Keywords with the same or similar meanings in a natural language sense tend to be "close" in units of Google distance, while words with dissimilar meanings tend to be farther apart.

Mathematically, the *normalized Google distance* [29] between two search terms *x* and *y* is:

$$\mathrm{NGD}(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log M - \min\{\log f(x), \log f(y)\}} \qquad \text{……..(4.5)}$$

where *M* is the total number of web pages searched by Google; *f*(*x*) and *f*(*y*) are the number of hits for search terms *x* and *y*, respectively; and *f*(*x*, *y*) is the number of web pages on which both *x* and *y* occur. If the two search terms *x* and *y* never occur together

on the same web page, but do occur separately, the normalized Google distance between them is infinite. If both terms always occur together, their NGD is zero.

The same method may be used with other text corpora like the King James version of the Bible or the Oxford English Dictionary and frequency count extractors, or the world-wide-web again and Yahoo as frequency count extractor.

This is experimentally evidenced by the fact that when Google doubled its size the sample semantics of rider, horse stayed the same. Determining the NGD between two Google search terms does not involve analysis of particular features or specific background knowledge of the problem area. Instead, it analyzes all features automatically through Google searches of the most general background knowledge

database: the World Wide Web.

Normalized Semantic Score (NSS)

Normalized Semantic Score (NSS) is an MSR that is derived from NGD. To be more precise, the relatedness between two words x and y is derived as follows:

$$NSS(x, y) = 1 - NGD(x, y) \quad \text{(4.6)}$$

The value of NSS varies between 0 and 1 and more is the value, closer is the association of a word to the respective category.

Normalized Compression Distance (NCD)

The NCD between two binary strings is defined in terms of compressed sizes of the two strings and of their concatenation; it is designed to be an effective approximation of the non computable but universal Kolmogorov distance between two strings. The Normalized Compression Distance NCD(x, y) between two binary strings x and y

is given mathematically by Equation 4.7 as :

$$NCD(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}. \quad \text{(4.7)}$$

The lower the value of NCD is for a particular word-pair, the more related the two words are.

Chapter 5

IMPLEMENTATION & RESULTS

5.1 Implementation of Latent Semantic Indexing (LSI)

LSI can be implemented for four main applications which are as follows:

1. *Nearest neighbor*: Returns a list of the terms in the LSI space that are most similar to a target term and the corresponding cosines to a given term.

2. *Matrix comparison*: Compares a set of texts or terms against each other.

3. *Sentence comparison*: Submits a sequence of texts and receive the cosines between each adjacent pair (used for calculating textual coherence).

4. *One to many comparison*: Compares a target text against a set of texts (used for vocabulary testing and essay grading).

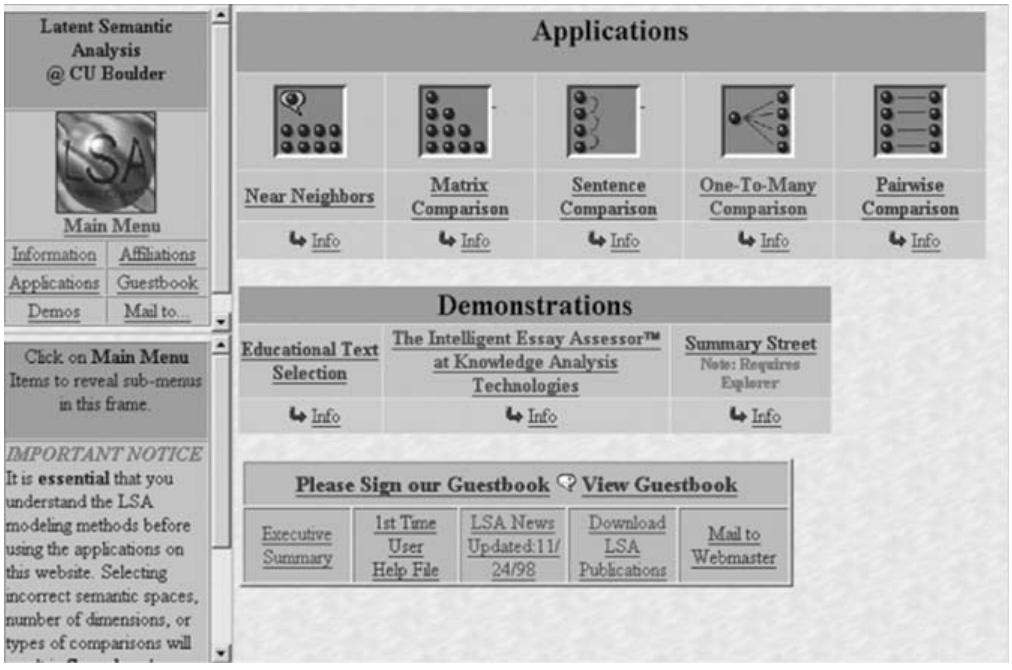

Figure 5.1: The home page of the LSI Web site

As with all of the applications, a suitable semantic space and the number of factors to employ (the maximum number of factors available will be used if this field is left blank) have to be selected. Implementation of LSI applications has been done through http://lsa.colorado.edu as shown in Figure 5.1. All the applications provided by this site are implemented on a corpus known as the 'General Reading Corpora'.

5.1.1 The Nearest Neighbor Application

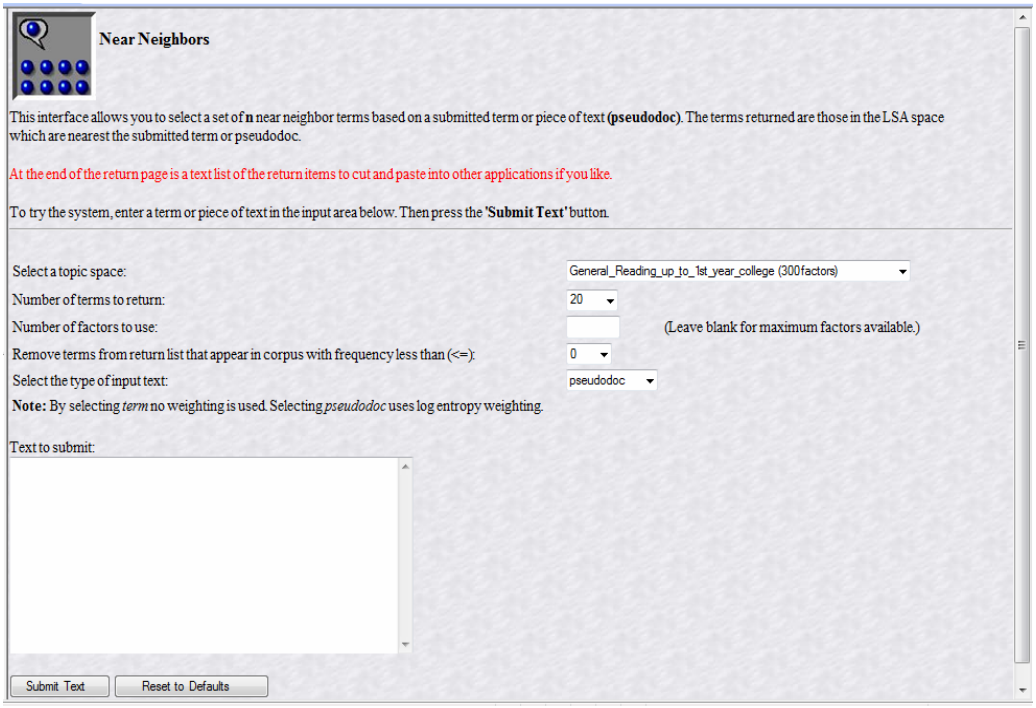

Figure 5.2: The nearest neighbor tool

LSI Similarity	Term
0.75	moop
0.74	barking
0.99	dog
0.86	barked
0.86	dogs
0.81	wagging
0.80	collie
0.79	leash
0.75	manilak
0.75	moops

Table 5.1: Results From the Nearest Neighbor Application for the Term ”Dog” With the Frequeny Cutoff Set to 10

5.1.2 Matrix Comparison Application

The only parameter that needs to be set other than the space and number of factors parameters while using this application is the kind of comparison—“term to term” or “document to document.” As in the nearest neighbor application, this parameter controls whether log entropy weighting is used. When “term to term” is set, no weighting is used. When “document to document” is set, all items are log entropy weighted.

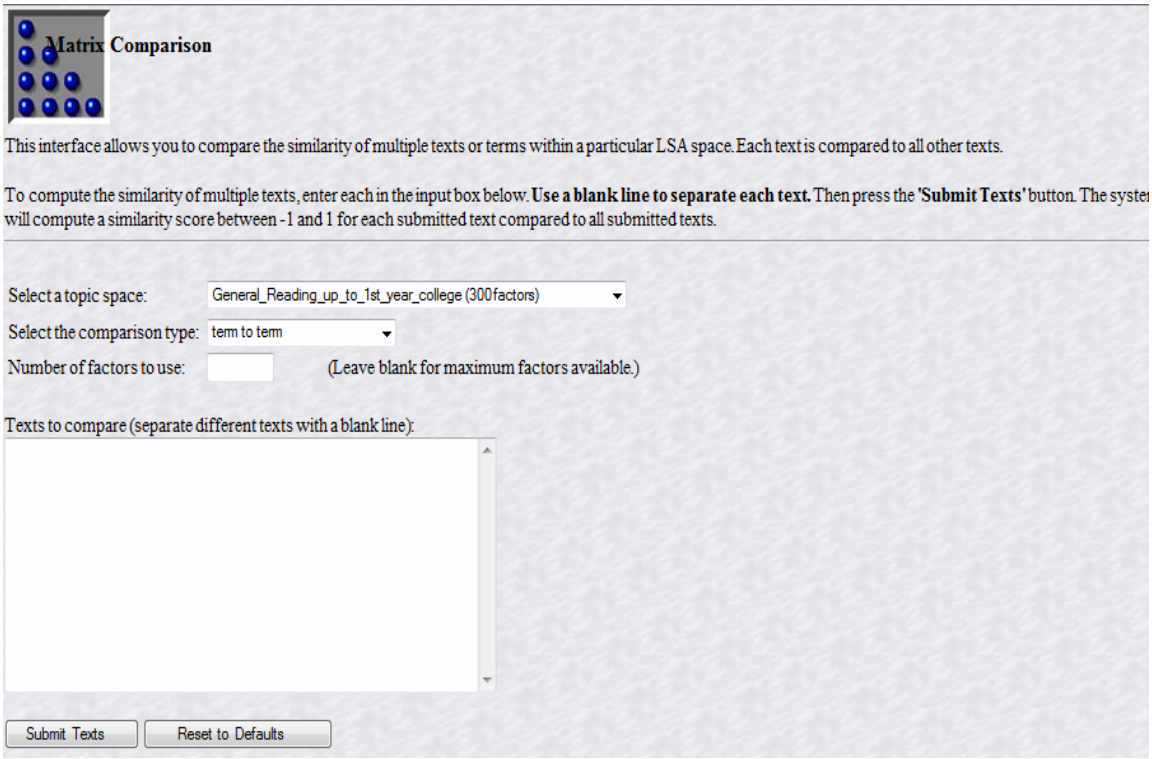

Figure 5.3: The matrix comparison tool

The submitted texts’ similarity matrix (in term space) is as shown below in Table 5.2:

Document	dog	cat	puppy	kitten
Dog	1	0.36	0.76	0.28
Cat	0.36	1	0.38	0.61
Puppy	0.76	0.38	1	0.43
Kitten	0.28	0.61	0.43	1

Table 5.2: Results from the Matrix Application for the Terms"Dog," "Cat," "Puppy," and"Kitten"

5.1.3 Sentence Comparison Application

The text coherence is found that by taking the mean of the cosines of the LSI vectors representing successive sentences in a text. The application indicates the cosine between each successive pair of sentences and provides the mean and standard deviation of these values.

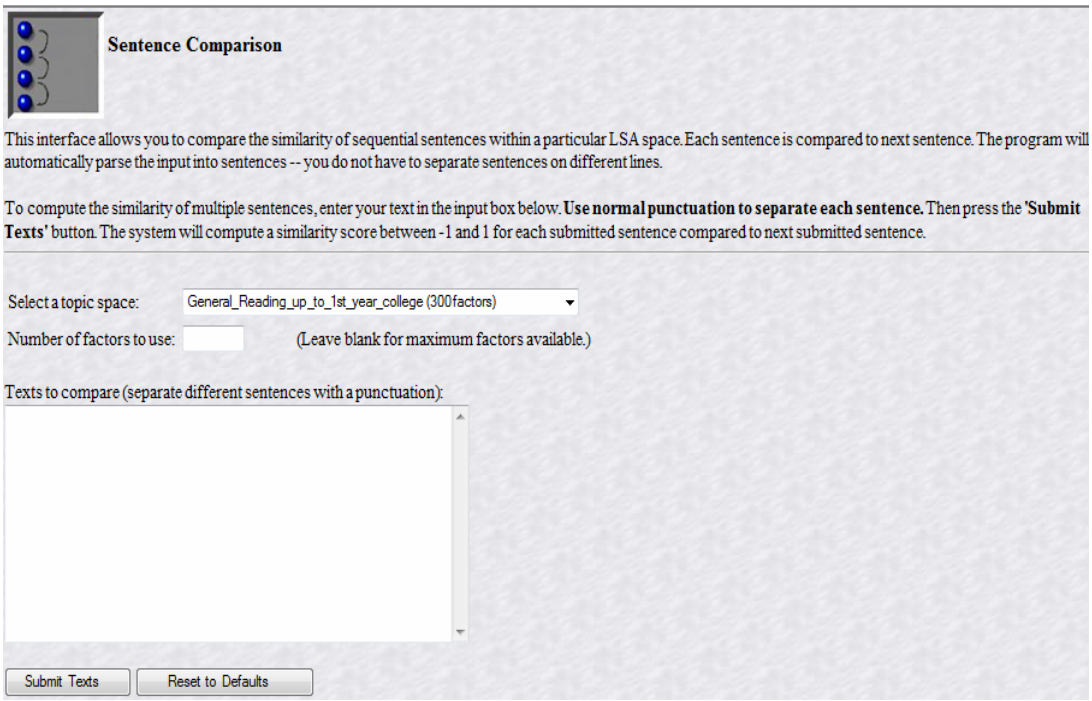

Figure 5.4 Sentence Comparison tool

COS	SENTENCES
0.32	*1:*
0.40	*2:* The largest dog bit all of the cats 2.

0.31	*3:* There are at least three cats 3.
0.75	*4:* A pet dog should also be washed several times 4.
	5: The quick brown fox jumped over the lazy dog

Mean of the Sentence to Sentence Coherence is: 0.44

Standard deviation of the Sentence to Sentence is: 0.18

Table 5.3: Sentence to Sentence Coherence Comparison Results

5.2.4 One to Many Comparison Application

The one to many application takes a single text and compares it to a number of other texts. The tool allows to generate vector lengths (Table 5.4). Vector lengths give an indication of the amount of information that is encoded by a text and they can be useful in determining the importance of individual terms or subtexts to the meaning vector associated with a larger text.

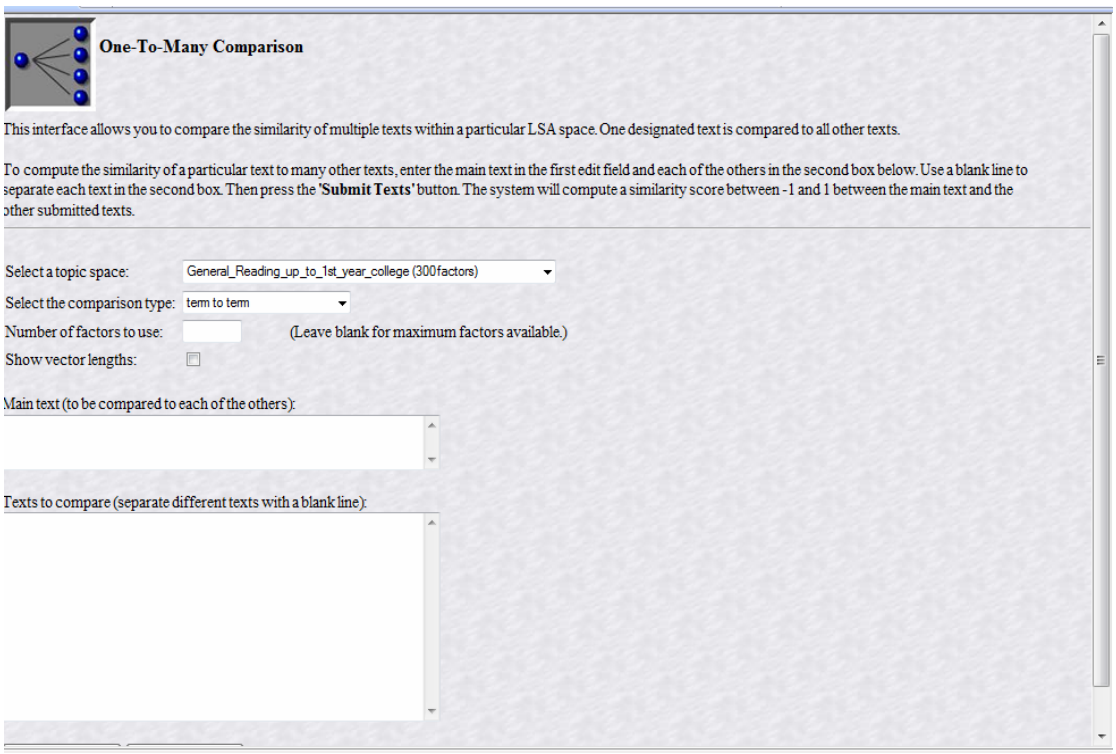

Figure 5.5: The one to many tool.

Texts	Vector Length
dog	3.36
cat	1.88
puppy	0.70
kitten	0.46

Table 5.4: Results from one to many application

5.2 Computing Measures of Similarity and Semantic Relatedness for text classification

Measures of similarity quantify how much two concepts are alike. The lexical database WordNet is particularly well suited for similarity measures.

WordNet::Similarity makes it possible to measure the semantic similarity or relatedness between a pair of concepts (or word senses). It provides six measures of similarity, and three measures of relatedness, all of which are based on the lexical database WordNet. These measures are implemented as Perl modules which take as input two concepts, and return a numeric value that represents the degree to which they are similar or related. As discussed above, the three similarity measures are based on path lengths between concepts: *lch* (Leacock & Chodorow) [22], *wup* (Wu & Palmer 1994), and *path*. The three remaining similarity measures are based on information content, which is a corpus–based measure of the specificity of a concept. These measures include *jcn* (Jiang & Conrath) [23], *lin* (Lin) [24], *res* (Resnik) [25]. These measures are computed for the various *Miller Charles* word pairs and results are compared with the values obtained by human judgement.

The output of semantic relatedness for a Miller Charles word pair (glass-magician) using WordNet::Similarity is as shown below in Figure: 5.6

Figure 5.6: Results of semantic relatedness for the word pair glass magician

Besides calculating similarity measures using WordNet as a database, relatedness is also calculated using other relatedness measures like Normalized Google Distance (NGD), Normalized Compression Distance (NCD) etc. which use the web as a database source. The results obtained are shown in Table 5.5

5.3 Results

Word Pair	Miller-Charles	NCD	NSS	NGD= 1-NSS	WordNet		LSA
					Leacock & Chodrow	Lin	
Cord-smile	0.13	0.2	0.79	0.21	1.291	0	0.015
rooster - voyage	0.08	0.26	0.781	0.219	0.51	0	0.005
noon - string	0.08	0.23	0.741	0.259	1.204	0.092	0.078
glass - magician	0.11	0.285	0.787	0.213	1.61	0.142	0.16
Monk - slave	0.55	0.2	0.975	0.025	2.079	0.201	-0.011
coast - forest	0.42	0.23	0	1	1.897	0.118	0.107
monk - oracle	1.1	0.23	0.421	0.579	1.6094	0.182	0.054
lad - wizard	0.42	0.23	0.641	0.359	2.079	0.224	0.167
forest - graveyard	0.84	0.31	0.737	0.263	1.491	0.112	-0.026
food - rooster	0.89	0.26	0.627	0.373	0.916	0.076	0.033

coast - hill	0.87	0.2	0	1	0.2	0.728	0.036
car - journey	1.16	0.26	0.157	0.843	0.798	0	0.099
crane - implement	1.68	0.31	0.201	0.799	2.079	0.332	-0.046
brother - lad	1.66	0.26	0.85	0.15	2.079	0.24	0.214
bird - crane	2.97	0.2	0.69	0.31	2.302	0	0.265
bird - cock	3.05	0.16	0.825	0.175	2.995	0.778	0.326
Food - fruit	3.08	0.2	0.403	0.597	1.386	0.155	0.363
Brother - monk	2.82	0.26	0.9	0.1	2.995	0.207	0.016
asylum - madhouse	3.61	0.28	0.579	0.421	2.995	0.981	0.113
furnace - stove	3.11	0.26	0.606	0.394	1.386	0.229	0.294
magician - wizard	3.5	0.28	0.622	0.378	3.688	1	0.224
Journey - voyage	3.84	0.26	1	0	2.995	0.827	0.413
coast - shore	3.7	0.2	0.641	0.359	2.995	0.963	0.392
implement - tool	2.95	0.31	0.199	0.801	2.995	0.914	0.127
Boy - lad	3.76	0.13	0.801	0.199	2.995	0.797	0.473
automobile - car	3.92	0.33	0.261	0.739	3.689	1	0.572
midday - noon	3.42	0.23	0.876	0.124	3.689	1	0.534
gem - jewel	3.84	0.2	0.819	0.181	3.689	1	0.211

Table 5.5: Semantic relatedness and similarity values obtained using various measures.

It can be noticed from Table 5.5 that the semantic relatedness between these pairs of words for both Miller Charles (based on human judgement) and Leacock & Chodrow measure lie in a scale from 0.0 to 4.0 (from no relatedness at all, to identical or strongly related words). On the other hand, the values of all other MSRs lie between 0.0 & 1.0. Therefore, both the Miller Charles and Leacock & Chodrow similarity values should be normalised to make them lie in the range from 0.0 to 1.0. This is achieved using the following mathematical formula:

Suppose the dissimilarity index is in the range of [d^{min} , d^{max}] and is not in the range of [0, 1]. It is to be transformed into the range of [0, 1]. Let d be the original dissimilarity and δ the normalized dissimilarity. Then the normalised value is given by Equation 5.1:

$$\delta = \frac{d - d^{min}}{d^{max} - d^{min}} \quad \text{.....(5.1)}$$

In our case:

d = original value of dissimilarity

d^{max} = upper limit of the range= 4

d^{min} = lower limit of the range= 0

The resulting values of similarity and semantic relatedness after normalization are given in Table 5.6.

Word Pair	Normalised Miller Charles values	WordNet		NCD	NGD	LSI
		Lin	Normalised Leacock & Chodrow values			
cord-smile	0.032	0	0.322	0.2	0.21	0.015
rooster - voyage	0.02	0	0.127	0.26	0.219	0.005
noon - string	0.02	0.092	0.301	0.23	0.259	0.078
glass - magician	0.027	0.142	0.402	0.285	0.213	0.16
monk - slave	0.137	0.201	0.52	0.2	0.025	-0.011
coast - forest	0.105	0.118	0.474	0.23	1	0.107
monk - oracle	0.275	0.182	0.402	0.23	0.579	0.054
lad - wizard	0.105	0.224	0.519	0.23	0.359	0.167
forest - graveyard	0.21	0.112	0.372	0.31	0.263	-0.026
food - rooster	0.222	0.076	0.229	0.26	0.373	0.033
coast - hill	0.217	0.728	0.05	0.2	1	0.036
car - journey	0.29	0	0.199	0.26	0.843	0.099
crane - implement	0.42	0.332	0.519	0.31	0.799	-0.046
brother - lad	0.415	0.24	0.519	0.26	0.15	0.214

bird - crane	0.742	0	0.575	0.2	0.31	0.265
bird - cock	0.762	0.778	0.748	0.16	0.175	0.326
food - fruit	0.77	0.155	0.346	0.2	0.597	0.363
brother - monk	0.705	0.207	0.748	0.26	0.1	0.016
asylum - madhouse	0.902	0.981	0.748	0.28	0.421	0.113
furnace - stove	0.777	0.229	0.346	0.26	0.394	0.294
magician - wizard	0.875	1	0.922	0.28	0.378	0.224
journey - voyage	0.96	0.827	0.748	0.26	0	0.413
coast - shore	0.925	0.963	0.748	0.2	0.359	0.392
implement - tool	0.737	0.914	0.748	0.31	0.801	0.127
boy - lad	0.94	0.797	0.748	0.13	0.199	0.473
automobile - car	0.98	1	0.922	0.33	0.739	0.572
midday - noon	0.855	1	0.922	0.23	0.124	0.534
gem - jewel	0.96	1	0.922	0.2	0.181	0.211

Table 5.6: Values obtained after Normalization of Miller Charles and Leacock & Chodrow similarity values

To evaluate the performance of these measures, the obtained values are plotted graphically and. The graphs are shown in Figures 5.7, 5.8, 5.9, 5.10, 5.11, 5.12.

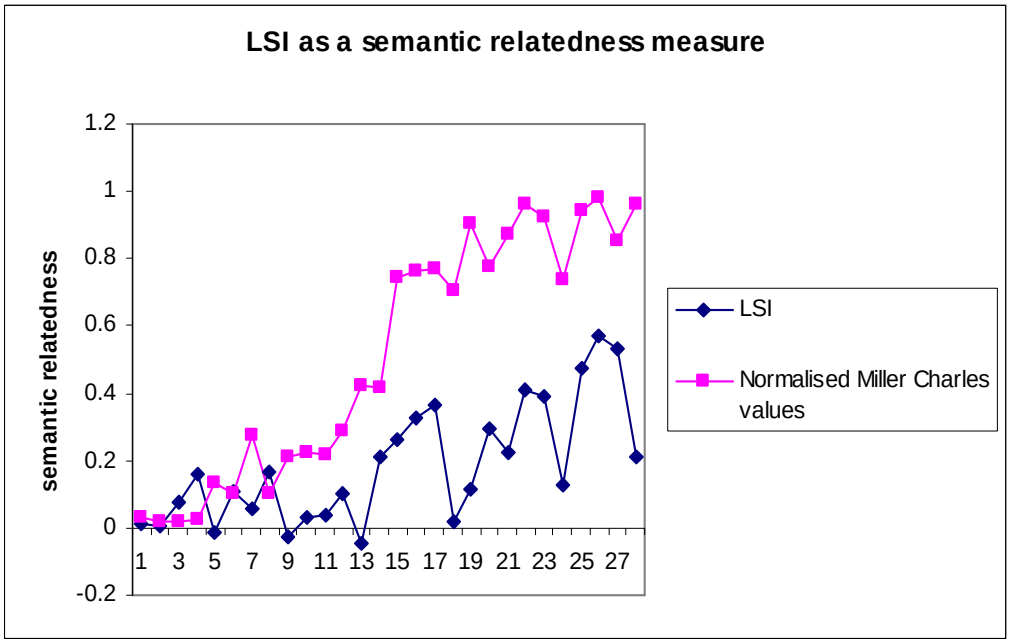

Figure 5.7: Plot of semantic relatedness values using LSI against standard values for Miller Charles word pairs

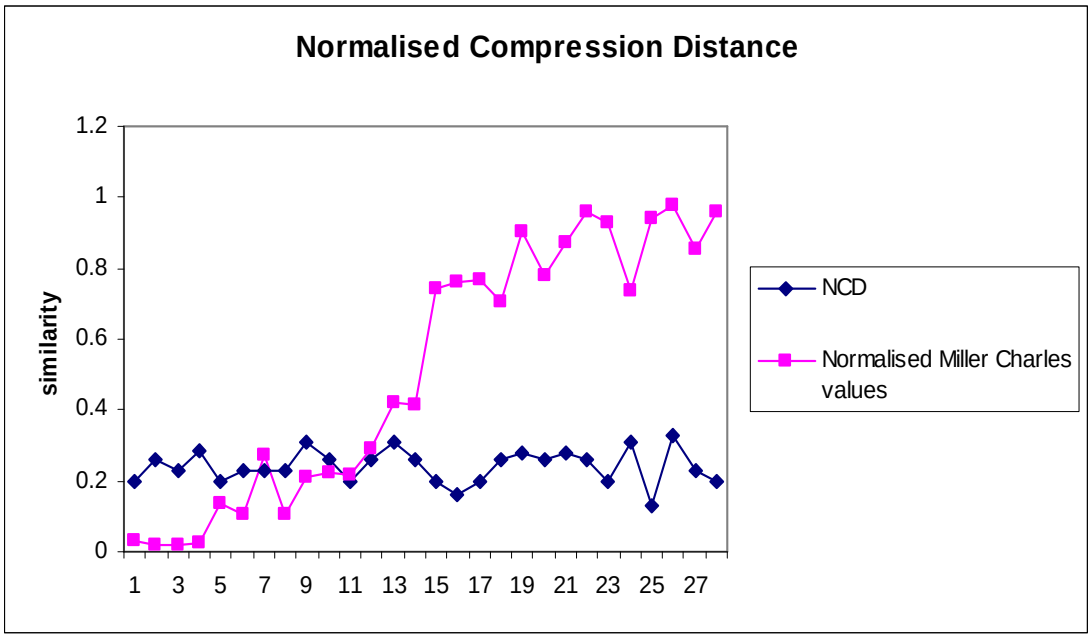

Figure 5.8: Plot of similarity values using NCD as a similarity measure

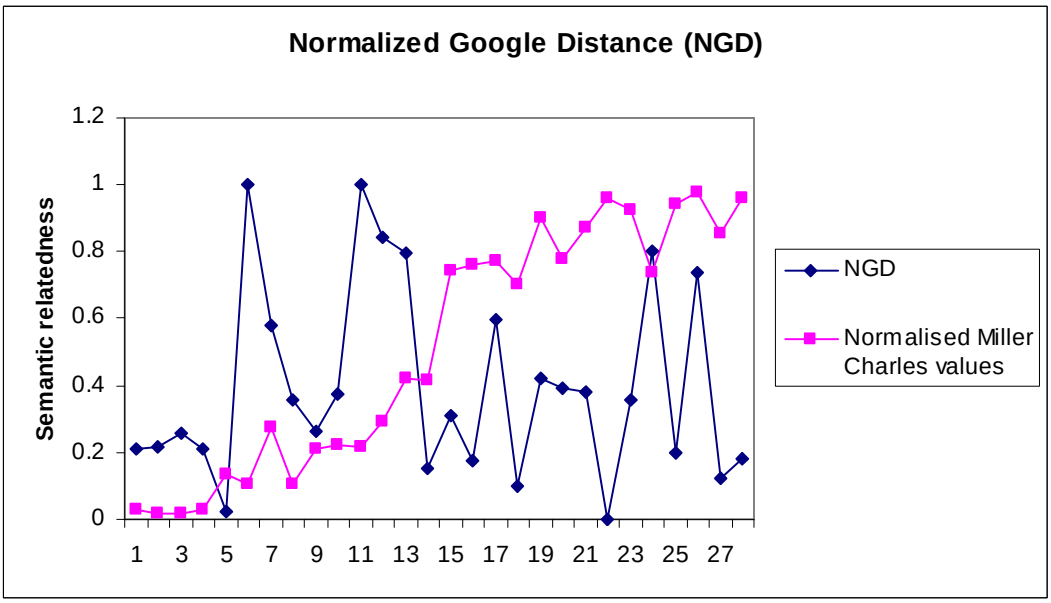

Figure 5.9: Plot of semantic relatedness values using NGD as a similarity measure

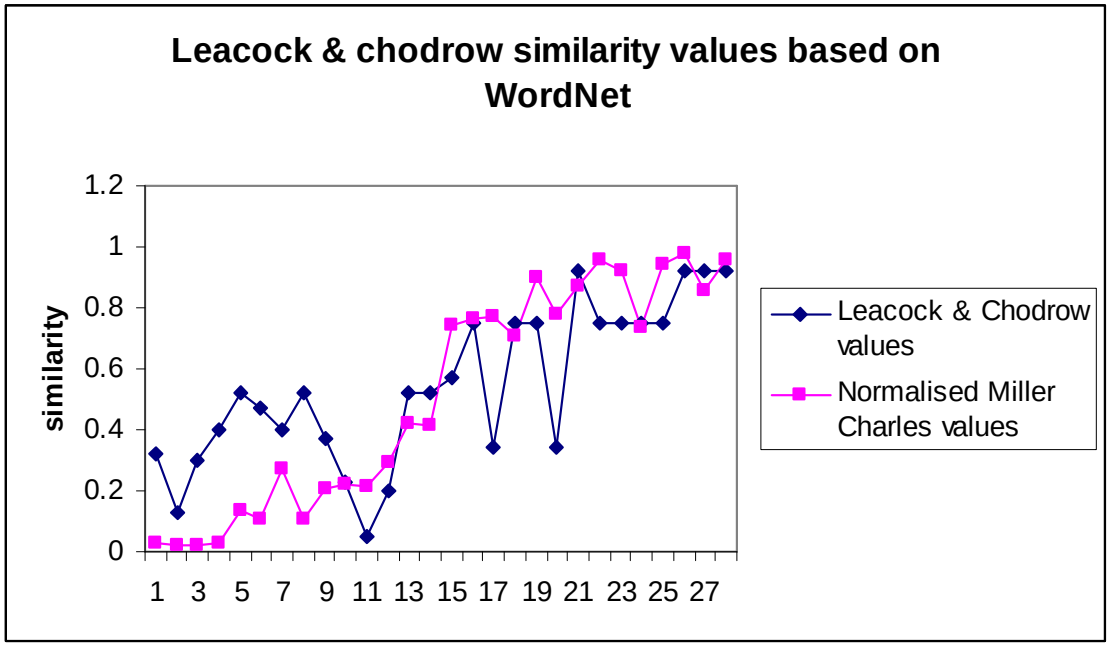

Figure 5.10: Plot of similarity values using Leacock & Chodrow similarity measure based on WordNet

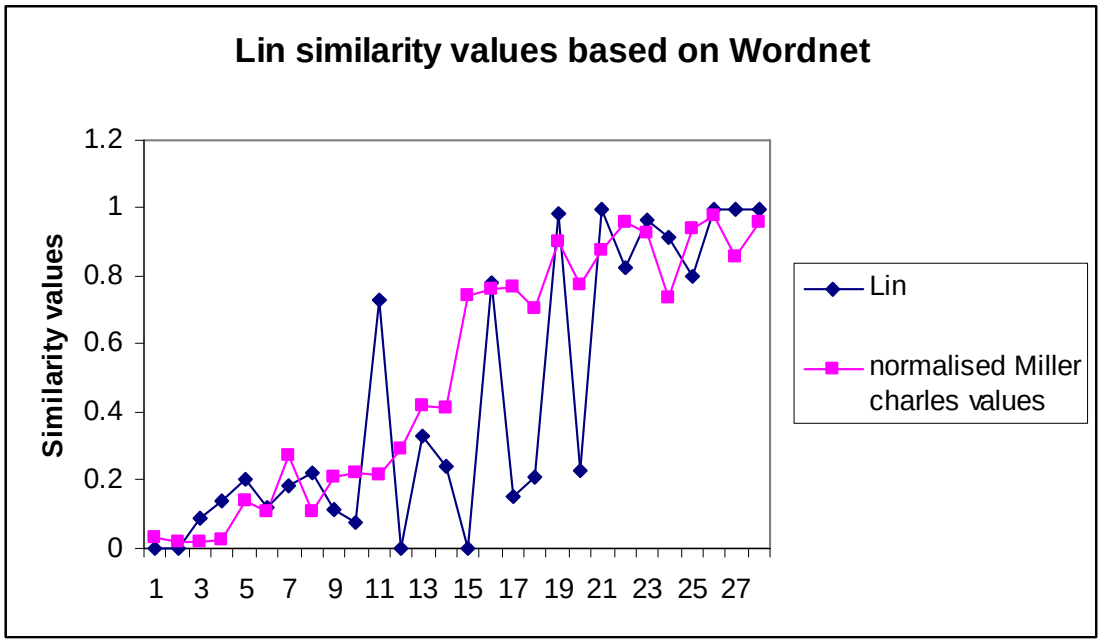

Figure 5.11: Plot of similarity values using Lin similarity measure based on WordNet

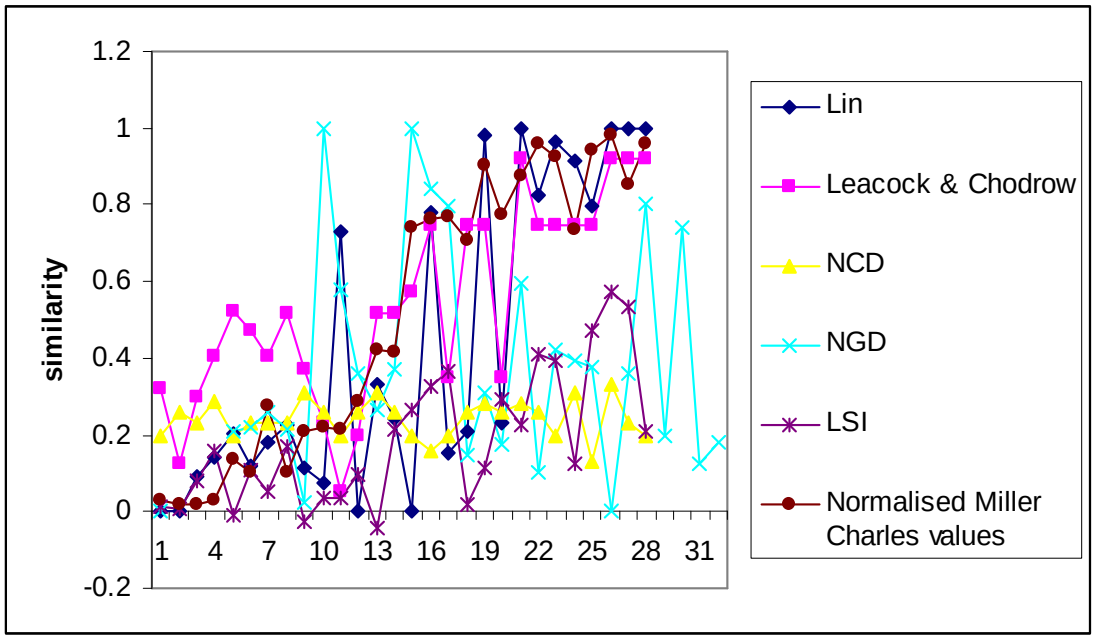

Figure 5.12: Plot of similarity values using all the measures against Normalized Miller Charles values

CHAPTER 6
CONCLUSIONS & FUTURE SCOPE

6.1 Conclusions

Due to large amount of information available today, Text Classification or categorization of text into fixed pre-defined categories has become one of the most important needs of retrieval of information that is relevant to a user's query.

In this thesis, we have studied various methods that can be used for text classification. These include Latent Semantic Indexing (LSI) and Measures of Similarity and Semantic Relatedness.

LSI is a statistical technique which is based on the principle that words used in the same context tend to have same meanings. It performs categorization of text based on the similarity between either documents or terms.

In this thesis work, the LSI algorithm has been implemented to find the nearest neighbor, matrix comparison, one to many comparison and sentence to coherence values, based on a fixed corpora. Though the results are satisfactory, but it can be seen that LSI has its limitations in that it makes no use of word order, *i.e.* there is no consideration for syntactic relations or logic, or morphology. Moreover, if the size of the corpora grows, then results are affected.

Therefore, the use of various measures of semantic relatedness is explored for text classification. It has been observed that the lexical database WordNet can be used to compute the similarity and semantic relatedness measures. The semantic relatedness measure takes into account all of the semantic links offered by WordNet. It considers WordNet as a graph, weighs edges depending on their type and depth and computes the maximum relatedness between any two nodes, connected via one or more paths. But due

to the very limited size of WordNet as a knowledge base, researchers have started to look for more comprehensive knowledge bases.

The advent of Wikipedia has fulfilled the need for a more comprehensive knowledge base. The results obtained from Wikipedia outperform those obtained using Word Net, hence showing the effectiveness of Wikipedia in determining the similarity between two words.

Although Wikipedia has proven to be better than WordNet, many terms are still unavailable on Wikipedia. This has motivated the use of the whole web as the knowledge base for calculating semantic relatedness. This has given rise to various other measures like Normalized Google Distance (NGD), Normalized Compression Distance(NCD) etc.

The values of WordNet measures like Leacock & Chodrow, Lin etc. and Web based measures like NGD, NCD, LSI etc. have been calculated and compared with the standard values of the Miller Charles benchmark dataset. It can be shown from the graphs that the measure which is closest to the standard Miller Charles values is NGD, which is a Web based method.

Thus, it can be concluded that the Miller Charles dataset should not only focus on WordNet because of its limited dictionary, but take into consideration the entire Web as a source of knowledge.

6.2 Future Scope

The major achievements of our work have been compiled and listed in Chapter 5, but there is still a lot of scope for improvement. The given work can be extended further:

- Large datasets can be used for our experiments on LSI.
- Knowledge of text classification can be used for designing a Semantic Web.

REFERENCES

[1] D.A. Grossman, O.Frieder, “Information Retrieval: Algorithms and Heuristics”, Second Edition, Vol.15, 2004.

[2] E.M. Voorhees, D.K. Harman, “TREC: Experiment and Evaluation in Information Retrieval”, MIT Press, 2005.

[3] F. Sebiastiani, “Machine Learning in Automated Text Categorization”, ACM Computing Surveys, 1999.

[4] Naive Bayes Classifier Learning Algorithm, Wikipedia Encyclopedia, from http://en.wikipedia.org/wiki/Naive_Bayes

[5] Rish, Irina, "An empirical study of the Naive Bayes Classifier", IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 2001.

[6] Support Vector Machine Classifier, Wikipedia Encyclopedia, from http://en.wikipedia.org/wiki/Support_vector_machine.

[7] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, R. A. Harshman, “Indexing by latent semantic analysis”, Journal of the American Society of Information Science, pp. 391–407, 1990.

[8] T. Landauer and D. Lanham,“Learning Human-like Knowledge by Singular Value Decomposition: a Progress Report”, Advances in Neural Information Processing Systems 10, Cambridge: MIT Press, pp. 45-51, 1998.

[9] S. Dumais, “Latent semantic indexing (LSI)”, TREC-3 report , The Third Text Retrieval Conference, NIST special publication , pp. 219–230, 1995.

[10] Zongli Jiang and Changdong Lu,"A Latent Semantic Analysis Based Method of Getting the Category Attribute of Words", International Conference on Electronic Computer Technology, 2009.

[11] Gracia and Eduardo Mena, “Web-Based Measure of Semantic Relatedness “, J. Bailey et al. (Eds.): WISE 2008, LNCS 5175, pp. 136–150, 2008. Springer-Verlag Berlin Heidelberg 2008.

[12] Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka, "Measuring the Similarity between Implicit Semantic Relations using Web Search Engines", WSDM’09,Barcelona, Spain, 2009.

[13] Saif Mohammad, Graeme Hirst, "Distributional Measures as Proxies for Semantic Relatedness", Kluwer Academic Publishers, 2005.

[14] Budanitsky, A., Hirst, G., “Evaluating WordNet-based measures of semantic distance”, Computational Linguistics 32(1), pp.13–47, 2006.

[15] Joachims, T., "Text categorization with Support Vector Machines: learning with many relevant features", In Proceedings of ECML-98, 10th European Conference on Machine Learning (Germany), pp.137–142, 1998.

[16] Joachims, T., "Transductive inference for Text Classification Using Support Vector machines", In Proceedings of ICML-99, 16th International Conference on Machine Learning (Bled, Slovenia), pp.200–209, 1999.

[17] J. I. Maletic, A. Marcus, “Using latent semantic analysis to identify similarities in source code to support program understanding”, In Proceedings of the 12th International Conference on Tools with Artificial Intelligences (ICTAI 2000), pp. 46–53, 2000.

[18] Salton, G. and McGill, M., “Introduction to Modern Information Retrieval”, McGraw-Hill, 1983.

[19] George Tsatsaroni,Vicky Panagiotopoulou, "A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness”

[20] Y. Li, Z. A. Bandar, and D. McLean, "An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources", IEEE Trans. On Knowledge and Data Engineering, pp:871–882, 2003.

[21] Graeme Hirst and David St-Onge, “Lexical chains as representations of context for the detection and correction of malapropisms”, In: Fellbaum, pp. 305–332, 1998

[22] Claudia Leacock and Martin Chodorow, ”Combining local context and WordNet similarity for word sense identification”, In: Fellbaum 1998, pp. 265–283.

[23] Jay J. Jiang and David W. Conrath, “Semantic similarity based on corpus statistics and lexical taxonomy”, In: Proceedings of International Conference on Research in Computational Linguistics, Taiwan, 1997.

[24] Dekang Lin, “An information-theoretic definition of similarity”, In: Proceedings of the 15th International Conference on Machine Learning, Madison, WI, 1998.

[25] Resnik, P., “Using information content to evaluate semantic similarity in a taxonomy”, In: 14th International Joint Conference on AI, Montreal (Canada), 1995.

[26] Strube, M., Ponzetto, S.P., “Wikirelate! computing semantic relatedness using Wikipedia", In: AAAI. AAAI Press, Menlo Park , 2006

[27] Bollegala, D., Matsuo, Y., Ishizuka, M., “Measuring semantic similarity between words using web search engines”, In: Proc. of WWW, Banff, Canada 2007.

[28] Miller, G.A., Charles, W.G., “Contextual Correlates of Semantic Similarity”, In: Language and Cognitive processes , 1991.

[29] Cilibrasi, R.L., Vit´anyi, P.M., “The Google similarity distance”, IEEE Transactions on Knowledge and Data Engineering , pp.370–383, 2007.

[30] Yang, Y. and Liu, X. , "A re-examination of text categorization methods”, In Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval (Berkeley, CA), pp.42–49, 1999.

[31] C.J.C. Burges, “A tutorial on support vector machines for pattern recognition”, Data Mining and Knowledge Discovery, pp.121–167, 1998

[32] Landauer, T. K., Foltz, P. W., and Laham, D., "An Introduction to Latent Semantic Analysis", Discourse Processes, vol. 25, no. 2&3, pp. 259-284, 1998.

[33] S. T. Dumais, J. Nielsen, “Automating the assignment of submitted manuscripts to reviewers”, In: Research and Development in Information Retrieval”, pp. 233–244, 1992.

[34] J. I. Maletic, A. Marcus, “Using latent semantic analysis to identify similarities in source code to support program understanding”, In: Proceedings of the 12th International Conference on Tools with Artificial Intelligences (ICTAI 2000), pp. 46–53, 2000.

[35] S. Kawaguchi, P. K. Garg, M. Matsushita, K. Inoue, Mudablue, “An automatic categorization system for open source repositories”, In: Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC 2004), pp. 184–193, 2004.

[36] A. Marcus, D. Poshyvanyk, “The conceptual cohesion of classes”, In: Proceedings Internationl Conference on Software Maintenance (ICSM 2005), IEEE Computer Society Press, Los Alamitos CA, pp. 133–142, 2005.

[37] Adrian Kuhn, Stephane Ducasse, Tudor Girba, “Semantic Clustering: Identifying Topics in Source Code”, Language and Software Evolution Group, LISTIC, Universite de Savoie, France, 2006.

[38] Chris Buckley, “Implementation of the smart information retrieval system”, Technical Report TR85-686, Cornell University, Ithaca, NY, USA, 1985.

[39] Susan T. Dumais, “Improving the retrieval of information from external sources, Behaviour Research Methods, Instruments and Computers”, pp. 23:229-236, 1991.

[40] Pedersen, Ted, Siddhartha Patwardhan, and Jason Michelizzi,. “Wordnet::similarity—measuring the relatedness of concepts”, In: Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04). AAAI Press, Cambridge, MA, pp. 1024–1025, 2004

[41] Wu, Zhibiao and Martha Palmer, Semantics and Lexical selection, In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp.133–138, 1994

LIST OF PUBLICATIONS

- Shirin Chandna and Shalini Batra, *"Comparitive Analysis of various Semantic Relatedness Measures"* communicated in CiiT International Journal of Artificial Intelligent Systems and Machine Learning, June, 2010.
- Shirin Chandna and Shalini Batra, *"Using Latent Semantic Indexing(LSI) and Measures of Semantic Relatedness for Text Classification"* communicated in Journal of Information Technology.