Concept Mining: A Conceptual Understanding based Approach

by

Shady Shehata

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2009

Abstract

Due to the daily rapid growth of the information, there are considerable needs to extract and discover valuable knowledge from data sources such as the World Wide Web. Most of the common techniques in text mining are based on the statistical analysis of a term either word or phrase. These techniques consider documents as bags of words and pay no attention to the meanings of the document content. In addition, statistical analysis of a term frequency captures the importance of the term within a document only. However, two terms can have the same frequency in their documents, but one term contributes more to the meaning of its sentences than the other term. Therefore, there is an intensive need for a model that captures the meaning of linguistic utterances in a formal structure. The underlying model should indicate terms that capture the semantics of text. In this case, the model can capture terms that present the concepts of the sentence, which leads to discover the topic of the document.

A new concept-based model that analyzes terms on the sentence, document and corpus levels rather than the traditional analysis of document only is introduced. The concept-based model can effectively discriminate between non-important terms with respect to sentence semantics and terms which hold the concepts that represent the sentence meaning.

The proposed model consists of concept-based statistical analyzer, conceptual ontological graph representation, concept extractor and concept-based similarity measure. The term which contributes to the sentence semantics is assigned two different weights by the concept-based statistical analyzer and the conceptual ontological graph representation. These two weights are combined into a new weight. The concepts that have maximum combined weights are selected by the concept extractor. The similarity between documents is calculated based on a new concept-based similarity measure. The proposed similarity measure takes full advantage of using the concept analysis measures on the sentence, document, and corpus levels in calculating the similarity between documents.

Large sets of experiments using the proposed concept-based model on different datasets in text clustering, categorization and retrieval are conducted. The experiments demonstrate extensive comparison between traditional weighting and the concept-based weighting obtained by the concept-based model. Experimental results in text clustering, categorization and retrieval demonstrate the substantial enhancement of the quality using: (1) concept-based term frequency (tf), (2) conceptual term frequency (ctf), (3) concept-based statistical analyzer, (4) conceptual ontological graph, (5) concept-based combined model.

In text clustering, the evaluation of results is relied on two quality measures, the F-Measure and the Entropy. In text categorization, the evaluation of results is relied on three quality measures, the Micro-averaged F1, the Macro-averaged F1 and the Error rate. In text retrieval, the evaluation of results relies on three quality measures, the precision at 10 documents retrieved P(10), the preference measure (bpref), and the mean uninterpolated average precision (MAP). All of these quality measures are improved when the newly developed concept-based model is used to enhance the quality of the text clustering, categorization and retrieval.

Acknowledgements

This thesis would not be possible without the support of many individuals, to whom I would like to express my gratitude. I will always be indebted to my supervisors, Prof. Fakhri Karray and Prof. Mohamed Kamel, for their support, encouragement, guidance, and most importantly trust. Prof. Karray's trust and support was instrumental in giving me confidence to achieve many accomplishments. I would like to thank Prof. Karray for his encouragement and guidance throughout my research. Prof. Kamel's input and guidance was invaluable to the quality and contribution of the work presented in this thesis, as well as in other publications. I would like to thank Prof. Kamel for his advice, valuable insights and feedback throughout my research. Without them this research would not have been possible.

I would like also to thank many faculty members of the University of Waterloo, most notably my committee members, Prof. Chrysanne DiMarco, Prof. Krzysztof Czarnecki and Prof. Kostas Kontogiannis for their valuable input and suggestions.

I wish to thank many of my colleagues at the Pattern Analysis and Machine Intelligence (PAMI) Lab, especially Khaled Hammouda, Mohamed El-Abd, Masoud Makrehchi, Moataz El Ayadi, Hanan Ayad and Rasha Kashef for valuable discussions and insights.

I would like to thank the PAMI administrative secretaries, Margaret Ulbrick, Heidi Campbell, and Rosalind Klein for their great help during my program. I am also thankful to the Electrical and Computer Engineering Department graduate studies coordinator Wendy Boles and the Electrical and Computer Engineering administrative staff Annette Dietrich, Lisa Hendel for their help and support during my graduate studies.

I would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Ontario Graduate Scholarship (OGS) program, the Faculty of Engineering and the Graduate Studies Office at the University of Waterloo, and the Learning Objects Repository Network (LORNET) for their financial support.

I also would like to thank Hazem Shehata for being a great friend and for his constant support and help during my studies. I would like to thank Muhammad Nummer, Hatem El-Behairy, Ayman Ismail, Hassan Hassan, Ahmed Youssef, Ismael El- Samahy, Mohamed El-Dery, Mohamed Hassan for being such great friends.

I would like to thank my wife, Reem, for her unconditional support and love, without which many things would not be possible. I would like also to thank my mother Nabila and my father Hassan for their support and encouragement throughout my life.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Preface

We are living in the age of Information Technology. The amazing growth of information that is being generated and stored today makes information an important issue to organize and analyze. In most organizations employee time and effort is wasted in ineffective searches through multiple information sources including the World Wide Web and other conventional data sources. This problem of information overload is further making worse due to the unstructured format of the majority of the data. Most of the textual information is in the natural language form. The vast amount of data found in an organization, some estimates run as high as 80%, are textual such as website, books, articles, reports, emails, etc. [1]. On the other hand most of the content of the Word Wide Web, the largest document collection, is designed for humans to read and not for computer programs to manipulate. Up till now, the Web has developed a medium of documents for people rather than for computers which can process data and information automatically [2]. While the amount of textual data in natural language form is increasing, a human ability to understand and process this information remains constant. A human editor can only recognize that a new event has occurred by carefully following all the web pages or other textual data sources. This is clearly inadequate for the volume and complexity of the information involved. The need for automated extraction of useful knowledge from huge amounts of textual data in order to assist human analysis is apparent. The rapid adoption of the business models is driving the demand for software that will help organizations to effectively analyze, manage, and manipulate data to leverage information to competitive advantage [1]. Text Mining is an automated technique that aims to discover high level information in huge amount of textual data and present it to the potential user. However, users always ask for further intelligent methods of text analysis such as an automated machine which can understand textual information and extracts the meaning out of text. A user wants to get more than just a few words of text. He/She may want to query an idea. This procedure would provide a new efficient mechanism for navigation through texts. By this mechanism, summaries of documents can be created automatically. Moreover, documents can be clustered, classified, or retrieved based on their meaning. Achieving this functionality could have great practical implications for our daily text processing activities.

1.2 Motivations

Text mining is statistical analysis of word frequencies within a document. Text mining generally ignores word order. For instance, ”The cat sat on the mat” is the same as ”the mat sat on the cat” to most text mining systems. This is called the ”bag of words” approach. One of the common drawbacks of Bag of words learning algorithms is that the models are very big. There is a record in the model for every word that appears in the source documents.

It is important to note that understanding the meaning of words couldn't be deduced from statistical analysis of word frequencies. Natural language was developed for humans to communicate with one another and to record information, and computers are a long way from understanding natural language. Humans have the ability to understand the meaning of text and humans can easily overcome obstacles that computers cannot easily handle such as spelling variations and contextual meaning. However, although human mind can understand the meaning of unstructured data, human lacks the computer's ability to process text in large volumes or at high speeds.

Herein lays the key to concept mining: creating technology that combines the human way of understanding with the speed and accuracy of a computer. Concept mining is related to understand the meaning of text. Inferring what a piece of text is about is a central step for machine understanding. Any word we use might have multiple meanings, and we use the context to disambiguate what is meant. We can formalize the idea of meaning by linking meaning to concepts and multiple words might be used to represent a particular concept. Therefore, there is a need for a representation that captures the semantics in text in a formal structure. This need arises from the necessity to perform a variety of tasks that involves the meaning of the linguistic input.

1.3 Contributions

Natural Language is ambiguous. Each word can have different meanings. Understanding the meaning of text relies on understanding concepts that mentioned in this text. The main objective of this research work is to introduce a concept mining method that aims to understand the meaning of text based on conceptual understanding. The proposed approach extracts concepts which capture semantics in text. This work aims to extract and use concept-based information, which represents semantics in text, in several applications to demonstrate the improvement of application output after using the concept mining method. This major objective comprises the following distinct contributions:

- Proposed and developed a new concept-based model that analyzes terms on the sentence, document and corpus levels rather than the traditional analysis of document only.
- Proposed and developed a new concept-based statistical analyzer that weights terms on the sentence and document levels.
- Designed and introduced a new concept-based similarity measure that measures the similarity between documents based on their meanings.
- Proposed a new conceptual ontological graph representation that captures the important concepts with respect to the text meaning based their positions in the proposed representation.
- Applied the concepts extracted by the conceptual ontological graph representation to enhance the quality of text categorization, retrieval and search engines.
- Introduced a concept-based combined model that combines between concept-based statistical analyzer and conceptual ontological graph representation.
- Showed the effectiveness of using the concepts extracted by the concept-based model as an accurate measure for enhancing the quality of the text clustering, categorization and retrieval.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 presents a review of the literature in the areas of text mining, text clustering, text categorization, text retrieval, natural language understanding focussing on thematic roles field and knowledge representation. Chapter 3 outlines the proposed concept-based model and its main components, in particular the concept-based statistical analyzer, conceptual ontological graph (COG), concept extractor, concept-based similarity measure. Chapter four presents experimental results to demonstrate the effect of the concept-based model in enhancing the quality of the text clustering, categorization, and retrieval. Finally, chapter five summarizes and suggests future work.

Chapter 2

Background and Literature Review

In this chapter a review is given on various topics that are deemed relevant to the proposed research work. Moreover, this chapter reviews text mining techniques including text clustering, categorization, retrieval, representation models and similarity measurements found in the literature. This chapter focuses on the natural language understanding field especially in the area of thematic roles and role labeling task. In addition, this chapter reviews the fundamentals of the knowledge representation area including conceptual graphs and ontology.

2.1 Text Mining

Text mining is the process of extracting important information and knowledge from unstructured text. The term text mining was first proposed by Feldman and Dagan in [3]. Text Mining is a field that is at the intersection of many other research fields, including, but not limited to:

- Data Mining (DM),
- Knowledge Discovery from Databases (KDD),
- Information Extraction (IE),
- Information Retrieval (IR), and
- Databases (DB)

The main difference between the text mining and the data mining is that data mining tools are designed to deal with structured data from databases or XML-based files. However, text mining

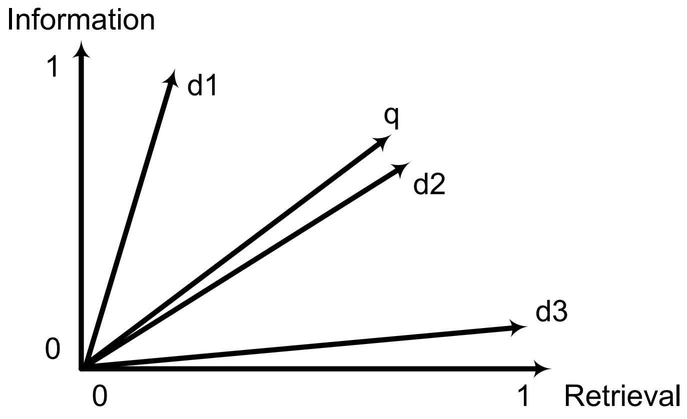

Figure 2.1: A Vector Space with Two Dimensions

deals with unstructured or semi-structured data such as text documents, HTML files, and emails. Thus, text mining is a much generalized solution for text, where large volumes of different types of information should be managed and merged.

2.1.1 Text Representation Model

In data mining, usually there is a fixed model for data that is used by most mining algorithms. This data model varies depending on the nature of the data. One of the most broadly used models in text mining systems is the Vector Space Model (VSM) [4]. Documents and terms are represented in a high dimensional space. Each dimension corresponds to a word in the documents collection. The documents vectors that are closed to the term vector represent the most relevant documents to the term which means that these documents contain words similar to the words in the term. Figure 2.1 represents a vector space with two dimensions [5]. The two dimensions correspond to the terms information and retrieval.

In Figure 2.1, the entities represented in the space are term q represented by the vector (0.72, 0.72), and three documents d1, d2, and d3 with the following coordinates respectively: (0.14, 0.99), (0.9, 0.7), and (0.99, 0.14). The coordinates of the term weights are derived from occurrence counts. There are many different terms that are used in the term weighting such as term frequency and document frequency. It can be combined as follows:

$$weight(i,j) = \begin{cases} (1 + log(tf_{i,j}))log\frac{N}{df_i} & tf_{i,j} \geq 1 \\ 0 & tf_{i,j} = 0 \end{cases} \tag{2.1}$$

where N is the total number of documents. The first clause applies for words occurring in the document, whereas for words that do not appear $tf_{i,j} = 0$, the $weight(i, j) = 0$. Another model for document representation is called N-gram [6]. The n-gram model considers the document as a sequence of characters. Using a sliding window of size n, the original character sequence is scanned to produce all n-character sub-sequences. There are two basic operations that have to be done on text before applying the feature selection. The first operation removes stop-words from text. Second operation applies word-stemming algorithm. Stop-words are common words such as "the", "and", "a", ... etc. that are believed to have no significance for capturing meaningful information about a document. Stemming algorithms converts any word form into its original base form. One of the popular stemming algorithms is the Porter stemmer [7].

2.1.2 Text Similarity Measure

When using VSM, the feature space constitutes a metric space where documents represented as points in a multidimensional space. There are two measures that are usually used which are the cosine measure, and the Jaccard measure [8, 9]. The cosine measure is expressed by

$$\cos(x, y) = \frac{x \cdot y}{\parallel x \parallel \parallel y \parallel} \tag{2.2}$$

where (.) denotes the vector dot product, and $||x||$ is the length of vector x. The cosine measure assigns high similarity values to documents that share the same set of words with high term frequencies, and lower values to those that do not. The Jaccard measure is expressed by

$$sim(x, y) = \frac{\mid x \cap y \mid}{\mid x \cup y \mid} \tag{2.3}$$

It locates the overlap between two documents by computing the number of terms that are shared between them. The Jaccard measure can work for either continuous or binary feature vectors. Minkowski distances [10], another class of distance functions, is defined by

$$\parallel x - y \parallel_p = \sqrt[p]{\sum_{i=1}^{n} \mid x_i - y_i \mid^p} \tag{2.4}$$

where $x, y \in R^2$. This distance function describes an infinite number of the distances indexed by p, which assumes values greater than or equal to 1. Some of the common values of p and their respective distance functions are:
Manhattan Distance

$$\parallel x - y \parallel_1 = \sum_{i=1}^{n} \mid x_i - y_i \mid \qquad p = 1 \tag{2.5}$$

Euclidean Distance

$$\| x - y \|_2 = \sqrt{\sum_{i=1}^{n} | x_i - y_i |^2} \qquad p = 2 \tag{2.6}$$

Tschebyshev Distance

$$\| x - y \|_\infty = \max_{i=1,2,\ldots,n} | x_i - y_i | \qquad p = \infty \tag{2.7}$$

In fact, there is no ideal association between comparing words and comparing meanings. Hence, it requires semantic representation of text and similarity measure to determine whether one part of a representation model is similar to another part or not.

2.2 Text Clustering

Document clustering aims to automatically divide documents into groups based on similarities of their contents. Each group (or cluster) consists of documents that are similar between themselves (have high intra-cluster similarity) and dissimilar to documents of other groups (have low inter-cluster similarity). Clustering documents can be considered as an unsupervised task that attempts to classify documents by discovering underlying patterns, i.e., the learning process is unsupervised, which means that no need to define the correct output (i.e., the actual cluster into which the input should be mapped to) for an input.

Document clustering has been investigated for use in a number of different areas of text mining and information retrieval. Initially, document clustering was investigated for improving the precision or recall in information retrieval systems [11,12] and as an efficient way of finding the nearest neighbors of a document [13]. More recently, clustering has been proposed for use in browsing a collection of documents [14] or in organizing the results returned by a search engine in response to a users query [15]. Document clustering has also been used to automatically generate hierarchical clusters of documents [16]. A somewhat different approach [17] finds the natural clusters in an already existing document taxonomy (Yahoo!), and then uses these clusters to produce an effective document classifier for new documents. Agglomerative hierarchical clustering and K-means are two clustering techniques that are commonly used for document clustering. Agglomerative hierarchical clustering is often portrayed as better than K-means, although slower. A widely known study, discussed in [18], indicated that agglomerative hierarchical clustering is superior to K-means. Scatter/Gather [14], a document browsing system based on clustering, uses a hybrid approach involving both K-means and agglomerative hierarchical clustering. K-means is used because of its efficiency and agglomerative hierarchical clustering is used because of its quality.

2.2.1 Clustering Techniques

In this section, we will give a brief discussion about the common clustering algorithms that are used in the experimental study of this thesis work.

Hierarchical techniques [18, 19] produce a nested sequence of partitions, with a single, all inclusive cluster at the top and singleton clusters of individual points at the bottom. Each intermediate level can be viewed as combining two clusters from the next lower level (or splitting a cluster from the next higher level). The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendogram. This tree graphically displays the merging process and the intermediate clusters. For document clustering, this dendogram provides a taxonomy, or hierarchical index.

There are two basic approaches to generating a hierarchical clustering:

a) Agglomerative: Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance.

b) Divisive: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split. We summarize the traditional agglomerative hierarchical clustering procedure as follows:

Simple Agglomerative Clustering Algorithm

1. Compute the similarity between all pairs of clusters, i.e., calculate a similarity matrix whose ij^{th} entry gives the similarity between the i^{th} and j^{th} clusters.

2. Merge the most similar (closest) two clusters.

3. Update the similarity matrix to reflect the pairwise similarity between the new cluster and the original clusters. 4. Repeat steps 2 and 3 until only a single cluster remains.

In contrast to hierarchical techniques, partitional clustering techniques create a one-level (unnested) partitioning of the data points. If K is the desired number of clusters, then partitional approaches typically find all K clusters at once. Contrast this with traditional hierarchical schemes, which bisect a cluster to get two clusters or merge two clusters to get one. Of course, a hierarchical approach can be used to generate a flat partition of K clusters, and likewise, the repeated application of a partitional scheme can provide a hierarchical clustering. There are a number of partitional techniques, K-means algorithm is widely used in document clustering. K-means is based on the idea that a center point can represent a cluster. In particular, for K-means we use the notion of a centroid, which is the mean or median point of a group of points. Note that a centroid almost never corresponds to an actual data point. The basic K-means clustering technique is presented below.

Basic K-means Algorithm for finding K clusters.

1. Select K points as the initial centroids.
2. Assign all points to the closest centroid.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until the centroids do not change.

k-Nearest Neighbour Clustering (k-NN) [20, 21]. This algorithm is used in classification and in clustering. It utilizes the property of nearest neighbours k, i.e., an object should be put in the same cluster as its nearest k neighbours. The algorithm accepts a user specified threshold, e, on the nearest-neighbour distance. For each new document, the similarity is compared to every other document, and the top k documents are chosen. Accordingly, the new document is grouped with the cluster where the majority of the top k documents are assigned.

Single Pass Clustering [20, 22]. Single pass clustering method also expects a similarity matrix as its input and outputs clusters. The clustering method takes each object sequentially and assigns it to the closest previously created cluster, or creates a new cluster with that object as its first member. A new cluster is created when the similarity to the closest cluster is less than a specified threshold. This threshold is the only externally imposed parameter. Commonly, the similarity between an object and a cluster is determined by computing the average similarity of the object to all objects in that cluster.

2.2.2 The Vector Space Model and Document Clustering

Many issues specific to documents are discussed more fully in information retrieval texts [11, 12]. We briefly review a few essential topics to provide a sufficient background for understanding document clustering. In the vector-space model each document, d, is considered to be a vector, in the term-space (set of document words). In its simplest form, each document is represented by the (TF) vector, $d_{tf} = (tf_1, tf_2, , tf_n)$, where tf_i is the frequency of the i_{th} term in the document. (Normally very common words are stripped out completely and different forms of a word are reduced to one canonical form.) In addition, this model weights each term based on its inverse document frequency (IDF) in the document collection. (This discounts frequent words with little discriminating power.) Finally, in order to account for documents of different lengths, each document vector is normalized so that it is of unit length. The similarity between two documents must be measured in some way if a clustering algorithm is to be used. There are a number of possible measures for computing the similarity between documents, but the most common one is the cosine measure, as discussed in section(2.1.2).

Given a set, S, of documents and their corresponding vector representations, the centroid vector is the averaging the weights of the various terms present in the documents of S. Analogously to documents, the similarity between two centroid vectors and between a document and a centroid vector are computed using the cosine measure. For K-means clustering, the cosine measure is used to compute which document centroid is closest to a given document.

2.2.3 Evaluation of Cluster Quality For Clustering

Two measures of cluster goodness or quality are used. One type of measure allows us to compare different sets of clusters without reference to external knowledge and is called an internal quality measure. The other type of measures evaluate how well the clustering is working by comparing the groups produced by the clustering techniques to known classes. This type of measure is called an external quality measure. One external measure is entropy [23], which provides a measure of goodness for un-nested clusters or for the clusters at one level of a hierarchical clustering. Another external measure is the F-measure which is more oriented toward measuring the effectiveness of the clustering technique. There are many different quality measures and the performance and relative ranking of different clustering algorithms can vary substantially depending on which measure is used. However, if one clustering algorithm performs better than other clustering algorithms on many of these measures, then we can have some confidence that it is truly the best clustering algorithm for the situation being evaluated.

2.2.3.1 Entropy

Entropy is a measure of quality of the clusters. Let C be a clustering solution. For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij} , the probability that a member of cluster j belongs to class i. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $E_j = -\sum_i p_{ij} log(p_{ij})$, where the sum is taken over all classes. The total entropy for a set of clusters is calculated as the sum of entropies of each cluster weighted by the size of that cluster:

$$E_C = \sum_{j=1}^{n} (\frac{M_j}{M} \times E_j), \tag{2.8}$$

where M_j is the size of cluster j, and M is the total number of data objects.

2.2.3.2 F measure

The second external quality measure is the F measure [24], a measure that combines the precision and recall ideas from information retrieval [11, 12]. The precision P and recall R of a cluster j with respect to a class i are defined as:

$$P = Precision(i, j) = \frac{M_{ij}}{M_j}, \tag{2.9}$$

$$R = Recall(i, j) = \frac{M_{ij}}{M_i}, \tag{2.10}$$

M_{ij}: is the number of members of class i in cluster j,
M_j: is the number of members of cluster j, and
M_i: is the number of members of class i.

The F-measure of a class i is defined as:

$$F(i) = \frac{2PR}{P + R}. \tag{2.11}$$

With respect to class i, the cluster with the highest F-measure is considered to be the cluster that maps to class i, and that F-measure becomes the score for class i. The overall F-measure for the clustering result C is the weighted average of the F-measure for each class i:

$$F_C = \frac{\sum_i (|i| \times F(i))}{\sum_i |i|}, \tag{2.12}$$

where $|i|$ is the number of objects in class i. The higher the overall F-measure, the better the clustering, due to the higher accuracy of the clusters mapping to the original classes.

2.3 Text Categorization

Text mining attempts to discover new, previously unknown information by applying techniques from natural language processing and data mining. Categorization, one of the traditional text mining techniques, is supervised learning paradigm where categorization methods try to assign a document to one or more categories, based on the document content. Classifiers are trained from examples to conduct the category assignment automatically. To facilitate effective and efficient learning, each category is treated as a binary classification problem. The issue here is whether or not a document should be assigned to a particular category or not.

The goal of text categorization is the classification of 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. This is a supervised learning problem. Since categories may overlap, each category is treated as a separate binary classification problem. The first step in text categorization is to transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task. Information Retrieval research suggests that word stems work well as representation units and that their ordering in a document is of minor importance for many tasks. This leads to an attribute-value representation of text. Each distinct word1 w_i corresponds to a feature, with the number of times word w_i occurs 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 at least 3 times and if they are not stop-words (like and, or, etc.).

This representation scheme leads to very high-dimensional feature spaces containing 10000 dimensions and more. Many have noted the need for feature selection to make the use of conventional learning methods possible, to improve generalization accuracy, and to avoid over-fitting.

2.3.1 Text Categorization Techniques

In this section, we will give a brief discussion about the common categorization algorithms that are used in the experimental study of this thesis work.

The Rocchio's Algorithm [25] is based on the Relevancy Feedback Algorithms for Document Relevancy. Relevancy Feedback Models are an effective way of modifying and expanding user queries (such as search engines).

Rocchio's Algorithm is one of the earliest methods used for queries. This algorithm is based on the idea that if the relevance for a query is known, an optimal query vector will maximize the average query-document similarity for relevant documents, and will simultaneously minimize query-document similarity for non relevant documents.

The intuitive idea of Rocchio's Algorithm is to iteratively increase the weights of those terms contained in labeled relevant documents while penalizing the terms in the irrelevant documents. Recent studies have shown that Rocchio's Algorithm has a poor performance when the proportion of relevant documents in the whole corpus is low.

Naive Byes [26] algorithms are among the most successful known algorithms for learning to classify text documents. It predicts by reading a set of examples in attribute value-representation and than by using the Bayes Theorem to estimate the posterior probabilities of all qualifications. For each instance of the example language a classification with the highest posterier probability is chosen as the prediction.

K-Nearest Neighbor [27], the goal of this categorization method is to simply separate the data based on the assumed similarities between various classes. Thus, the classes can be differentiated from one another by searching for similarities between the data provided. The K-Nearest Neighbor is suitable for data streams. KNN does not build a classifier in advance. When a new sample arrives, KNN finds the K neighbors nearest to the new samples from the training space based on some suitable similarity or distance metric.

KNN is a good choice when simplicity and accuracy are the predominant issues. KNN can be superior when a resident, trained and tested classifiers has a short useful lifespan, such as in the case with the data streams where new data is added rapidly and the training set is ever changing. KNN does not rely on prior probabilities, and it is computationally efficient. The main computation is the sorting of the training documents in order to find out the K nearest neighbors for the test document.

Support Vector Machine (SVM) is a supervised learning algorithm developed over the past decade by Vapnik and others [28]. The algorithm addresses the general problem of learning to discriminate between positive and negative members of a given class of n-dimensional vectors.

The SVM algorithm operates by mapping the given training set into a possibly high-dimensional feature space and attempting to locate in that space a plane that separates the positive from the negative examples. Having found such a plane, the SVM can then predict the classification of an unlabeled example by mapping it into the feature space and asking on which side of the separating plane the example lies. Much of the SVM's power comes from its criterion for selecting a separating plane when many candidates planes exist: the SVM chooses the plane that maintains a maximum margin from any point in the training set. Statistical learning theory suggests that, for some classes of well-behaved data, the choice of the maximum margin hyperplane will lead to maximal generalization when predicting the classification of previously unseen examples [28]. The SVM algorithm can also be extended to cope with noise in the training set and with multiple classes [29].

For example, if a training data set containing n examples, each of which is a vector of m numbers. These vectors may be thought of as points in an m-dimensional space. In theory, a simple way to build a binary classifier is to construct a hyperplane (i.e., a plane in a space with more than three dimensions) separating class members (positive examples) from non-members (negative examples) in this space. Unfortunately, most real-world problems involve non-separable data for which there does not exist a hyperplane that successfully separates the positive from the negative examples. One solution to the inseparability problem is to map the data into a higher-dimensional space and define a separating hyperplane there. This higher-dimensional space is called the feature space, as opposed to the input space occupied by the training examples. With an appropriately chosen feature space of sufficient dimensionality, any consistent training set can be made separable. However, translating the training set into a higher-dimensional space incurs both computational and learning-theoretic costs. Furthermore, artificially separating the data in this way exposes the learning system to the risk of finding trivial solutions that overfit the data.

SVMs elegantly sidestep both difficulties. They avoid overfitting by choosing the maximum margin separating hyperplane from among the many that can separate the positive from negative examples in the feature space. Also, the decision function for classifying points with respect to the hyperplane only involves dot products between points in the feature space. Because the algorithm that finds a separating hyperplane in the feature space can be stated entirely in terms of vectors in the input space and dot products in the feature space, a support vector machine can locate the hyperplane without ever representing the space explicitly, simply by defining a function, called a kernel function, that plays the role of the dot product in the feature space. This technique avoids the computational burden of explicitly representing the feature vectors.

2.3.2 Evaluation of Text Categorization

In order to evaluate the quality of the text categorization, three widely evaluation measures in document categorization and retrieval literatures are used. These measures are the Macro-averaged performance F1 measure (the harmonic mean of precision and recall), the Micro-averaged performance F1 measure, and the error rate.

Recall that in binary classification (relevant/not relevant), the following quantities are considered:

- p^+ = the number of relevant documents, classified as relevant.
- p^- = the number of relevant documents, classified as not relevant.
- n^- = the number of not relevant documents, classified as not relevant.
- n^+ = the number of not relevant documents, classified as relevant.

Obviously, the total number of documents N is equal to:

$$N = p^+ + n^+ + p^- + n^- \tag{2.13}$$

For the class of relevant documents:

$$Precision(P) = \frac{p^+}{p^+ + n^+} \tag{2.14}$$

$$Recall(R) = \frac{p^+}{p^+ + p^-} \tag{2.15}$$

The $F - measure_\alpha$ is defined as:

$$F_\alpha = \frac{(1+\alpha) * P * R}{(\alpha * P) + R} \tag{2.16}$$

The error rate is expressed by:

$$Error = \frac{n^+ + p^-}{N} \tag{2.17}$$

Generally, the Macro-averaged measure is determined by first computing the performance measures per category and then averaging these to compute the global mean. The Micro-averaged measure which is determined by first computing the totals p^+, p^-, n^+, and n^- for all categories and then use these totals to compute the performance measures. Micro-averaged gives equal weight to every document, while Macro-averaged gives equal weight to each category.

2.4 Information Retrieval

Although the term information retrieval seems to be very wide, information retrieval generally focuses on narrative information. The items typically processed by information retrieval systems include letters, newspaper and magazine articles, books, medical summaries, research papers, Web pages, and so on. These items are generally referred to as documents.

Sometimes information retrieval (IR) is used as a more general term, covering all kinds of retrieval tasks, and document retrieval or text retrieval is used to refer to the task outlined above. Very often, however, information retrieval and document retrieval are used synonymously as document retrieval is the prevalent area of research. One important property of documents is that the information is encoded in natural language, which exhibits the following problematic properties:

- Ambiguity: e.g., shot has many different meanings, including the act of firing a gun, a photograph, an attempt, or an injection.
- Imprecision: there are many different ways to express a concept, e.g., sonographic detection of fetal ureteral obstruction, obstetric ultrasound, and prenatal ultrasonic diagnosis all refer to the same concept of using ultrasound to diagnose pregnancy [30].
- Implicitness: much of the information in a text is not expressed explicitly. Although start, begin, and initiate could be considered equivalent, the preference of one term over an other by an author might convey important information which is difficult to formalize.
- Vagueness: Natural language is often very vague, e.g., rather large or relatively small are not easily quantifiable in numbers.

As user requests for information, called queries, are also formulated in natural language or use natural-language terms, they pose the same problems, making it even harder to find the relevant information.

In a document retrieval system it is thus not certain that all relevant documents are found, or that all retrieved documents are actually relevant to the query. The ratio of documents retrieved versus the number of available documents relevant to the query, i.e., the fraction returned out of all desirable documents is called recall. The ratio of the number of relevant documents retrieved versus the total number of documents retrieved, or the useful fraction of what was actually retrieved is called precision. These two numbers are the most common measures for the performance of IR systems. Database management systems, on the other hand, can give very precise answers to detailed queries but cannot provide information on the basis of queries that are only vaguely worded.

2.4.1 Document Index

An index corresponds to a library catalog: instead of having to go through the shelves one can look up the topic of interest in the catalog and finds the positions of the relevant documents. Similarly, an index of an information retrieval system allows to find the documents matching a particular query without having to look at the documents themselves. This speeds up the search considerably (by several orders of magnitude). However, an index has to be built before it can be used. One common index type are inverted files. Inverted files work as follows. Each document in the collection is a assigned a list of attributes which are supposed to represent the document. The most common type of attributes are keywords. The inverted file is then the sorted list of keywords of all documents, where each keyword has links to the documents that contain that keyword.

Controlled vocabulary approaches used to be quite popular for the selection of index terms from documents. Under this approach only words which are in a list of allowed index terms will be used for indexing. This method is closely related to traditional manual indexing methods using a controlled indexing language. While this method theoretically allows to build a very good index, it has several drawbacks. First, the controlled vocabulary has to be specified. This is only possible for a relatively small domain. Second, words that are not included in the vocabulary are not searchable, and third, searching cannot be done by end users but only by expert intermediaries who are acquainted to the indexing language and know how to specify a query. The modern approach, however, is to use the full text of the documents as indexing terms (hence the term full-text retrieval). After the removal of stop words (words which are too frequent to be of any use, e.g., determiners and other function words), the remaining word forms are normally conflated, i.e. supposedly semantically related word forms are mapped to a common form (the actual index term). Finally, the index terms are weighted according to their frequency and stored in the inverted file. Term conflation is usually done for two reasons: first, as stated above, different morphological forms of a word should be mapped to a common form. The assumption is that the word forms are semantically related, and that in a search for, stemming, occurrences of stemmed and stem are also relevant. The second, though less important reason is that the conflation of terms reduces the size of the index.

2.4.2 Search Engines

Search engines play an important role in the World Wide Web. One of the most important features in a search engine is the ranking algorithm for relevant documents. Recently, most of the search engines rank relevant document based on the hyper links that refer to documents. However, ranking for document relevance should be relied on document content. Some basic operations are required for information retrieval including stemming and eliminating stop words as discussed earlier. The following sections explore two common search platforms named Lucene Search Engine and Terrier Platform that are used in the experimental study of the thesis.

2.4.3 Lucene Search Engine

For Lucene search engine, the baseline retrieval method is the popular TF-IDF [3] (Term Frequency/Inverse Document Frequency) term weighting. The Lucene score of query q for document d correlates to the cosine-distance or dot-product between document and query vectors in a Vector Space Model (VSM) of Information Retrieval. A document whose vector is closer to the query vector in that model is scored higher. The score[1] is computed as follows:

$$score(q,d) = coord(q,d) * queryNorm(q) * \sum_{tinq} (tf(tind) * idf(t)2 * t.getBoost() * norm(t,d)) \tag{2.18}$$

where $tf(t \text{ in } d)$ correlates to the term's frequency, defined as the number of times term t appears in the currently scored document d. Documents that have more occurrences of a given term receive a higher score, $idf(t)$ stands for Inverse Document Frequency which correlates to the inverse of docFreq (the number of documents in which the term t appears). This means rarer terms give higher contribution to the total score. $coord(q,d)$ is a score factor based on how many of the query terms are found in the specified document. Typically, a document that contains more of the query's terms will receive a higher score than another document with fewer query terms. $queryNorm(q)$ is a normalizing factor used to make scores between queries comparable. This factor does not affect document ranking (since all ranked documents are multiplied by the same factor), but rather just attempts to make scores from different queries (or even different indexes) comparable. This is a search time factor computed by the Similarity in effect at search time. $t.getBoost()$ is a search time boost of term t in the query q as specified in the query text. Finally, $norm(t,d)$ encapsulates a few (indexing time) boost and length factors such as document boost.

[1]Further details about the Lucene score can be found at http://lucene.apache.org

2.4.4 Terrier Information Retrieval Platform

For Terrier platform, the baseline retrieval method is the Okapi BM25 formula [31] [32]. Given a query $Q = T1, ..., Tn$, BM25 assigns to a document D the relevance score

$$S^{(D)}_{BM25} = \sum_{i=1}^{n} wT_i . \frac{(k_1 + 1).f_{D,T_i}}{f_{D,T_i} + k_1.((1 - b) + b.\frac{|D|}{avgdl})} \tag{2.19}$$

where f_{D,T_i} is the number of occurrences of T_i within D, $|D|$ is the length of the document D (number of tokens), and $avgdl$ is the average document length in the text collection. wT_i is Ti's inverse document frequency:

$$wT_i = log(\frac{\#documents}{\#documentscontainingT_i}) \tag{2.20}$$

2.4.5 Evaluation of Text Retrieval

Most of the retrieval evaluation measures are derived in some way from recall and precision. Precision is the proportion of retrieved documents that are relevant, and recall is the proportion of relevant documents that are retrieved [33].

$$P = Precision(i, j) = \frac{N_r}{T} \tag{2.21}$$

and

$$R = Recall(i, j) = \frac{N_r}{T_r} \tag{2.22}$$

where:

- N_r is the number of relevant records retrieved,
- T is the total number of irrelevant and relevant records retrieved, and
- T_r is the total number of relevant records in the database.

In order to evaluate the quality of the text retrieval, three widely evaluation measures in information retrieval literature are used. These measures are the precision at 10 documents retrieved P(10) [34], the preference measure, bpref, which is a function of the number of times judged non-relevant documents are retrieved before relevant documents [34], and the mean uninterpolated average precision (MAP) [34].

2.4.6 Information Retrieval and Natural Language

One of the first problems related to the use of natural language in information retrieval is that of morphologic variation. This refers to the fact that words may occur in inflected forms, or that derivation is used to produce new but related words, or that words are combined into compounds. Morphologic variations can very often be regarded as semantically related and thus equivalent for retrieval purposes. In English, the number of possible inflected and derived forms is relatively small, and variation is mostly restricted to the attachment of suffixes. Compounds which are not yet lexicalized are in most cases written as separate words. Stemming, i.e., the removal of suffixes using a list of possible suffixes, is therefore considered a practical way to map related word forms to a common stem.

2.5 Natural Language Processing and Understanding

Natural Language Processing (NLP) is both a modern computational technology and a method of investigating and evaluating claims about human language itself. NLP is a term that links back into the history of Artificial Intelligence (AI), the general study of cognitive function by computational processes, with an emphasis on the role of knowledge representations. The need for representations of human knowledge of the world is required in order to understand human language with computers.

Natural language understanding (NLU) is a subfield of natural language processing (NLP). NLU is concerned with the work of mapping from some surface representation of linguistic material expressed as speech or text to an underlying representation of the meaning carried by that surface representation.

2.5.1 The Study of Language

Language is an essential aspect of human behavior and it represents a vital part of our lives. In written form, it is considered as a large record of knowledge that is transferred from one generation to next generation. In spoken form it serves as the primary means for people communication. Language is studied in several different academic fields. Each field defines its own set of problems and has its own methods for addressing them. The linguist, for instance, studies the structure of language itself, considering questions such as why specific combinations of words create sentences but other words do not, and why a sentence can have some meanings but other sentences do not [35].

The psycholinguist, on the other hand, studies the processes of human language production and comprehension, considering questions such as how people determine the suitable structure of a sentence and when they decide on the appropriate meaning for words. Philosophers consider what it means to have beliefs, goals, and intentions, and how these cognitive capabilities relate to language [35]. It is important to note that taking advantage of what is known from all the other disciplines is required to build a computational model.

2.5.2 Different Levels of Language Analysis

Knowledge about the structure of the language itself should be used in a natural language-system. This considerable knowledge refers to many aspects such as what the words are, what the words mean, how words join to construct a sentence, and how word meanings contribute to sentence meanings. Linguistic behavior is not the only aspect that makes humans intelligent. World knowledge and human abilities for reasoning are important aspects for human intelligence. A person, who knows about the world knowledge in general, can answer a question without knowing a lot about the language structure. The following are some different forms of knowledge that are relevant for natural language understanding [35]:

- *Phonetic and Phonological Knowledge*: concerns how words are linked to the sounds that recognize them. This knowledge is central for speech-based systems.
- *Morphological Knowledge*: concerns how words are created from more basic meaning units called morphemes. A morpheme is the primitive unit of meaning in a language (for example, the meaning of the word ”friendly” is derivable from the meaning of the noun ”friend” and the suffix ”-ly”, which transforms a noun into an adjective).
- *Syntactic Knowledge*: concerns how words can be composed to construct correct sentence. It also concerns how to determine the structural role of each word in the sentence and what phrases are subparts of what other phrases.
- *Semantic Knowledge*: concerns what words mean and how these meanings can be combined in sentences to form the meanings of a sentence.
- *Pragmatic Knowledge*: concerns how sentences are used in different contexts and how context can affect the interpretation of the sentence.

- *Discourse Knowledge*: concerns the effect of the right away preceding sentences on the interpretation of the next sentence. This information is especially important for interpreting pronouns.
- *World Knowledge*: concerns the general knowledge about world structure that users of language should obtain. It also concerns what each language user knows about the beliefs and goals of other users.

2.5.3 Thematic Roles

All human languages at their core semantic structure seem to have a form of predicate argument layout. This underlying structure allows the creation of a composite meaning representation from the meaning of the individual parts of a linguistic input. Grammar has an important role for supporting the predicate-argument structure. Consider the following examples:

- *My daughter wants a doll.*
- *My son wants to play outside.*

These two examples can be classified as having one of the following syntactic argument frames: (Noun Phrase(NP) wants NP) or (NP wants Infinitive Phrase (Inf-NP)). In this case, some facts could be driven for the particular predicate "wants" (1) There are two arguments to this predicate (2) Both arguments must be NPs (3) The first argument is pre-verbal and plays the role of the subject (4) the second argument is a post-verbal and plays the role of the direct object. By observing and analyzing semantic information associated with these frames, extensions of these frames into the semantic realm which are semantic roles and semantic restriction on these roles are considered in the following sections. In each of these cases, the pre-verbal argument always acts as the entity doing the wanting, while the post-verbal plays the role of the entity that is wanted. More generally, the predicate argument structure permits the link between the arguments in surface structures of the input text and their associated semantic roles. The study of roles associated with verbs is usually referred to thematic role or case role analysis [36]. Thematic roles, first proposed by Gruber and Fillmore [37], are set of categories that provide a shallow semantic language to characterize the verb arguments. For example: the agent of an action is the person or thing that is doing the action, the patient is the person or thing that is having done to it, and there are other roles like instrument and goal which describe other classes of semantic relationships. In terms of grammatical relations, all English verbs take a subject, which is the noun phrase that appears before the verb. Many verbs take an object noun phrase, which normally appear after the verb. Many languages make a distinction between active voice and passive voice. Active corresponds to the ordinary way of expressing the argument of a verb. For example: the agent is articulated as the subject and the patient as the object. However, in the passive voice, the patient becomes the subject and the agent is demoted to an implicit role. Over the last thirty years, a wide variety of work is done on mapping of conceptual structure to grammatical function, in the area referred to as linking theory [38]. There are different types of mapping of the conceptual (deep) roles to grammatical function which is called alternations. Many scholars show that semantic properties of verbs help predict which surface alternations they can take [39]. There are several possible patterns of arguments for each verb. Sub-Categorization Frame is the specific set of arguments that a verb can appear with. The following examples present some common sub-categorization frames in English language:

- Intransitive Verb: NP [subject]. *"The man walked"*
- Transitive Verb: NP [subject], NP [object]. *"Andrew hit the ball"*.
- Ditransitive Verb: NP [subject], NP [object], NP [indirect object]. *"Mike sent John an email"*.

Sub-categorization frames capture syntactic regularities about grammars. Similarly, there are semantic regularities which are called selectional restrictions or selectional preferences [5]. Selectional restriction is a semantic constraint forced by a lexeme on the concepts that can fill the various argument roles associated with it. For example: the verb bark prefers dogs as subject. The verb eat prefers edible things as objects.

One recent attempt to list these elements for a number of predicates of English language is the FrameNet project [40]. There are also other attempts such as PropBank which will be discussed later. The Berkeley FrameNet project is creating an on-line lexical resource for English, based on frame semantics and supported by corpus evidence [40]. The aim is to document the range of semantic and syntactic combinatory possibilities-valences-of each word in each of its senses, through computer-assisted annotation of example sentences and automatic tabulation and display of the annotation results. Each entry for a word in the FrameNet lists every set of argument it can take like thematic roles, syntactic phrases, and their grammatical functions. The thematic roles in the FrameNet are specific much more than the earlier examples. There are different domains and each domain comprises many different frames. Each FrameNet thematic role is defined as a part of a frame. For example: the Cognition domain has frames like static cognition (believe, think, understand), cogitation (brood, ruminate), judgment (accuse, admire, rebuke), etc. All of the cognition frames defined the thematic roles COGNIZER. In the judgment frame, the COGNIZER is referred to as the JUDGE; the frame also includes an EVALUEE, a REASON, and a ROLE. Consider the following example from the FrameNet dataset: ”She admired his courage in clouting the skinhead but he spoilt it by saying it had been instinctive and now he wished he had n't.” The frame is the judgment frame. The frame elements are: TARGET (a word that invoked the correspondent frame): admired, COGNIZER: She, and REASON: his courage in clouting the skinhead.

The PropBank project at the University of Pennsylvania is creating a corpus of text annotated with information about basic semantic propositions [41]. PropBank involves adding a layer of semantic annotation to the Penn English TreeBank without attempting to confirm or disconfirm any particular semantic theory. PropBank provides consistent argument labeling that will facilitate the automatic extraction of relational data. Here are some examples:

- *John broke the window*
- *The window broke*

In these examples, an argument ”the window” receives the same label [ARG1] as a subject in both sentences. In order to ensure reliable human annotation, there are explicit guidelines for labeling all of the syntactic and semantic frames of each particular verb [42].

Some researchers proposes an extension of a dependency-annotated TreeBank, PropBank using a class-based verb lexicon VerbNet [43] to create an intermediate level of representation which can be useful for machine translation applications [44].

WordNet is a lexical database system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets. WordNet was developed by the Cognitive Science Laboratory at Princeton University [45].

2.5.4 The Role Labeling Task

Recently, there are many attempts to label thematic roles to a sentence automatically. Gildea and Jurafsky [46] were the first to apply a statistical learning technique to the FrameNet database.

They present a discriminative model for determining the most probable role for a constituent given the frame, predicator, and other features. These probabilities, trained on the FrameNet database, are based on the verb, the head words of the constituents, the voice of the verb (active, passive), and the syntactic category (S, NP, VP, PP, etc.) and grammatical function (subject, object) of the constituent to be labeled. They evaluated their model on a pre-release version of the FrameNet I corpus, which at that time contained about 50,000 sentences and 67 frame types. Their model was trained by first using the parser of Collins [47], and deriving features from that parse, the original sentence, and the correct FrameNet annotation of that sentence. [48] propose a machine learning algorithm for shallow semantic parsing, extending the work of Gildea and Jurafsky [46], and others [49]. Their algorithm is based on Support Vector Machines which gives an improvement in performance over earlier classifiers by [46] . The shallow semantic parsing is formulated as a multi-class classification problem. Support Vector Machines (SVMs) is used to identify the arguments of a given predicate in a sentence and classify them by the semantic roles that they play (AGENT, THEME, GOAL, etc.).

There are two methods of semantic classification which are constituent-by-constituent (c-by-c) and word-by-word (w-by-w) classification. In the former, a deep syntactic parser is used to derive the constituents which are afterwards labeled with semantic roles. In the latter, the words are classified based on BIO tagging [50] which has three cases: at the beginning of semantic role, inside semantic role, or outside semantic role. There are two main corpora: PropBank (UPenn) [41] and FrameNet (UC Berkeley) [40] that contain sentences tagged with semantic arguments. Other researches at Stanford University presented a model of natural language generation from semantics using the FrameNet semantic role and frame ontology. They trained the model using the FrameNet corpus and applied it to the task of automatic semantic role and frame identification, producing results competitive with previous work [51]. Some recent research work deals with the semantic role labeling as a chunking problem along with multi-class classification problem [52]. A semantic role labeler (or chunker) that groups syntactic chunks (i.e. base phrases) into the arguments of a predicate. This is accomplished by casting the semantic labeling as the classification of syntactic chunks (e.g. NP-chunk, PP-chunk) into one of several classes such as the beginning of an argument (B-ARG), inside an argument (I-ARG) and outside an argument (O). This amounts to tagging syntactic chunks with semantic labels using the IOB representation. The chunker is realized using support vector machines as one versus all classifiers.

Fortunately, there are numerous efforts for semantic labeling. However, to the best of our knowledge, there is research work that employs the full potential of the output of the role labeling task in text mining process. It is important to note that extracting relations between verbs and between their arguments in the same sentence has a promising potential for understanding the meaning of a sentence.

2.5.5 Applications of Natural Language Understanding

One efficient way to describe natural language research is to consider the different applications that researchers work on. The ultimate goal of the researchers is to have a computer system that can understand natural language. Most of the natural language understanding applications can be categorized into two major categories: text-based applications and dialogue-based applications. Text-based applications include the processing of written text, such as books, newspapers, reports, manuals, or e-mail messages [35]. Text-based natural language research is ongoing in applications such as

- Searching for relevant documents on certain topics from a database of texts.
- Extracting information from messages or articles on a certain idea.
- Translating documents from one language to another language.
- Summarizing texts for certain purposes.

Dialogue-based applications include human to machine communication. This involves spoken language, potential applications includes:

- Question-answering systems, where natural language is used to query a database.
- Automated customer service over the telephone tutoring systems, where the machine interacts with a student.
- Spoken language control of a machine
- General cooperative problem-solving systems

Some of the problems encountered by dialogue systems are extremely different than in text-based systems. First, the language used is extremely different. The system needs to participate actively in order to maintain a natural and smoothing dialogue. Dialogue requires acknowledgments to verify that the meanings of things are actually understood. Even with these differences, the basic processing techniques are fundamentally the same [35].

2.5.6 Evaluating Language Understanding Systems

System understanding might vary from application to application. There are two main methods for evaluating the language understanding systems including black box evaluation and glass box evaluation. The Black box method evaluates system performance without considering the inside components or how the system works. We run the system and evaluate its performance based on the task it was designed for. For instance, if the task is a question answering system, then the evaluation will be measured on how good the system is at producing the correct answers. In the glass box method, system's sub components are identified to be evaluated individually with each one appropriate test. Systems evaluations are of crucial importance to the natural language understanding field [53].

2.5.7 Representing the Understanding

As we mentioned earlier, a vital component of understanding necessitates computing a representation of the meaning of sentences and texts. Sentence by itself couldn't be used as a representation of its meaning. One reason is that most words have multiple meanings, called senses. The words "cook" and "dish" for instance, have a sense as a verb and a sense as a noun. The "still" word has senses as a noun, verb, adjective, and adverb. Therefore, natural language is ambiguous and this ambiguity would prevent the system from making the appropriate inferences which are needed to represent the understanding. However, the disambiguation problem seems much easier than it actually is. Ambiguity generally is not noticed by people. A person understands a sentence without considering each of the possible senses of a word. However, a program must explicitly consider them one by one. To represent meaning, a more precise language is needed. The tools that are used to do this precision come from mathematics and logic. Precise language also involves the use of formal specified representation languages. Formal representation languages are specified from basic building blocks. Useful representation languages have the following two properties [35]:

- The representation must be precise and unambiguous. Every different analysis of a sentence should be expressed by a distinguished formula in the representation.
- The innate structure of the natural language sentences should be captured by the representation. In that case, sentences that seem to be structurally similar should have similar structural representations. The meanings of two sentences that are rephrased of each other should be similar related to each other.

2.6 Knowledge Representation

Knowledge representation research field applies theories and techniques from other disciplines including [54]:

- *Logic*: provides the formal structure and rules of inference.
- *Ontology*: defines the concepts and relations that exist in the application domain.
- *Computation*: provides the implemented applications of knowledge representation.

If the logic field is not included, then a knowledge representation is ambiguous and as a consequence there is no criterion for determining whether statements are conflicting or not. If ontology field is not involved, then terms and symbols are imprecise and confusing. Finally, if there are no computable models, there are no implemented computer applications that could combine between logic and ontology. "Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domain" [54].

2.6.1 Ontology

There are many definitions for the word "ontology". The following definition is the most common one which is "An ontology is an explicit and formal specification of a conceptualization" [55]. The computer science usage of the term ontology is derived from the term ontology in philosophy. The word "ontology" has a long history in philosophy, in which it refers to the subject of existence. In information science, an ontology is the product of an attempt to formulate an rigorous conceptual schema about a domain. The domain ontology is usually a hierarchical data structure that contains all relevant entities, their relationships, and rules within that domain [56]. In the knowledge sharing context, term ontology means a specification of a conceptualization. Ontology is a formal description, like a formal specification of a program, of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set of concept definitions. Ontology provides shared and common understanding scheme of a domain. Common scheme can be used for communication between people and application systems. Thus, ontology plays a major role in supporting information exchange processes in various areas. Ontology is also shared among agents that communicate and work together. An ontology which is not tied to a particular problem domain but describes general entities is known as a *foundation ontology* or *upper ontology* [56]. There are several large and general ontology such as

- *OpenCyc*: general knowledge base and commonsense reasoning engine [57].
- *WordNet*: a lexical database for the English language [45].
- *World Fact Book*: a substantial knowledge base covering basic geographic, economic, political, and demographic knowledge about the world's nations. [58].
- *UMLS*: National Library of Medicine's (NLM) Unified Medical Language System [59].

There are also several ontology languages such as

- Resource Description Framework (RDF) and RDF-Schema (RDFS) [60]
- DARPA Agent Mark-up Language (DAML) [61]
- Ontology Inference Layer (OIL) [62]
- DAML+OIL [63]
- Web Ontology Language (OWL) [64]

The following are some applications that are based on ontology [65]:

- Ontology embedded into an environmental decision support system for waste water treatment plant management: describes an ontology application in the domain of waste water treatment.
- Developing software agents based on product and process ontology: discusses how ontologies can be used to provide a unified view of system engineers and domain experts during the process of developing knowledge-based systems.
- Evaluation of an ontology-based information retrieval tool: discusses the use of explicit domain ontology in an IR tool using a quantitative method for evaluating the performance of the tool.
- MKBEEM is an ontology domain modeling support for multi-lingual services in e-commerce: describes the role of ontologies on that IST European project.

2.6.2 Conceptual Graphs

Conceptual graphs (CGs) are logic systems that are based on the existential graphs of Charles Sanders Peirce [66] and the semantic networks [67] of artificial intelligence. Conceptual Graphs (CGs) are basically labeled graphs where their nodes represent ”concepts” and their arcs represents ”relations” which connect between nodes. The main objective of the CGs is to express the meaning in a precise logical form. This form is humanly readable, and computationally achievable. Conceptual graphs have a simple mapping to language which leverages CGs to be considered as an intermediate language between formal computer languages and natural languages.

2.6.3 Forms of Knowledge Interchange

Conceptual graphs are formally defined by an abstract syntax. This syntax is independent of any notation. However, the CGs formalism can be represented in either graphical or character-based notations. For example, the following is the display form (DF) of a conceptual graph that represents the English sentence *John is going to Boston by bus* [54]. In DF, concepts are represented by rectangles: [Go], [Person: John], [City: Boston], and [Bus]. Conceptual relations are represented by circles or ovals: (Agnt) relates [Go] to the agent John, (Dest) relates [Go] to the destination Boston, and (Inst) relates [Go] to the instrument bus. The arcs that link the relations to the concepts are represented by arrows. For relations with more than two arguments, the arcs are numbered. The linear form (LF) is more compact notation than DF. However, it is more humanly readable than DF. It is exactly equivalent in expressive power to the abstract syntax and the display form. Following is the LF for the same sentence:

```
[Go]- (Agnt) -> [Person: John] (Dest) -> [City: Boston] (Inst) ->
[Bus]
```

In this form, the concepts are represented by square brackets rather than boxes. The conceptual relations are represented by parentheses instead of circles. The hyphen on the first line indicates that the relations attached to [Go] are continued on subsequent lines. Both DF and LF are designed for communication with humans or between humans and machines. For communication between machines, the conceptual graph interchange form (CGIF) has a simpler syntax. Following is the CGIF for the same sentence:

```
[Go *x] (Agnt ?x [Person ’John’]) (Dest ?x [City ’Boston’]) (Inst ?x
[Bus])
```

CGIF is designed for transfer between computer systems that use CGs as their internal representation. For communication with systems that use other internal representations, CGIF can be translated to another logic-based formalism called the Knowledge Interchange Format (KIF):

```
(exists ((?x Go) (?y Person) (?z City) (?w Bus)) (and (Name ?y John)
(Name ?z Boston) (Agnt ?x ?y) (Dest ?x ?z) (Inst ?x ?w)))
```

Although DF, LF, CGIF, and KIF look very different in terms of structure, their semantics is defined by the same logical foundations. Semantic information expressed in any one of them can be translated to the others without loss or distortion and they can all be translated to a statement of the following form in typed predicate calculus:

$$(\exists x : Go)(\exists y : Person)(\exists z : City)(\exists w : Bus)$$
$$(name(y,' John') \wedge name(z,' Boston') \wedge agnt(x, y) \wedge dest(x, z) \wedge inst(x, w))$$

2.7 Summary

In this chapter, we introduced the literature review of the text mining techniques especially the common algorithms in text clustering, categorization and retrieval search engines. These techniques are used in the experiments of the proposed work. In addition, we reviewed the required background on the natural language understanding field especially in the area of thematic roles and role labeling task. The next chapter introduces the proposed concept-based model that advances the areas of natural language processing and text mining.

Chapter 3

Concept-based Model

This chapter introduces the proposed concept-based model including the concept-based statistical analyzer, conceptual ontological graph (COG) representation, concept extractor, and concept-based similarity measure. Moreover, we introduce an illustrative example to show the objectives of each individual component. In addition, we present a mathematical framework of the concept-based model.

3.0.1 Overview

The proposed concept-based model consists of concept-based statistical analyzer, conceptual ontological graph (COG) representation, concept extractor, and concept-based similarity measure. The aim of the concept-based statistical analyzer is to weight each term on the sentence and the document levels rather than the traditional analysis of document only. The conceptual ontological graph representation presents the sentence structure while maintaining the sentence semantics in the original document. Each concept in the COG representation is weighted based on its position in the representation. The concept extractor combines the two different weights computed by the concept-based statistical analyzer and the COG representation as depicted in Fig.(3.1). The weights that are computed by the concept-based statistical analyzer and the COG representation are called $weight_{stat}$ and $weight_{COG}$ respectively.

A raw text document is the input to the proposed model. Each document has well defined sentence boundaries. Each sentence in the document is labeled automatically based on the PropBank notations [68]. After running the semantic role labeler [68], each sentence in the document might have one or more labeled verb-argument structures. The number of generated labeled verb-argument structures is entirely dependent on the amount of information in the sentence. The

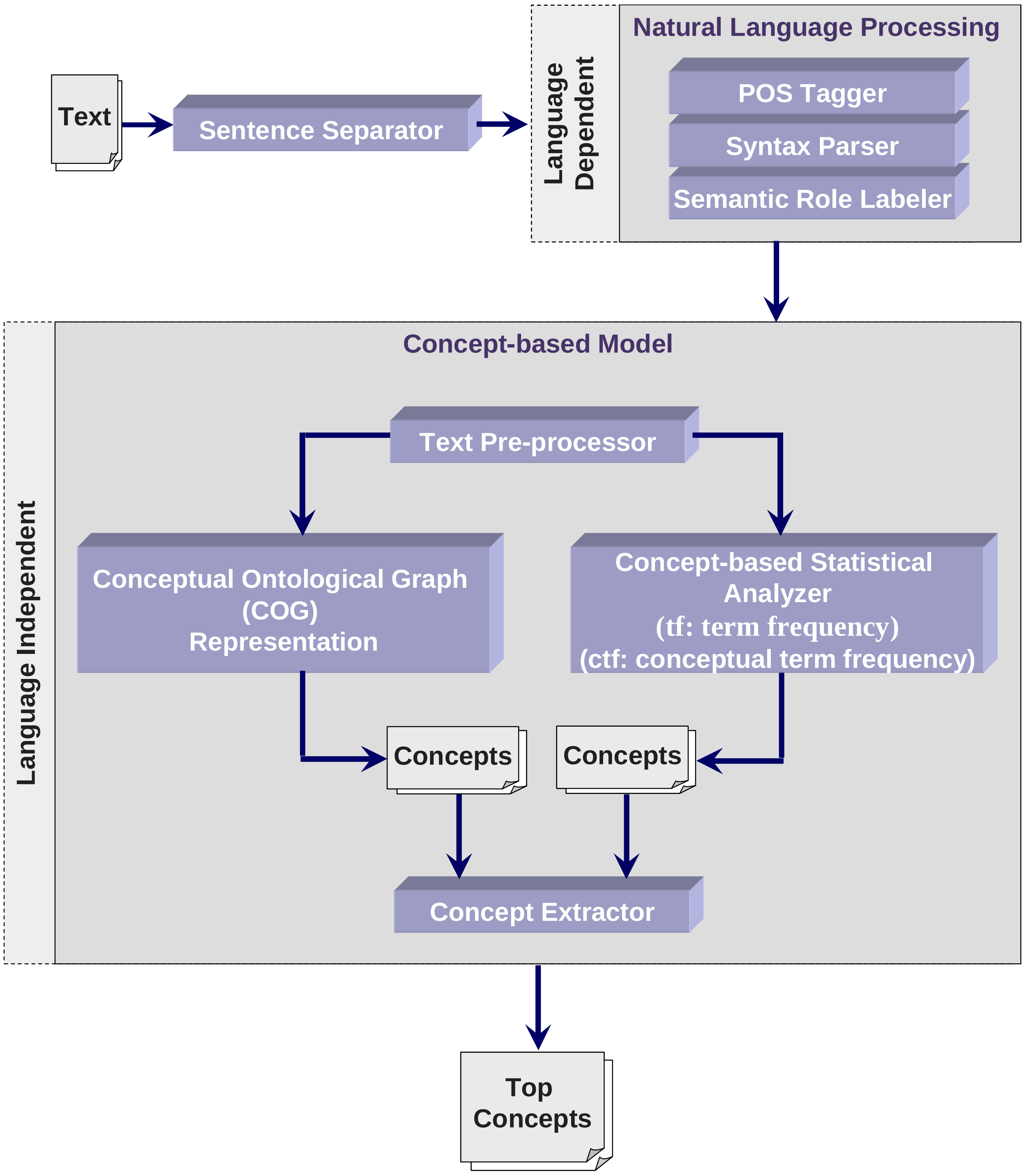

Figure 3.1: Concept-based Model

sentence that has many labeled verb-argument structures includes many verbs associated with their arguments. The labeled verb-argument structures, the output of the role labeling task, are captured and analyzed by the concept-based model on the sentence, document, and corpus levels.

In this model, both the verb and the argument are considered *terms*. One term can be an argument to more than one verb in the same sentence. This means that this term can have more than one semantic role in the same sentence. In such cases, this term plays important semantic roles that contribute to the meaning of the sentence. In the concept-based model, a labeled term either word or phrase is considered a *concept*.

While explaining each component in concept-based model, we will associate an illustrative example to show the input and the output of each individual component.

3.1 Concept-based Statistical Analyzer

The objective behind the concept-based analysis task is to achieve an accurate analysis of concepts on the sentence, document, and corpus levels rather than a single-term analysis on the document only.

3.1.1 Sentence-Based Concept Analysis

To analyze each concept at the sentence-level, a new concept-based frequency measure, called the conceptual term frequency (ctf) is proposed. The ctf calculations of concept c in sentence s and document d are as follows:

3.1.1.1 Calculating ctf of concept c in sentence s

The ctf is the number of occurrences of concept c in verb argument structures of sentence s. The concept c, which frequently appears in different verb argument structures of the same sentence s, has the principal role of contributing to the meaning of s. In this case, the ctf is a local measure on the sentence level.

3.1.1.2 Calculating ctf of concept c in document d

A concept c can have many ctf values in different sentences in the same document d. Thus, the ctf value of concept c in document d is calculated by:

$$ctf = \frac{\sum_{n=1}^{sn} ctf_n}{sn} \tag{3.1}$$

where sn is the total number of sentences that contain concept c in document d. Taking the average of the ctf values of concept c in its sentences of document d measures the overall importance of concept c to the meaning of its sentences in document d. A concept, which has ctf values in most of the sentences in a document, has a major contribution to the meaning of its sentences that leads to discover the topic of the document. Thus, calculating the average of the ctf values measures the overall importance of each concept to the semantics of a document through the sentences.

To illustrate the calculation of ctf in a document, consider a concept c which appears twice in document d in the first and the second sentences. The concept c appears five times in the verb argument structures of the first sentence s_1 and three times in the verb argument structures of second sentence s_2. In this case the ctf value of concept c is equal to $\frac{5+3}{2} = 4$

3.1.2 Document-Based Concept Analysis

To analyze each concept at the document-level, the concept-based term frequency tf, the number of occurrences of a concept (word or phrase) c in the original document, is calculated. The tf is a local measure on the document level.

3.1.3 Corpus-based Concept Analysis

To extract concepts that can discriminate between documents, the concept-based document frequency df, the number of documents containing concept c, is calculated. The df is a global measure on the corpus level. This measure is used to reward the concepts that only appear in a small number of documents as these concepts can discriminate their documents among others.

The following is the concept-based weighting $weight_{stat}$ which is used to discriminate between non-important terms with respect to sentence semantics and terms which hold the concepts that present the meaning of the sentence.

$$weight_{stat_i} = tfweight_i + ctfweight_i \tag{3.2}$$

In calculating the value of $weight_{stat_i}$ in equation (3.2), the $tfweight_i$ value presents the weight of concept i in document d at the document-level and the $ctfweight_i$ value presents the weight of the concept i in the document d at the sentence-level based on the contribution of concept i to the semantics of the sentences in d. The sum between the two values of $tfweight_i$ and $ctfweight_i$ presents an accurate measure of the contribution of each concept to the meaning of the sentences and to the topics mentioned in a document.

In equation (3.8), the tf_{ij} value is normalized by the length of the document vector of the term frequency tf_{ij} in the document d, where $j = 1, 2, ..., cn$

$$tfweight_i = \frac{tf_{ij}}{\sqrt{\sum_{j=1}^{cn} (tf_{ij})^2}}, \tag{3.3}$$

where cn is the total number of the concepts which has a term frequency value in the document d.

In equation (3.9), the ctf_{ij} value is normalized by the length of the document vector of the conceptual term frequency ctf_{ij} in the document d where $j = 1, 2, ..., cn$

$$ctfweight_i = \frac{ctf_{ij}}{\sqrt{\sum_{j=1}^{cn} (ctf_{ij})^2}}, \tag{3.4}$$

where cn is the total number of concepts which has a conceptual term frequency value in the document d.

The process of calculating ctf, tf, and df measures in a corpus is attained by the proposed algorithm which is called (Concept-based Analysis Algorithm).

The concept-based analysis algorithm describes the process of calculating the ctf, tf, and df of the matched concepts in the documents. The procedure begins with processing a new document (at line 2) which has well defined sentence boundaries. Each sentence is semantically labeled according to [68]. The lengths of the matched concepts and their verb argument structures are stored for the concept-based similarity calculations in section 3.5.

For each labeled sentence (in the for loop at line 6), concepts of the verb argument structures which represent the sentence semantics are weighted by the $weight_{tf}$, $weight_{ctf}$, $weight_{stat}$ and $weight_{combined}$ according to the values of the tf, ctf, and df (at lines 9, 10, and 11). The concepts list L_{tf}, L_{ctf}, L_{stat}, and $L_{combined}$ are sorted descendingly based on the $weight_{tf}$, $weight_{ctf}$, and $weight_{stat}$ values respectively. The maximum weighted concepts are chosen as top concepts from the concepts lists. (at lines 21,22,and 23). The concept-based statistical analysis algorithm is capable of extracting the top concepts in a document (d) in $O(m)$ time, where m is the number of concepts.

Algorithm 3.1 Concept-Based Analysis Algorithm

```
1:  $R$ is a corpus
2:  $d$ is a new Document
3:  $L_{tf}$ is an empty List ($L_{tf}$ is a top concept list)
4:  $L_{ctf}$ is an empty List ($L_{ctf}$ is a top concept list)
5:  $L_{stat}$ is an empty List ($L_{stat}$ is a top concept list)
6:  for each labeled sentence $s$ in $d$ do
7:      $c_i$ is a new concept in $s$
8:      for each concept $c_i$ in $s$ do
9:          compute $tf_i$ of $c_i$ in $d$
10:         compute $ctf_i$ of $c_i$ in $s$ in $d$
11:         compute $df_i$ of $c_i$ in $R$
12:         compute $weight_{stat_i}$ of concept $c_i$
13:         add concept $c_i$ to $L_{tf}$
14:         add concept $c_i$ to $L_{ctf}$
15:         add concept $c_i$ to $L_{stat}$
16:     end for
17: end for
18: sort $L_{tf}$ descendingly based on $max(weight_{tf})$
19: sort $L_{ctf}$ descendingly based on $max(weight_{ctf})$
20: sort $L_{stat}$ descendingly based on $max(weight_{stat})$
21: return $max(weight_{tf})$ from list $L_{tf}$
22: return $max(weight_{ctf})$ from list $L_{ctf}$
23: return $max(weight_{stat})$ from list $L_{stat}$
```

3.2 Illustrative Example

Consider the following sentence:

*"We have **noted** how some electronic techniques, **developed** for the defense effort, have eventually been **used** in commerce and industry."*

3.2.1 Example: Role Labeler

In this sentence, the semantic role labeler identifies three target words (verbs), marked by bold, which are the verbs that represent the semantic structure of the meaning of the sentence. These verbs are *noted*, *developed*, and *used*. Each one of these verbs has its own arguments as follows:

- [ARG0 We] [TARGET **noted**] [ARG1 how some electronic techniques developed for the defense effort have eventually been used in commerce and industry]
- We have noted how [ARG1 some electronic techniques] [TARGET **developed**] [ARGM-PNC for the defense effort] have eventually been used in commerce and industry
- We have noted how [ARG1 some electronic techniques developed for the defense effort] have [ARGM-TMP eventually] been [TARGET **used**] [ARGM-LOC in commerce and industry]

Arguments labels[2] are numbered Arg0, Arg1, Arg2, and so on depending on the valency of the verb in sentence. The meaning of each argument label is defined relative to each verb in a lexicon of Frames Files [68].

Despite this generality, Arg0 is very consistently assigned an Agent-type meaning, while Arg1 has a Patient or Theme meaning almost as consistently [68]. Thus, this sentence consists of the following three verb-argument structures:

1. First verb-argument structure:
   - [ARG0 We]
   - [TARGET noted]
   - [ARG1 how some electronic techniques developed for the defense effort have eventually been used in commerce and industry]
2. Second verb-argument structure:
   - [ARG1 some electronic techniques]
   - [TARGET developed]

[2]Each set of argument labels and their definitions is called a frameset and provides a unique identifier for the verb sense. Because the meaning of each argument number is defined on a per-verb basis, there is no straightforward mapping of meaning between arguments with the same number. For example, arg2 for verb *send* is the recipient, while for verb *comb* it is the thing searched for and for verb *fill* it is the substance filling some container [68].

Table 3.1: Example of Concept-based Statistical Analyzer

Number	Concepts	CTF
(1)	noted	1
(2)	electronic techniques developed defense effort eventually commerce industry	1
(3)	electronic techniques	3
(4)	developed	3
(5)	defense effort	3
(6)	electronic techniques developed defense effort	2
(7)	used	2
(8)	commerce industry	2
	Individual Concepts	CTF
(9)	electronic	3
(10)	techniques	3
(11)	defense	3
(12)	effort	3
(13)	used	2
(14)	commerce	2
(15)	industry	2

- [ARGM-PNC for the defense effort]

3. Third verb-argument structure:
    - [ARG1 some electronic techniques developed for the defense effort]
    - [ARGM-TMP eventually]
    - [TARGET used] [ARGM-LOC in commerce and industry]

3.2.2 Example: Stemmer

After each sentence is labeled by a semantic role labeler, a cleaning step is performed to remove stop-words that have no significance, and to stem the words using the popular Porter Stemmer algorithm [7]. The terms generated after this step are called *concepts*.

In this example, stop words are removed and concepts are shown without stemming for better readability as follows:

1. Concepts in the first verb-argument structure:
    - noted
    - electronic techniques developed defense effort eventually commerce industry
2. Concepts in the second verb-argument structure:
    - electronic techniques
    - developed
    - defense effort
3. Concepts in the third verb-argument structure:
    - electronic techniques developed defense effort
    - used
    - commerce industry

3.2.3 Example: Concept-based Statistical Analyzer

It is imperative to note that these concepts are extracted from the same sentence. Thus, the concepts mentioned in this example sentence are:

- noted
- electronic techniques developed defense effort used commerce industry
- electronic techniques
- developed
- defense effort
- electronic techniques developed defense effort
- used
- commerce industry

The traditional analysis methods assign same weight for the words that appear in the same sentence. However, the concept-based statistical analyzer discriminates among terms that represent the concepts of the sentence. This discrimination is entirely based on the semantic analysis of the sentence. In this example, some concepts have higher conceptual term frequency ctf than others as shown in Table 3.1. In such cases, these concepts (with high ctf) contribute to the meaning of the sentence more than concepts (with low ctf).

As shown in Table 3.1, the concept-based statistical analyzer computes the ctf measure for:
* The concepts which are extracted from the verb-argument structures of the sentence, which are in Table 3.1 from row (1) to row (8).
* The concepts which are overlapped with other concepts in the sentence. These concepts are in Table 3.1 from row (3) to row (8),
* The individual concepts in the sentence, which are in Table 3.1 from row (9) to row (15).

In this example, the topic of the sentence is about the *electronic techniques*. These concepts have the highest ctf value with 3. In addition, the concept *noted* which has the lowest ctf, has no significant effect on the topic of the sentence.

It is important to note that concepts such as *commerce* and *industry* have the ctf value with 2 which is not the highest. However, these concepts are important to the sentence semantics.

They provide significant information about what electronic techniques eventually have been used in. Thus, selecting the maximum ctf weight by the concept-based statistical analyzer only, leads to lose important concepts such as *commerce* and *industry*.

3.3 Conceptual Ontological Graph (COG)

The COG representation provides nested levels of concepts in a hierarchical manner. Each concept in the COG representation is weighted based on its position in the representation. The COG representation is used to provide a definite separation among concepts that contribute to the meaning of a sentence. This separation is needed to distinguish between important concepts and non-important concepts with respect to the meaning of a sentence.

The COG represents the sentence structure while maintaining the sentence semantics in the original documents. The output of the role labeling task, which are verbs and their arguments are represented as conceptual graphs in the COG representation. A conceptual graph is constructed for each verb-argument structure. It is one to one relationship between the verb-argument structure and the conceptual graph. These conceptual graphs are the levels of the COG representation. The number of COG levels equals to the number of the verb-argument structures in a sentence.

The COG representation is a directed hypergraph $COG = (CG, A)$ where CG is a set of nodes $\{cg_1, cg_2, ..., cg_n\}$, such that each node cg represents a conceptual graph; and A is a set of links $\{a_1, a_2, ..., a_m\}$, such that each link a is the relation between an ordered pair of nodes (or conceptual graphs) (cg_i, cg_j). These links represent the levels of the COG representation.

Each node cg in the COG representation is a conceptual graph $CG = (C, R)$ where C is a set of vertices $\{c_1, c_2, ..., c_k\}$, such that each vertex c represents a concept (either a verb or an argument); and R is a set of edges $\{r_1, r_2, ..., r_z\}$, such that each edge r is the relation between an ordered pair of vertices (or concepts). These relations, such as subjects, objects and targets, are the labels of the verbs and its arguments generated by the semantic role labeling task.

First, the COG representation captures the verbs and their arguments in a sentence based on the role labeling. Secondly, a conceptual graph is constructed for each verb-argument structure. Lastly, the entire conceptual graphs are placed and linked in one sentence-based conceptual ontological graph representation. When node x is linked to node y in the COG representation, it means that there is still detailed information about the topic of node x, and is provided through concepts and their relations in the next level by another conceptual graph represented in node y.

It is crucial to note that the contribution of the COG representation lies in its semantic-based hierarchical nature. This is due to the fact that most of the nodes in the COG representation are linked to other nodes (conceptual graphs). This means that there are more detailed information in a sentence and it is represented by these graphs. Thus, the COG representation introduces the concepts of a sentence in a descending way. The highest nodes represent the most general (or shallow) concepts of the sentence. The lowest nodes represent the least key concepts mentioned in the sentence. This hierarchical manner provides different levels of depth for concept-based analysis within a sentence.

Consequently, the COG representation shares the same ontological behavior of common ontologies regarding the semantics of the sentence. Thus, COG can be considered as a semantic-based ontology that provides a hierarchy of the levels of importance for each concept in a sentence.

3.3.1 Relations in the Conceptual Ontological Graph

As mentioned earlier in section(3.3), COG representation is a directed hypergraph of conceptual graphs (nodes). Each node in the COG representation is a conceptual graph and there are relations between nodes (conceptual graphs) in the COG representation.

There are two types of relations in the COG representation: internal relations that link between concepts in each conceptual graph and external relations that link between the entire conceptual graphs (or nodes) in the COG representation.

Constructing the internal relations is straight forward because these relations are provided by the semantic role labeler. These internal relations are the semantic labels (such as subject or object) of the verb-argument structures.

Constructing the external relations between nodes (conceptual graphs) in the COG representation is based on the following:

- As mentioned in section(3.3), a concept in the conceptual graph is either a verb or an argument. There are two types of concepts: concept (of a verb) and concept (of an argument).
- If a concept x (of an argument) in a conceptual graph cg_x consists of a concept y (of verb) in a conceptual graph cg_y with respect to words, then there is a link (or directed edge) from node cg_x pointed to node cg_y.

3.3.2 Conceptual Ontological Graph Construction

The construction of the COG representation consists of the following steps:

1. Determine the number of verb-argument structures for each sentence (e.g., a sentence that consists of one verb has only one verb-argument structure)
2. Construct a conceptual graph for each verb-argument structure.
3. Compute the amount of overlapping with respect to words between concepts (of arguments) in each conceptual graph and concepts (of verbs) in the other conceptual graphs.
4. Use the amount of words overlapping calculated from the third step to define the relations between conceptual graphs in the COG representation.
5. Combine the generated conceptual graphs from the second step and their relations from the fourth step to construct the COG representation.

Five types that represent the levels of the COG hierarchy are proposed and assigned to conceptual graphs as follows:

- ***Type One***: means that there is only one conceptual graph that corresponds to one verb-argument structure. In this case, the COG representation consists of one node only.
- ***Type Main***: the conceptual graph of main type has the maximum number of outgoing links (or directed edges) that point to other nodes in the COG representation. This means that the conceptual graph of main type has the maximum number of concepts (of arguments) that consists of concepts (of verbs) in other conceptual graphs with respect to words. The conceptual graph of main type is the highest level of the COG representation. Main type provides the shallow meaning of a sentence.
- ***Type Container***: the conceptual graph of container type has at least one outgoing link that points to another conceptual graph. Moreover, the conceptual graph of container type does not have the maximum number of outgoing links (unlike the main type). The conceptual graph of container type is neither the highest level nor the lowest level. Usually, container type is placed between the highest and the lowest levels in the COG representation. The container type provides more detailed information than the main type.

- ***Type Referenced***: the conceptual graph of referenced type has at least one incoming link from another conceptual graph. This means that the concept (of verb) in the conceptual graph of referenced type appeared at least one time in a concept (of argument) in another conceptual graph that has either main or container type. The conceptual graph of referenced type is the lowest level in the COG representation. Usually, the referenced type provides the most detailed information about the sentence.

- ***Type Unreferenced***: the conceptual graph of unreferenced type has zero directed edges neither incoming nor outgoing. This means that the conceptual graph of unreferenced type is an isolated node with degree zero. The difference between *unreferenced* and *one* types is that in the *unreferenced* type, the COG representation has other nodes. In *one* type there is only one node in the COG representation.

It is crucial to note that the incoming links measure the importance of each conceptual graph (or node) in the COG representation. The conceptual graph that has many incoming links means that its concepts are important because these concepts appear frequently in different verb-argument structures in a sentence. The process of constructing the COG is attained by the proposed algorithm which is called (COG Constructor Algorithm):

The following terminologies are used in the COG construction algorithm; Out-degree: is the number of edges going out of a vertex in a directed graph. In-degree: is the number of edges coming into a vertex in a directed graph. Degree: is the number of edges connected to a vertex in a directed graph.

After running the COG constructor algorithm, the highest level in the COG representation is the conceptual graph of *main* type. The conceptual graph of *container* type will be placed after the conceptual graph of *main* type. The conceptual graph of *referenced* type is the lowest level in the COG representation and linked with other conceptual graphs of either *main* or *container* types. An isolated node is added to the COG representation that represents the conceptual graph of *unreferenced* type.

For implementation and performance purposes, it is imperative to note that the COG representation maintains the identification number of each concept and each relation, rather than, the values of the nodes. There is a hash table that includes the unique concepts that appeared in each conceptual graph. Thus, the COG is a hierarchy of the identification numbers of the concepts that appear in each sentence. This is the source of the efficiency of the representation. It is also important to note that ontology languages are unable to capture the semantic-based levels in the

Algorithm 3.2 COG Constructor Algorithm

```
1: $s$ is a new sentence
2: declare $L_a$ as an empty List ($L_a$ is a concept list of arguments)
3: $L_v$ as an empty List ($L_v$ is a concept list of verbs)
4: create an empty directed graph $COG$
5: for each verb-argument structure in $s$ do
6:     add concept $c_a$ (of argument) to $L_a$
7:     add concept $c_v$ (of verb) to $L_v$
8:     create a conceptual graph $CG$
9:     add node $N_{CG}$ to the $COG$ graph
10: end for
11: for each concept $c_i$ of $CG_i$ in $L_a$ do
12:     for each concept $c_j$ of $CG_j$ in $L_v$ do
13:         if ($c_j$ is a substring in $c_i$) then
14:             create a directed edge from $CG_i$ to $CG_j$ in $COG$
15:         end if
16:     end for
17: end for
18: for each node $N_{CG}$ in $COG$ do
19:     if ($N_{CG}$ has the maximum out-degree) then
20:         assign $N_{CG}$ the main type
21:     end if
22:     if (out-degree of $N_{CG} \geq 1$) and (out-degree of $N_{CG}$ is not the maximum) then
23:         assign $N_{CG}$ the container type
24:     end if
25:     if (in-degree of $N_{CG} \geq 1$) then
26:         assign $N_{CG}$ the referenced type
27:     end if
28:     if (degree of $N_{CG} == 0$) then
29:         assign $N_{CG}$
30:     end if
31: end for
32: return the directed graph $COG$
```

hierarchical structure of the sentence semantics. The above algorithm is capable of constructing the COG representations for a document d in O(m) time, where m is the number of concepts in d.

3.3.3 Concept-based Weighting Scheme

The concept-based weighting is one of the main factors that captures the importance of a concept in both the sentence and the document levels. The proposed weighting scheme consists of a new measure called L_{COG} that measures the importance of each concept with respect to the sentence semantics based on its level in the COG representation. Instead of extracting concepts from only one level in the COG representation, concepts appeared in all levels of the COG representation are weighted and ranked based on their highest weight values. The top ranked concepts are extracted and used to build a concept-based document index for text retrieval.

* *Concept-based Analysis at the Document Level*

To analyze each concept at the document-level, the concept-based term frequency tf, the number of occurrences of a concept (word or phrase) i in the original document, is calculated as shown in equation (3.8).

* *Concept-based Analysis at the Sentence Level*

To analyze each concept at the sentence-level, a new measure called L_{COG} is proposed to rank concepts with respect to the sentence semantics in the COG representation. The proposed L_{COG} measure is assigned to *One*, *Unreferenced*, *Main*, *Container*, and *Referenced* levels in the COG representation with values 1,2,3,4, and 5, respectively. These values represent the importance of each level in the COG representation with respect to sentence meaning. The proposed L_{COG} measure rewards concepts in the conceptual graphs that appear in the lowest level of the COG representation. Moreover, the L_{COG} measure also rewards concepts in a conceptual graph that has more than one type (such as container and referenced type). In this case, the weight of concept i is the sum of L_{COG} values that assigned to the conceptual graph that contains concept i as shown in equation(3.5). If a concept in a conceptual graph has many incoming and outgoing links in the COG representation then this concept frequently appears in different verb-argument structures of the same sentence. This means that this concept has the principal role of contributing to the meaning of the sentence.

The proposed $weight_{COG}$ is assigned to each concept i in the COG representation and is calculated by:

$$weight_{COG_i} = tfweight_i * \frac{\sum_{j=1}^{T} L_{COG_i}}{sn_i} \tag{3.5}$$

where T is the number of types assigned to the conceptual graph that has concept i, and sn is the total number of sentences that contain concept i in document d. In equation(3.5), the $tfweight_i$ value represents the weight of concept i in document d at the document-level as shown in equation(3.8). The $\sum_{j=1}^{T} L_{COG_i}$ value represents the importance of the concept i in the document d at the sentence-level based on the contribution of concept i to the semantics of the sentences represented by the levels of the COG representation. Taking the average of the $\sum_{j=1}^{T} L_{COG_i}$ value of concept i by dividing this value on the total number of i's sentences sn of document d measures the overall importance of concept i to the meaning of its sentences in document d. A concept, which has $\sum_{j=1}^{T} L_{COG_i}$ values in most of the sentences in a document, has a major contribution to the meaning of its sentences that leads to discover the topic of the document. Thus, calculating the average of the $\sum_{j=1}^{T} L_{COG_i}$ values measures the overall importance of each concept to the semantics of a document through the sentences. The multiplication between the two values of $tfweight_i$ and $\frac{\sum_{j=1}^{T} L_{COG_i}}{sn}$ ranks the concepts in document d with respect to the contribution of each concept to the meaning of the sentences and to the topics mentioned in a document.

3.3.4 Example: Conceptual Ontological Graph Representation

Generally,

In this example for the same sentence *"We have **noted** how some electronic techniques, **developed** for the defense effort, have eventually been **used** in commerce and industry."*, three conceptual graphs are generated for each verb-argument structure as follows:

1. (noted)
   - ARG0 [We]
   - ARG1 [how some electronic techniques, developed for the defense effort, have eventually been used in commerce and industry]
2. (used)
   - ARG1 [some electronic techniques developed for the defense effort]
   - ARGM-TMP [eventually]
   - ARGM-LOC [in commerce and industry]
3. (developed)

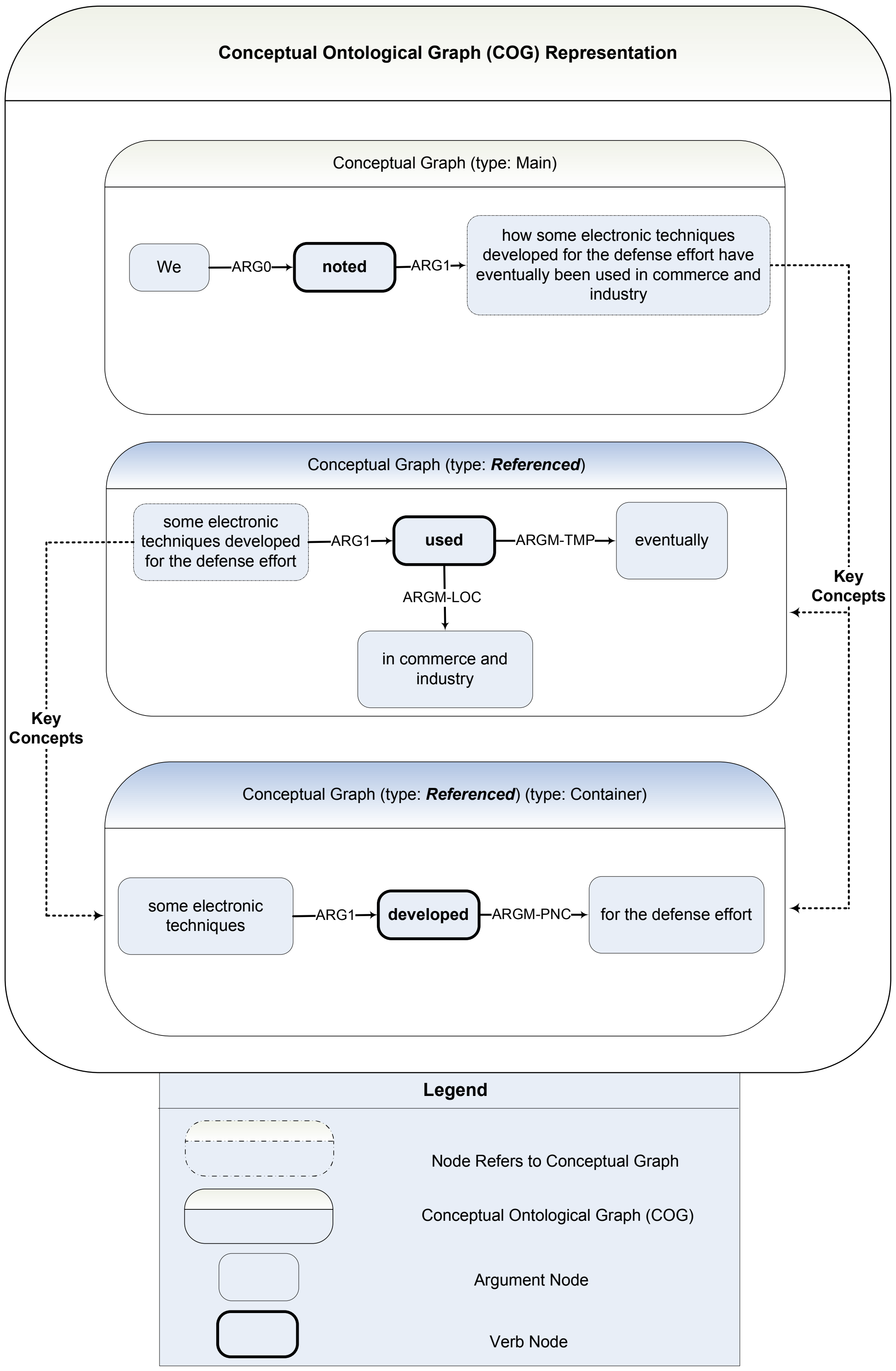
Conceptual Ontological Graph (COG) Representation
Conceptual Graph (type: Main)
We
ARG0
noted
ARG1
how some electronic techniques developed for the defense effort have eventually been used in commerce and industry
Conceptual Graph (type: *Referenced*)
some electronic techniques developed for the defense effort
ARG1
used
ARGM-TMP
eventually
ARGM-LOC
in commerce and industry
Key Concepts
Key Concepts
Conceptual Graph (type: *Referenced*) (type: Container)
some electronic techniques
ARG1
developed
ARGM-PNC
for the defense effort
Legend
Node Refers to Conceptual Graph
Conceptual Ontological Graph (COG)
Argument Node
Verb Node

Figure 3.2: Conceptual Ontological Graph (COG) Representation

- ARG1 [some electronic techniques]
- ARGM-PNC [for the defense effort]

These conceptual graphs are presented as three nodes in the COG representation as shown in (Fig. 3.2).

For better follow-up, the three conceptual graphs generated for the three verb-argument structures of the verbs *noted*, *used*, and *developed* are called by their verbs: (noted - conceptual graph), (used - conceptual graph) and (developed - conceptual graph) as shown in (Fig. 3.2).

The (noted - conceptual graph) has three concepts: (1)*[We]*, (2)*[how some electronic techniques, developed for the defense effort, have eventually been used in commerce and industry]* and (3)*[noted]*. The (noted - conceptual graph) node in the COG representation has the maximum out-degree (outgoing links) value of two because the concept *[how some electronic techniques, **developed** for the defense effort, have eventually been **used** in commerce and industry]* contains the two verbs (marked by bold and italic) of the other conceptual graphs which are *used* and *developed*. In this case, there are two outgoing links pointed from (noted - conceptual graph) to (used - conceptual graph) and (developed - conceptual graph). Thus, the (noted - conceptual graph) has the *main* type. This conceptual graph provides the general meaning of a sentence presented in *X noted Y*. More information about *Y* can be deduced from the other conceptual graphs: (used - conceptual graph) and (developed - conceptual graph).

The (used - conceptual graph) has four concepts: (1)*[some electronic techniques developed for the defense effort]*, (2)*[eventually]*, (3)*[in commerce and industry]*, and (4)*[used]*. The in-degree of the (used - conceptual graph) has the value of one presented in the incoming link pointed from the (noted - conceptual graph) as mentioned earlier. Thus, the (used - conceptual graph) has the *referenced* type. Moreover, in the (used - conceptual graph), the concept *[some electronic techniques **developed** for the defense effort]* contains the verb *developed* (marked by bold and italic) of the (developed - conceptual graph). In this case, there is an outgoing link pointed from (used - conceptual graph) to (developed - conceptual graph). The out-degree of the (used - conceptual graph) has the value of one. Thus, the (used - conceptual graph) has also the *container* type because the out-degree value is one and at the same time it is not the maximum value (not like the main type with the value of two). This conceptual graph provides more detailed information than the (noted - conceptual graph).

The (developed - conceptual graph) has three concepts: (1)*[some electronic techniques]*, (2)*[for the defense effort]* and (3)*[developed]*. The in-degree of the (developed - conceptual graph) has the value of two presented by the two incoming links from the (noted - conceptual graph) and the (used - conceptual graph) as mentioned earlier. Therefore, the (developed - conceptual graph) has the *referenced* type. This graph provides another level of details regarding the sentence semantics.

After removing the stop words which have no significance, the selected concepts, which provide important information with respect to the sentence semantics, are the concepts that appear in the conceptual graphs that have more than one type. In this example, concepts that appear in (used - conceptual graph) that has the *referenced* and *container* types are assigned L_{COG} value with 9 (5 for *referenced* type plus 4 for *container* type). These concepts, which are *electronic, techniques, developed, defense, effort, commerce* and *industry*, play important role of contributing to the sentence semantics.

At this point, concepts are assigned two different weights using two different techniques which are the concept-based statistical analyzer and the COG representation. It is important to note that both of them achieve the same functionality, in different ways. The output of the two techniques are the important weighted concepts with respect to the sentence semantics that each technique captures. However, the weighted concepts that are computed by each technique could not be exactly the same. In other words, the important concepts to the concept-based statistical analyzer could be non-important to the COG representation and vice versa. Therefore, the third component, which is the concept extractor, combines the two different weights computed by the concept-based statistical analyzer and the COG representation to denote the important concepts with respect to the two techniques.

3.4 Concept Extractor

The concept extractor algorithm describes the process of combining the $weight_{stat}$ (computed by the concept-based statistical analyzer) and the $weight_{COG}$ (computed by the COG representation) into one new combined weight called $weight_{comb}$. The concept extractor selects the top concepts that have the maximum $weight_{comb}$ values.

The proposed $weight_{comb}$ is calculated by:

$$weight_{comb_i} = weight_{stat_i} * weight_{COG_i} * \log(\frac{N}{df_i}) \tag{3.6}$$

where N is the total number of documents and df_i is the concept-based document frequency of concept c_i (the number of documents containing concept c_i). The $\log(\frac{N}{df_i})$ value analyzes concepts on the corpus level. It rewards concept c_i which appears in a small number of documents.

The process of selecting the top concepts from the concepts extracted by the concept-based statistical analyzer and the COG representation is attained by the proposed Concept Extractor Algorithm.

Algorithm 3.3 Concept-based Extractor Algorithm

```
1: $R$ is a corpus
2: $d_{doci}$ is a new Document
3: $L$ is an empty List ($L$ is a top concept list)
4: for each labeled sentence $s$ in $d$ do
5:     create a $COG$ for $s$ as in 3.2 algorithm
6:     $c_i$ is a new concept in $s$
7:     for each concept $c_i$ in $s$ do
8:         compute $weight_{stat_i}$ of concept $c_i$
9:         compute $weight_{COG_i}$ of concept $c_i$ based on $L_{COG_i}$
10:        compute $weight_{comb_i}$
11:        add concept $c_i$ to $L$
12:    end for
13: end for
14: sort $L$ descendingly based on $weight_{comb}$
15: return $max(weight_{comb})$ from list $L$
```

The procedure begins with processing a new document (at line 2) which has well defined sentence boundaries. Each sentence is semantically labeled according to [68].

For each labeled sentence (in the for loop at line 4), concepts of the verb argument structures which represent the semantic structures of the sentence are extracted to construct the COG representation (at line 5). The same extracted concepts are weighted by the $weight_{stat}$ (at line 8).

During the concept-based statistical analysis, each concept in the COG representation is also weighted with different value by the $weight_{COG}$ (line 9). The $weight_{stat}$ and the $weight_{COG}$ are combine into the $weight_{comb}$ and added to the concepts list L (at line 10 and 11).

The concepts list L is sorted descendingly based on the $weight_{comb}$ values. The maximum weighted concepts are chosen as top concepts from the concepts list L. (at line 14 and 15)

The concept extractor algorithm is capable of extracting the top concepts in a document (d) in $O(m)$ time, where m is the number of concepts.

3.4.1 Example: Concept Extractor

In this example, for the same sentence *"We have* ***noted*** *how some electronic techniques,* ***developed*** *for the defense effort, have eventually been* ***used*** *in commerce and industry."*

The concept extractor combines the weights computed by the concept-based statistical analyzer and the COG representation into one combined weight. The $weight_{COG}$ values for the concepts that have the *referenced* and *container* types are combined with the $weight_{stat}$ values into new $weight_{comb}$ values.

The concepts that have the maximum $weight_{comb}$ are selected as the top concepts with respect to sentence semantics. In this example, these concepts are *electronic, techniques, developed, defense, effort, commerce,* and *industry.*

The next section presents the concept-based similarity between two documents d1 and d2 that is calculated by the earlier five weight schemes $weight_{tf}$, $weight_{ctf}$, $weight_{stat}$, $weight_{COG}$, and $weight_{comb}$.

3.5 A Concept-Based Similarity Measure

Concepts convey local context information, which is essential in determining an accurate similarity between documents.

A concept-based similarity measure, based on matching concepts at the sentence, document, corpus and combined approach rather than on individual terms (words) only, is devised. The concept-based similarity measure relies on three critical aspects. First, the analyzed labeled terms are the concepts that capture the semantic structure of each sentence. Secondly, the frequency of a concept is used to measure the contribution of the concept to the meaning of the sentence, as well as to the main topics of the document. Lastly, the number of documents that contains the analyzed concepts is used to discriminate among documents in calculating the similarity. These aspects are measured by the proposed concept-based similarity measure which measures the importance of each concept at the sentence-level by the ctf measure, document-level by the tf measure, and corpus-level by the df measure. The concept-based measure exploits the information extracted from the concept-based analysis algorithm to better judge the similarity between the documents. This similarity measure is a function of the following factors:

1. The number of matching concepts, m, in the verb arguments structures in each document (d),

2. The total number of sentences, s, in each document d,
3. The total number of the labeled verb argument structures, v, in each sentence s,
4. The ctf_i of each concept c_i in s for each document d where $(i = 1, 2, ..., m)$ as mentioned in subsections (3.1.1.1 and 3.1.1.2)
5. The tf_i of each concept c_i in each document d where $(i = 1, 2, ..., m)$,
6. The df_i of each concept c_i where $(i = 1, 2, ..., m)$,
7. The length, l, of each concept in the verb argument structure in each document, d, and
8. The length, Lv, of each verb argument structure which contains a matched concept.
9. The total number of documents, N, in the corpus.

The conceptual term frequency (ctf) is an important factor in calculating the concept-based similarity measure between documents. The more frequent the concept appears in the verb argument structures of a sentence in a document, the more conceptually similar the documents are.

The concept-based matching consists of either an exact match or partial match between two concepts. Exact match means that both concepts have the same words. Partial match means that one concept includes all the words that appear in the other concept.

Consider the following concepts $c_1 = "w_1w_2w_3"$ and $c_2 = "w1w2"$ where c_1, c_2 are concepts and w_1, w_2, w_3 are individual words.

After removing stop words, if $c2 \subset c1$ then $c1$ holds more conceptual information than $c2$. In this case, the length of $c1$ is used in the similarity measure between $c1$ and $c2$.

It is important to note that the concept length is only used during the comparison between two concepts and it has nothing to do with identifying the importance of a concept with respect to the sentence semantics. The important concepts with respect to the sentence semantics are identified by the conceptual term frequency (ctf) and the concept-based term frequency (tf) as mentioned in subsections 3.1.1 and 3.1.2.

The concept-based similarity between two documents d_1 and d_2 is calculated by:

$$sim_c(d_1, d_2) = \sum_{i=1}^{m} max(\frac{l_{i_1}}{Lv_{i_1}}, \frac{l_{i_2}}{Lv_{i_2}}) * weight_{i_1} * weight_{i_2}, \tag{3.7}$$

where $weight_i$ can be one of the five weight schemes $weight_{tf_i}$, $weight_{ctf_i}$, $weight_{stat_i}$, $weight_{COG_i}$, and $weight_{comb_i}$. Equation 3.7 assigns a higher score, as the matching concept length approaches the length of its verb argument structure, because this concept tends to hold more conceptual information related to the meaning of its sentence. In equation 3.8, the tf_{ij} value is normalized by the length of the document vector of the term frequency tf_{ij} in document d, where $j = 1, 2, ..., cn$

$$tfweight_i = \frac{tf_{ij}}{\sqrt{\sum_{j=1}^{cn} (tf_{ij})^2}}, \tag{3.8}$$

* cn is the total number of the concepts which has a term frequency value in document d.

* In equation 3.9, the ctf_{ij} value is normalized by the length of the document vector of the conceptual term frequency ctf_{ij} in document d where $j = 1, 2, ..., cn$

$$ctfweight_i = \frac{ctf_{ij}}{\sqrt{\sum_{j=1}^{cn} (ctf_{ij})^2}}, \tag{3.9}$$

where cn is the total number of concepts which has a conceptual term frequency value in document d.

3.5.1 Cosine Measure

For the single-term similarity measure, the cosine correlation similarity measure in [69] is adopted with the popular TF-IDF [3] (Term Frequency/Inverse Document Frequency) term weighting. The cosine measure is chosen due to its wide use in the document clustering literature. Recall that the cosine measure calculates the cosine of the angle between the two document vectors. Accordingly, the single-term similarity measure (sim_s) is:

$$sim_s(d_1, d_2) = \cos(x, y) = \frac{d_1 \cdot d_2}{\| d_1 \| \, \| d_2 \|}, \tag{3.10}$$

The vectors d_1 and d_2 are represented as single-term weights calculated by using the TF-IDF weighting scheme.

3.6 Mathematical Framework

This section consists of three sections. The first section presents a mathematical notation of the concept-based model. A mathematical formulation that compares between the sensitivity of the concept-based similarity and cosine similarity measures is introduced in the second section. The third section presents a comparison between the information content of a concept and its individual word.

3.6.1 Notation

The mathematical notation of the concept-based model is explained as follows:

* List of words (dictionary) is defined as, $dic = w_1, w_2, ..., w_N$, where N is the total number of all words in the dictionary.

* A concept c is a string of words, $c = "w_{i_1} w_{i_2} ... w_{i_n}"$ where n is the total number of words in concept c, where $i_k \in 1, 2, ..., N$ for $k = 1, .., n$ as the repetition of words is allowed.

* A sentence s is a string of concepts, $s = "c_{i_1} c_{i_2} ... c_{i_m}"$, where m is the total number of concepts generated from the verb argument structures in sentence s. Thus, $c_i \subset s$ if c_i is a substring of s

* A document doc is a string of words, $doc = "w_{i_1} w_{i_2} ... w_{i_t}"$, where t is the total number of words in document d, where $i_k \in 1, 2, ..., N$ for $k = 1, .., t$

* The function $freq(str_{sub}, str_{total})$ is the number of times that substring str_{sub} appears in string str_{total}

* The concept-based term frequency of a concept c_i in document doc is $tf_{c_i,doc} = freq(c_i, doc)$

* The conceptual term frequency of a concept c_i in sentence s is $ctf_{c_i,s} = freq(c_i, s)$

* The conceptual term frequency ctf of concept c_i in document doc is calculated by $ctf_{c_i,doc} = \frac{\sum_{m=1}^{sn} ctf_m}{sn}$ where sn is the total number of sentences that contain concept c in document doc as in equation(3.1)

* The concept-based weighting of a concept is $weight_i = (tf_{c_i,doc} + ctf_{c_i,doc}) * \log(\frac{R}{df_i})$, where R is the total number of documents in the corpus and df_i is the number of documents containing concept c_i as in equation(3.6).

3.6.2 Concept-based Sensitivity (Discrimination Ability)

Define $l(c_i)$ = length of concept c_i with respect to words, and
$Lv(c_i)$ = length of the verb argument structure contains a matched concept c_i.

The concept-based similarity between documents doc_1 and doc_2 using concepts is $sim_c(doc_1, doc_2) = \sum_{i=1}^{m} max(\frac{l_{i_1}}{Lv_{i_1}}, \frac{l_{i_2}}{Lv_{i_2}}) * weight_{i_1} * weight_{i_2}$ as calculated in equation(3.7).

Consider that document doc_1 contains concepts $C_1 = c_{1,1}, c_{2,1}, ..., c_{N_1,1}$ where N_1 is the number of concepts in doc_1. Similarly, doc_2 contains concepts $C_2 = c_{1,2}, c_{2,2}, ..., c_{N_2,2}$ where N_2 is the number of concepts in doc_2[3].

Let $C = C_1 \cup C_2 = c_1, ..., c_N$ where $N = |C|$ is the number of unique concepts in C

Define

$$\underline{d_{1c}} = [weight(c_1, doc_1), ..., weight(c_N, doc_1)]^T \tag{3.11}$$

and

$$\underline{d_{2c}} = [weight(c_2, doc_2), ..., weight(c_N, doc_2)]^T \tag{3.12}$$

To compare between the concept-based similarity and the cosine similarity between documents doc_1 and doc_2 with respect to the sensitivity (discrimination ability), assume the following case. First we calculated the similarity between doc_1 and doc_2 using the concept-based and cosine measures. Secondly, assume that doc_2 is changed by $\triangle$. Lastly, we calculate the similarity between doc_1 and doc_2 again using concept-based and cosine similarities to find out which similarity measure is more sensitive to the change that happened to document doc_2.

The sensitivity (discrimination ability) of the concept-based similarity is calculated by:

$$S_{concept}(\underline{d_{1c}}, \underline{d_{2c}} + \triangle\underline{d_{2c}}) - S_{concept}(\underline{d_{1c}}, \underline{d_{2c}}) = \underline{d_{1c}}^T A(\underline{d_{2c}} + \triangle\underline{d_{2c}}) - \underline{d_{1c}}^T A\underline{d_{2c}} \tag{3.13}$$

where $A_{ij} = max(\frac{l_{c_i}}{Lv_{c_i}}, \frac{l_{c_j}}{Lv_{c_j}})$, $i = 1, .., N$ and $j = 1, ..., N$.

For the single-term, The cosine similarity between documents d_1 and d_2 is $sim_s(d_1, d_2) = \cos(x, y) = \frac{d_1 \cdot d_2}{\|d_1\| \ \|d_2\|}$ as calculated in equation(3.10). The TFIDF weight of the single-term is $TFIDF = tf_i * \log(\frac{N}{df_i})$. The sensitivity (discrimination ability) of the cosine similarity is calculated by:

$$S_{cosine}(\underline{d_{1w}}, \underline{d_{2w}} + \triangle\underline{d_{2w}}) - S_{cosine}(\underline{d_{1w}}, \underline{d_{2w}}) = \underline{d_{1w}}^T \triangle \underline{d_{2w}} \tag{3.14}$$

[3]In this section, we dropped the permutation notation for simplicity

where d_{1w} and d_{2w} are vectors presented by TF-IDF weight.

$$S_{concept}(\underline{d_{1c}}, \underline{d_{2c}} + \triangle\underline{d_{2c}}) - S_{concept}(\underline{d_{1c}}, \underline{d_{2c}}) = \underline{d_{1c}}^T A \triangle \underline{d_{2c}} \tag{3.15}$$

To find relations between d_{1c} and d_{1w} and between $\triangle\underline{d_{2w}}$ and $\triangle\underline{d_{2c}}$ in equations(3.14,3.15), assume that each concept is a word and each entry in A is equal to one since both the concept length l_{c_i} and the verb argument structure length Lv_{c_i} are equal to one.

From (3.15) and after simple manipulations,

$$S_{concept}(\underline{d_{1c}}, \underline{d_{2c}} + \triangle\underline{d_{2c}}) - S_{concept}(\underline{d_{1c}}, \underline{d_{2c}}) = (\sum_{i=1}^{N} weight(c_i, doc_1))(\sum_{i=1}^{N} weight(c_i, doc_2)) \tag{3.16}$$

$> \sum_{i=1}^{N} weight(c_i, doc_1) weight(c_i, doc_2) = \underline{d_{1c}}^T \underline{d_{2c}}$ since all components of d_{1c} and d_{2c} are positive.

At the same time, since the concept-based weighting of a concept is $weight_i = (tfi + ctfi) * \log(\frac{N}{df_i})$ and the TFIDF weighting of a single-term is $TFIDF = tf_i * \log(\frac{N}{df_i})$, it is clear that the ctf_i is an added value to the concept-based weighting.

By approximation, the d_{1c} value is bigger than d_{1w} and the $\triangle\underline{d_{2c}}$ value is bigger than the $\triangle\underline{d_{2w}}$ value. This is due to the fact that the concept-based weighting has ctf_i value as an additional measure to the $TFIDF$ weighting. Hence, the sensitivity (discrimination ability) of the concept-based similarity is higher than the cosine similarity. This means that the concept-based model is deeper in analyzing the similarity between two documents than the traditional approaches. This is due to the fact that the analysis is achieved on both the document and the sentence levels.

The sensitivity (discrimination ability) of the concept-based similarity is higher than the cosine similarity in case that each concept is a word. However, each concept usually consists of more than one word which means that the information content of a concept is higher than its individual words.

3.6.3 Concept-based Entropy

The entropy of a concept is higher than the sum of the entropy of its individual words. This is formally proven in the following theorem.

Consider a document doc containing words from a dictionary $dict = w_1, ..., w_N$ and concepts $c_1, ..., c_M$ of length k words where M is any number of concepts. Define the concept conditional entropy as:

$$I_k(c) = -\sum_{j=1}^{M} p(c_j|doc)\log p(c_j|doc) \tag{3.17}$$

and the word conditional entropy as

$$I(w) = -\sum_{j=1}^{N} p(w_i|doc)\log p(w_i|doc) \tag{3.18}$$

then we want to prove that $I_k(c) \geq I(w)$

Proof

Any concept c_j of length k words is generally represented as a string of k words selected from the dictionary. i.e. $c_j = "w_{i_1}j...w_{i_k}j"$. Hence, the summation in equation (3.17) is equivalent to taking the summation over all permutations of the words in the dictionary (with repetition) and size k words, i.e.

$$I(c) = -\sum_{i_1,i_2,...,i_k} p(w_{i_1}, w_{i_2}, ..., w_{i_k}|doc)\log p(w_{i_1}, ..., w_{i_k}|doc) \tag{3.19}$$

where $\sum_{i_1,i_2,...,i_k}$ is a short hand for $\sum_{i_1=1}^{N}\sum_{i_k=1}^{N}$. Using the definition of conditional probability of equation (3.19) can be simplified to:

$$I_k(c) = -\sum_{i_1,i_2,...,i_k} p(w_{i_1}|doc)p(w_{i_2}, ..., w_{i_k}|w_{i_1}, doc)[\log p(w_{i_1}|doc) + \log(p(w_{i_2}, ..., w_{i_k}|w_{i_1}, doc))] \tag{3.20}$$

$I_k(c) = -\sum_{i_1} p(w_{i_1}|doc)\log(p(w_{i_1}|doc))\sum_{i_2,...,i_k} p(w_{i_2}, ..., w_{i_k}|w_{i_1}, doc)$
$-\sum_{i_1,i_2,...,i_k} p(w_{i_1}|doc)p(w_{i_2}, ..., w_{i_k}|w_{i_1}, doc)\log(p(w_{i_2}, ..., w_{i_k}|w_{i_1}, doc))$

but

$$\sum_{i_2,...,i_k} p(w_{i_2}, ..., w_{i_k}|w_{i_1}, doc) = 1 \tag{3.21}$$

and

$$-\sum_{i_1,i_2,...,i_k} p(w_{i_1}|doc)p(w_{i_2}, ..., w_{i_k}|w_{i_1}, doc)\log(p(w_{i_2}, ..., w_{i_k}|w_{i_1}, doc)) > 0 \tag{3.22}$$

Then

$$-\sum_{i_1,i_2,...,i_k} p(w_{i_1}|doc)p(w_{i_2},...,w_{i_k}|w_{i_1},doc)\log(p(w_{i_2},...,w_{i_k}|w_{i_1},doc)) > -\sum_{i_1} p(w_{i_1}|doc)\log(p(w_{i_1}|doc)) \tag{3.23}$$

$$I_k(c) \geq -\sum_{i_1} p(w_{i_1}|doc)\log(p(w_{i_1}|doc)) = I(w) \tag{3.24}$$

3.7 Summary

In this chapter, a new concept-based model is introduced. The concept-based model consists of concept-based statistical analyzer, conceptual ontological graph (COG) representation, concept extractor, and concept-based similarity measure. Moreover, three algorithms are presented which are (1) concept-based statistical analysis algorithm, (2) COG constructor algorithm and (3) concept extractor algorithm. The concept-based statistical analysis algorithm analyzes terms on the sentence, document and corpus levels. The COG constructor algorithm provides the required steps for creating the COG representation. the concept extractor algorithm combines between the weights of the statistical analyzer and the COG representation to choose the best concepts out of the two approaches. In addition, an illustrative example is introduced to show the input and the output of each component. Moreover, a concept-based measure that measures the similarity between two documents and a mathematical framework of the concept-based model are presented.

Chapter 4

Experimental Results

This chapter presents the experimental work of applying the concepts extracted by the concept-based model in text clustering, categorization, and retrieval. The chapter introduces a comparison among five weighings $weight_{tf}$, $weight_{ctf}$, $weight_{stat}$, $weight_{cog}$, and $weight_{combind}$ in text clustering, categorization, and retrieval applications. We discuss the results of the concept-based model compared to the traditional techniques. In addition, some observations are reported regarding the text clustering, categorization and retrieval quality achieved by the concept-based model.

4.1 Overview

In order to address the stability of the algorithm, data variation and to test the effectiveness of using the concepts extracted by the proposed concept-based model as an accurate measure to weight terms in the document. Extensive sets of experiments for a large scale evaluation of the proposed model in text clustering, categorization, and retrieval are conducted. In the experiments, we are comparing with ground truth. In the datasets, the text directly is analyzed, rather than, using metadata associated with the text documents. This clearly demonstrates the effect of using concepts on the text mining process.

For the single-term weighting, the popular TF-IDF [3] (Term Frequency/Inverse Document Frequency) term weighting is adopted. The TF-IDF weighting is chosen due to its wide use in the document clustering, categorization, and retrieval literature.

The concept-based weighting is one of the main factors that captures the importance of a concept in a document. Thus, to study the effect of the concept-based weighting on the the quality of the text clustering, categorization, retrieval techniques, the entire sets of the experiments are repeated using the following concept-based weighting schemes:

- The $weight_{tf_i}$ as in equation (3.8)
- The $weight_{ctf_i}$ as in equation (3.9)
- The $weight_{stat_i}$ as in equation (3.2)
- The $weight_{cog_i}$ as in equation(3.5)
- The $weight_{combind_i}$ as in equation(3.6)

Table 4.1: Clustering Improvement using Concept-based Term Frequency ($weight_{tf}$)

DataSet		***Single-Term***		***Concept-based***		***Improvement***
		F-measure	Entropy	F-measure	Entropy	
Reuters	HAC (ward)	0.723	0.251	0.782	0.113	+8.16%F, -54.98%E
	HAC (complete)	0.623	0.315	0.766	0.121	+22.95%F, -61.58%E
	Single Pass	0.411	0.523	0.623	0.279	+51.58%F, -46.65%E
	k-NN	0.511	0.348	0.736	0.174	+44.03%F, -50.0%E
ACM	HAC (ward)	0.697	0.317	0.741	0.178	+6.31%F, -43.84%E
	HAC (complete)	0.481	0.362	0.712	0.188	+48.02%F, -48.06%E
	Single Pass	0.398	0.608	0.607	0.316	+52.51%F, -48.02%E
	k-NN	0.491	0.402	0.697	0.235	+41.95%F, -41.54%E
Brown	HAC (ward)	0.581	0.385	0.688	0.136	+18.41%F, -64.67%E
	HAC (complete)	0.547	0.401	0.653	0.211	+19.37%F, -47.38%E
	Single Pass	0.437	0.551	0.611	0.263	+39.81%F, -52.26%E
	k-NN	0.462	0.316	0.657	0.149	+42.20%F, -52.84%E
20 Newsgroups	HAC (ward)	0.535	0.316	0.661	0.152	+23.55%F, -51.89%E
	HAC (complete)	0.471	0.345	0.643	0.243	+36.51%F, -29.56%E
	Single Pass	0.312	0.643	0.524	0.371	+67.94%F, -42.30%E
	k-NN	0.462	0.457	0.621	0.256	+34.41%F, -43.98%E

Table 4.2: Clustering Improvement using Conceptual Term Frequency ($weight_{ctf}$)

DataSet		***Single-Term***		***Concept-based***		***Improvement***
		F-measure	Entropy	F-measure	Entropy	
Reuters	HAC (ward)	0.723	0.251	0.892	0.062	+23.37%F, -75.29%E
	HAC (complete)	0.623	0.315	0.817	0.068	+31.13%F, -78.41%E
	Single Pass	0.411	0.523	0.783	0.125	+90.51%F, -76.09%E
	k-NN	0.511	0.348	0.861	0.066	+68.49%F, -81.03%E
ACM	HAC (ward)	0.697	0.317	0.883	0.054	+26.68%F, -82.96%E
	HAC (complete)	0.481	0.362	0.843	0.173	+75.25%F, -52.20%E
	Single Pass	0.398	0.608	0.764	0.201	+91.95%F, -66.94%E
	k-NN	0.491	0.402	0.845	0.143	+72.09%F, -64.42%E
Brown	HAC (ward)	0.581	0.385	0.876	0.083	+50.77%F, -78.44%E
	HAC (complete)	0.547	0.401	0.852	0.127	+55.75%F, -68.32%E
	Single Pass	0.437	0.551	0.726	0.133	+66.13%F, -75.86%E
	k-NN	0.462	0.316	0.843	0.118	+82.46%F, -62.65%E
20 Newsgroups	HAC (ward)	0.535	0.316	0.852	0.064	+59.25%F, -79.74%E
	HAC (complete)	0.471	0.345	0.823	0.082	+74.73%F, -76.23%E
	Single Pass	0.312	0.643	0.704	0.153	>+100%F, -76.20%E
	k-NN	0.462	0.457	0.815	0.121	+76.40%F, -73.52%E

4.2 Text Clustering

The experimental setup consisted of four datasets. The first data set contains 23,115 ACM abstract articles collected from the ACM digital library. The ACM articles are classified according to the ACM computing classification system into five main categories: general literature, hardware, computer systems organization, software, and data. The second data set has 12,902 documents from the Reuters 21578 dataset. There are 9,603 documents in the training set and 3,299 documents in the test set. Out of the 5 category sets, the topic category set contains 135 categories, but only 90 categories have at least one document in the training set. These 90 categories were used in the experiment. The third dataset consisted of 361 samples from the Brown corpus [70]. Each

Table 4.3: Clustering Improvement using Concept-based Statistical Analyzer ($weight_{stat}$)

DataSet		***Single-Term***		***Concept-based***		***Improvement***
		F-measure	Entropy	F-measure	Entropy	
Reuters	HAC (ward)	0.723	0.251	0.925	0.012	+27.93%F, -95.21%E
	HAC (complete)	0.623	0.315	0.907	0.025	+45.58%F, -92.06%E
	Single Pass	0.411	0.523	0.816	0.067	+98.54%F, -87.18%E
	k-NN	0.511	0.348	0.917	0.015	+79.45%F, -95.68%E
ACM	HAC (ward)	0.697	0.317	0.918	0.043	+31.70%F, -86.43%E
	HAC (complete)	0.481	0.362	0.895	0.135	+86.07%F, -62.7%E
	Single Pass	0.398	0.608	0.791	0.152	+98.74%F, -75.0%E
	k-NN	0.491	0.402	0.891	0.111	+81.46%F, -72.38%E
Brown	HAC (ward)	0.581	0.385	0.906	0.018	+55.93%F, -95.32%E
	HAC (complete)	0.547	0.401	0.901	0.021	+64.71%F, -94.76%E
	Single Pass	0.437	0.551	0.804	0.045	+83.98%F, -91.83%E
	k-NN	0.462	0.316	0.902	0.023	+95.23%F, -92.72%E
20 Newsgroups	HAC (ward)	0.535	0.316	0.901	0.035	+68.41%F, -88.92%E
	HAC (complete)	0.471	0.345	0.892	0.074	+89.38%F, -78.55%E
	Single Pass	0.312	0.643	0.773	0.087	>+100%F, -86.46%E
	k-NN	0.462	0.457	0.865	0.065	+87.22%F, -85.77%E

sample has 2000+ words. The Brown corpus main categories used in the experiment were: press: reportage, press: reviews, religion, skills and hobbies, popular lore, belles-letters, learned, fiction: science, fiction: romance, and humor. The fourth dataset consists of 20,000 messages collected from 20 Usenet newsgroups.

The similarities which are calculated by using the concept-based model are used to compute four similarity matrices among documents as shown in Figure(4.1). Three standard document clustering techniques are chosen for testing the effect of the concept-based similarity on clustering [71]: (1) Hierarchical Agglomerative Clustering (HAC), (2) Single Pass Clustering, and (3) k-Nearest Neighbor (k-NN)[3].

[3]Though k-NN is mostly known to be used for classification, it has also been used for clustering (example could

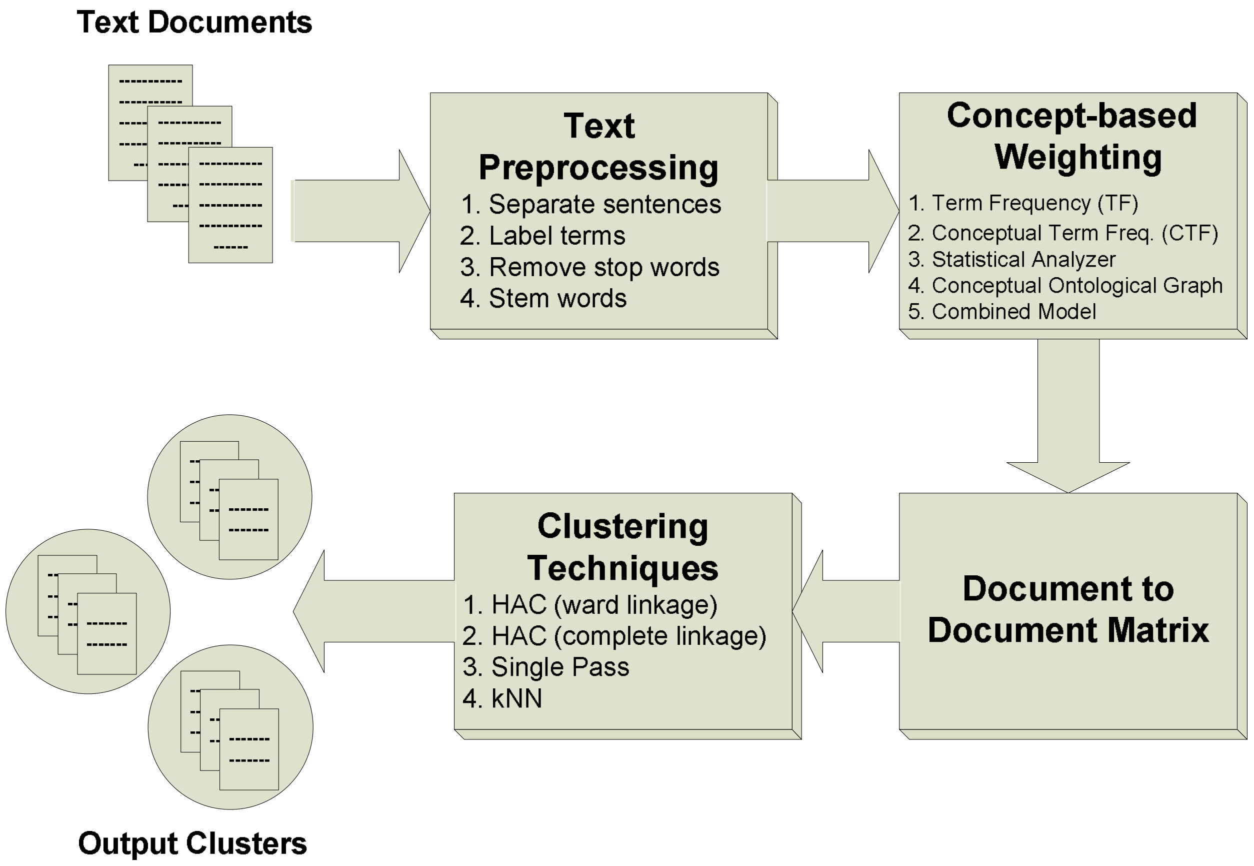

Figure 4.1: Concept-based Model For Text Clustering

Table 4.4: Clustering Improvement using Conceptual Ontological Graph ($weight_{cog}$)

DataSet		***Single-Term***		***Concept-based***		***Improvement***
		F-measure	Entropy	F-measure	Entropy	
Reuters	HAC (ward)	0.723	0.251	0.914	0.038	+26.41%F, -84.86%E
	HAC (complete)	0.623	0.315	0.897	0.045	+43.98%F, -85.71%E
	Single Pass	0.411	0.523	0.792	0.086	+92.70%F, -83.55%E
	k-NN	0.511	0.348	0.897	0.042	+75.53%F, -87.93%E
ACM	HAC (ward)	0.697	0.317	0.927	0.031	+32.99%F, -90.22%E
	HAC (complete)	0.481	0.362	0.915	0.062	+90.22%F, -82.87%E
	Single Pass	0.398	0.608	0.824	0.085	>+100%F, -86.01%E
	k-NN	0.491	0.402	0.916	0.047	+86.55%F, -88.30%E
Brown	HAC (ward)	0.581	0.385	0.924	0.013	+59.03%F, -96.62%E
	HAC (complete)	0.547	0.401	0.916	0.018	+67.45%F, -95.51%E
	Single Pass	0.437	0.551	0.857	0.029	+96.10%F, -94.73%E
	k-NN	0.462	0.316	0.918	0.015	+98.70%F, -95.25%E
20 Newsgroups	HAC (ward)	0.535	0.316	0.885	0.057	+65.42%F, -81.96%E
	HAC (complete)	0.471	0.345	0.857	0.079	+81.95%F, -77.10%E
	Single Pass	0.312	0.643	0.759	0.094	>+100%F, -85.38%E
	k-NN	0.462	0.457	0.843	0.081	+82.46%F, -82.27%E

In order to evaluate the quality of the clustering, two quality measures widely used in the text mining literature for the purpose of document clustering [9] are adopted. The first is the ***F-measure***, which combines the Precision and Recall measures from the Information Retrieval literature. The precision P and recall R of a cluster j with respect to a class i are defined as:

$$P = Precision(i, j) = \frac{M_{ij}}{M_j}, \tag{4.1}$$

$$R = Recall(i, j) = \frac{M_{ij}}{M_i}, \tag{4.2}$$

be found in [72]).

Table 4.5: Clustering Improvement using The Concept-based Combined Model $weight_{comb}$

DataSet		***Single-Term***		***Concept-based***		***Improvement***
		F-measure	Entropy	F-measure	Entropy	
Reuters	HAC (ward)	0.723	0.251	0.938	0.0075	+29.73%F, -97.01%E
	HAC (complete)	0.623	0.315	0.921	0.017	+47.83%F, -94.60%E
	Single Pass	0.411	0.523	0.857	0.032	>+100%F, -93.88%E
	k-NN	0.511	0.348	0.926	0.0083	+81.21%F, -97.61%E
ACM	HAC (ward)	0.697	0.317	0.936	0.014	+34.28%F, -95.58%E
	HAC (complete)	0.481	0.362	0.925	0.031	+92.30%F, -91.43%E
	Single Pass	0.398	0.608	0.851	0.054	>+100%F, -91.11%E
	k-NN	0.491	0.402	0.927	0.026	+88.79%F, -93.53%E
Brown	HAC (ward)	0.581	0.385	0.931	0.0064	+60.24%F, -98.33%E
	HAC (complete)	0.547	0.401	0.927	0.0082	+69.46%F, -97.95%E
	Single Pass	0.437	0.551	0.894	0.017	>+100%F, -96.91%E
	k-NN	0.462	0.316	0.928	0.0075	>+100%F, -97.62%E
20 Newsgroups	HAC (ward)	0.535	0.316	0.923	0.013	+72.52%F, -95.88%E
	HAC (complete)	0.471	0.345	0.908	0.037	+92.78%F, -89.27%E
	Single Pass	0.312	0.643	0.843	0.041	>+100%F, -93.62%E
	k-NN	0.462	0.457	0.916	0.032	+98.26%F, -92.99%E

M_{ij}: is the number of members of class i in cluster j,
M_j: is the number of members of cluster j, and
M_i: is the number of members of class i.

The F-measure of a class i is defined as:

$$F(i) = \frac{2PR}{P+R}. \tag{4.3}$$

With respect to class i, the cluster with the highest F-measure is considered to be the cluster that maps to class i, and that F-measure becomes the score for class i. The overall F-measure for the clustering result C is the weighted average of the F-measure for each class i:

$$F_C = \frac{\sum_i (|i| \times F(i))}{\sum_i |i|}, \tag{4.4}$$

where $|i|$ is the number of objects in class i. The higher the overall F-measure, the better the clustering, due to the higher accuracy of the clusters mapping to the original classes.

The second measure is the ***Entropy***, which provides a measure of quality for unnested clusters or for the clusters at one level of a hierarchical clustering. Entropy measures how homogeneous a cluster is. The higher the homogeneity of a cluster, the lower the entropy is, and vice versa. The entropy of a cluster containing only one object (perfect homogeneity) is zero.

For every cluster j in the clustering result C, the probability p_{ij} that a member of cluster j belongs to class i is computed. The entropy of each cluster j is calculated using the standard formula $E_j = -\sum_i p_{ij} log(p_{ij})$, where the sum is taken over all classes. The total entropy for a set of clusters is calculated as the sum of entropies of each cluster weighted by the size of that cluster:

$$E_C = \sum_{j=1}^{n} (\frac{M_j}{M} \times E_j), \tag{4.5}$$

where M_j is the size of cluster j, and M is the total number of data objects.

Usually, internal measures are used to evaluate the clustering quality. However, we have the clusters labels and the ground truth. Thus, we intend to use external measures to validate the concept-based model.

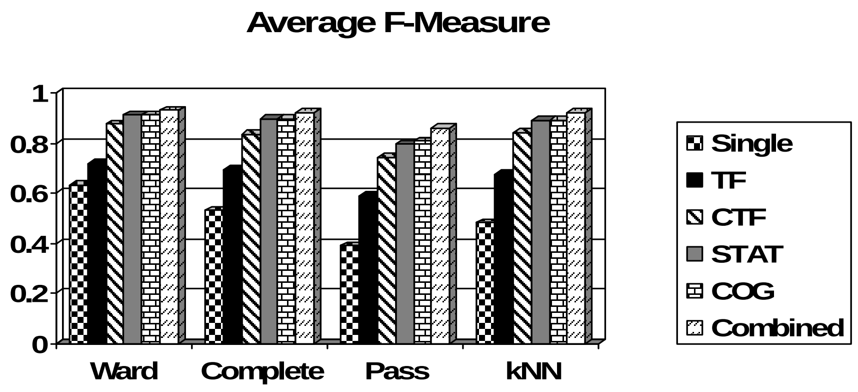

Figure 4.2: Clustering Improvements (F-Measure)

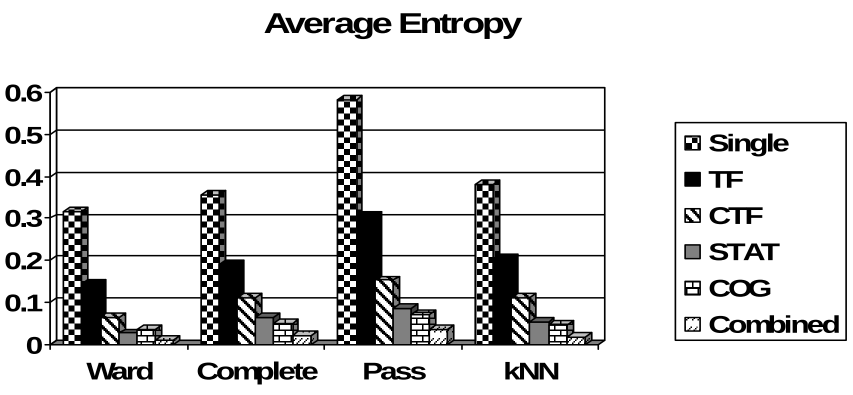

Figure 4.3: Clustering Improvements (Entropy)

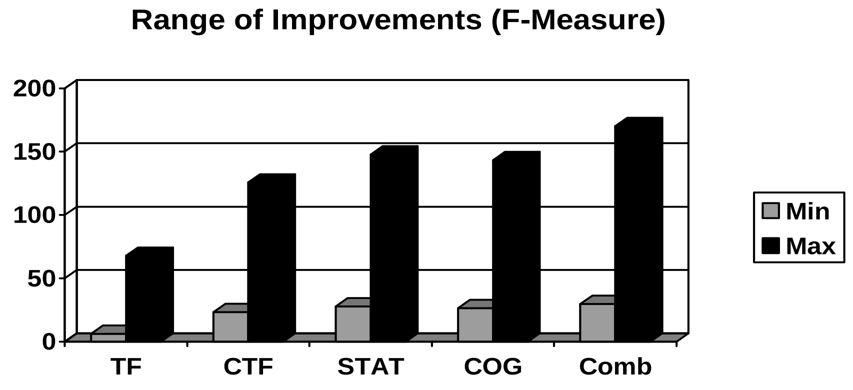

Figure 4.4: Clustering Range of Improvements (F-Measure)

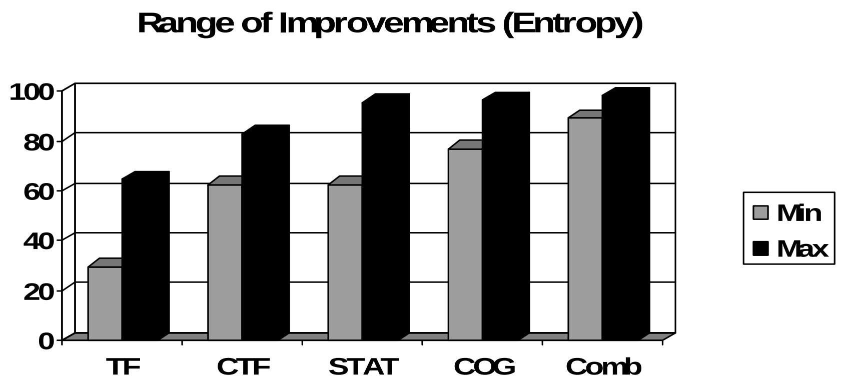

Figure 4.5: Clustering Range of Improvements (Entropy)

Table 4.6: Text Categorization Improvement using Concept-based Term Frequency ($weight_{tf}$)

DataSet		***Single-Term***			***Concept-based***			***Improvement***
		Micro Avg(F1)	Macro Avg(F1)	Avg Error	Micro Avg(F1)	Macro Avg(F1)	Avg Error	
Reuters	SVM	0.9023	0.7421	0.0871	0.9065	0.7843	0.0794	+0.46%, +5.68%, -8.84%
	NB	0.8132	0.6127	0.2754	0.8214	0.6725	0.1522	+1.00%, +9.76%, -44.73%
	Rocchio	0.8543	0.6513	0.1632	0.8612	0.7031	0.1173	+0.80%, +7.95%, -28.12%
	kNN	0.8736	0.6865	0.1074	0.8747	0.7256	0.0835	+0.12%, +5.69%, -22.25%
ACM	SVM	0.6851	0.4973	0.1782	0.7165	0.5823	0.1652	+4.58%, +17.09%, -7.29%
	NB	0.6134	0.4135	0.4215	0.6522	0.5217	0.3584	+6.32%, +26.16%, -14.9%
	Rocchio	0.6327	0.4826	0.2733	0.6611	0.5643	0.2463	+4.48%, +16.92%, -9.87%
	kNN	0.6543	0.4952	0.2103	0.6986	0.5724	0.1869	+6.77%, +15.58%, -11.1%
Brown	SVM	0.8537	0.6143	0.1134	0.8653	0.6925	0.0813	+1.35%, +12.72%, -28.3%
	NB	0.7864	0.5071	0.3257	0.8094	0.6289	0.2147	+2.92%, +24.01%, -34.0%
	Rocchio	0.8067	0.5728	0.2413	0.8263	0.6574	0.1584	+2.42%, +14.76%, -34.3%
	kNN	0.8291	0.5934	0.1256	0.8422	0.6853	0.0936	+1.58%, +15.48%, -25.4%
Newsgroups	SVM	0.7951	0.5923	0.1325	0.8125	0.6218	0.0892	+2.18%, +4.98%, -32.67%
	NB	0.7023	0.4975	0.3621	0.7325	0.5643	0.2841	+4.30%, +13.4%, -21.54%
	Rocchio	0.7124	0.5346	0.2571	0.7432	0.5735	0.1463	+4.32%, +7.27%, -43.09%
	kNN	0.7341	0.5581	0.1423	0.7623	0.5946	0.0976	+3.84%, +6.54%, -31.41%

4.3 Text Categorization

The experimental setup consisted of four datasets. The first data set contains 23,115 ACM abstract articles collected from the ACM digital library. The ACM articles are classified according to the ACM computing classification system into five main categories: general literature, hardware, computer systems organization, software, and data. The second data set has 12,902 documents from the Reuters 21578 dataset. There are 9,603 documents in the training set and 3,299 documents in the test set. Out of the 5 category sets, the topic category set contains 135 categories, but only 90 categories have at least one document in the training set. These 90 categories were used in the experiment. The third dataset consisted of 361 samples from the Brown corpus [70]. Each

Table 4.7: Text Categorization Improvement using Conceptual Term Frequency ($weight_{ctf}$)

DataSet		***Single-Term***			***Concept-based***			***Improvement***
		Micro Avg(F1)	Macro Avg(F1)	Avg Error	Micro Avg(F1)	Macro Avg(F1)	Avg Error	
Reuters	SVM	0.9023	0.7421	0.0871	0.9143	0.8176	0.0647	+1.32%, +10.17%, -25.71%
	NB	0.8132	0.6127	0.2754	0.8652	0.7524	0.0925	+6.39%, +22.8%, -66.41%
	Rocchio	0.8543	0.6513	0.1632	0.8814	0.7689	0.0874	+3.17%, +18.05%, -46.44%
	kNN	0.8736	0.6865	0.1074	0.8975	0.7936	0.0731	+2.73%, +15.60%, -31.93%
ACM	SVM	0.6851	0.4973	0.1782	0.7428	0.6475	0.1458	+8.42%, +30.21%, -18.1%
	NB	0.6134	0.4135	0.4215	0.6831	0.5914	0.2536	+11.3%, +43.03%, -39.83%
	Rocchio	0.6327	0.4826	0.2733	0.6937	0.6073	0.1847	+9.64%, +25.85%, -32.41%
	kNN	0.6543	0.4952	0.2103	0.7132	0.6289	0.1645	+9.0% , +27.0%, -21.77%
Brown	SVM	0.8537	0.6143	0.1134	0.8892	0.7641	0.0742	+4.15%, +24.39%, -34.56%
	NB	0.7864	0.5071	0.3257	0.8357	0.6985	0.1385	+6.26%, +37.74%, -57.47%
	Rocchio	0.8067	0.5728	0.2413	0.8611	0.7162	0.0926	+6.74%, +25.04%, -61.62%
	kNN	0.8291	0.5934	0.1256	0.8753	0.7483	0.0883	+5.57%, +26.11%, -29.69%
Newsgroups	SVM	0.7951	0.5923	0.1325	0.8536	0.7143	0.0835	+7.357%, +20.59%, -36.98%
	NB	0.7023	0.4975	0.3621	0.7824	0.6524	0.1683	+11.40%, +31.13%, -53.52%
	Rocchio	0.7124	0.5346	0.2571	0.7968	0.6872	0.0862	+11.84%, +28.54%, -66.4%
	kNN	0.7341	0.5581	0.1423	0.8137	0.6953	0.0851	+10.84%, +24.58%, -62.15%

sample has 2000+ words. The Brown corpus main categories used in the experiment were: press: reportage, press: reviews, religion, skills and hobbies, popular lore, belles-letters, learned, fiction: science, fiction: romance, and humor. The fourth dataset consists of 20,000 messages collected from 20 Usenet newsgroups.

For each dataset, stop words are removed from the concepts that are extracted by the proposed model. The extracted concepts are stemmed using the Porter stemmer algorithm [7]. Concepts are used to build standard normalized feature vectors using the standard vector space model for document representation.

The concept-based weights which are calculated by the concept-based model are used to com-

Table 4.8: Text Categorization Improvement using Concept-based Statistical Analyzer ($weight_{stat}$)

DataSet		***Single-Term***			***Concept-based***			***Improvement***
		Micro Avg(F1)	Macro Avg(F1)	Avg Error	Micro Avg(F1)	Macro Avg(F1)	Avg Error	
Reuters	SVM	0.9023	0.7421	0.0871	0.9254	0.8521	0.0523	+2.56%, +14.82%, -39.95%
	NB	0.8132	0.6127	0.2754	0.8963	0.7843	0.0674	+10.21%, +27.97%, -72.62%
	Rocchio	0.8543	0.6513	0.1632	0.9052	0.8062	0.0625	+5.95%, +23.78%, -61.7%
	kNN	0.8736	0.6865	0.1074	0.9174	0.8153	0.0541	+5.01%, +18.76%, -49.62%
ACM	SVM	0.6851	0.4973	0.1782	0.7621	0.6854	0.1264	+11.23%, +37.82%, -87.36%
	NB	0.6134	0.4135	0.4215	0.7063	0.6213	0.2064	+15.14%, +37.87%, -79.36%
	Rocchio	0.6327	0.4826	0.2733	0.7248	0.6457	0.1537	+14.55%, +33.79%, -43.76%
	kNN	0.6543	0.4952	0.2103	0.7415	0.6821	0.1325	+13.32%, +37.74%, -36.99%
Brown	SVM	0.8537	0.6143	0.1134	0.9037	0.8054	0.0637	+5.85%, +31.10%, -43.82%
	NB	0.7864	0.5071	0.3257	0.8461	0.7613	0.0891	+7.95%, +50.12%, -72.64%
	Rocchio	0.8067	0.5728	0.2413	0.8837	0.7824	0.0765	+9.54%, +36.59%, -68.29%
	kNN	0.8291	0.5934	0.1256	0.8965	0.7965	0.0652	+8.12%, +34.22%, -48.08%
Newsgroups	SVM	0.7951	0.5923	0.1325	0.8723	0.7835	0.0534	+9.7%, +32.28%, -59.69%
	NB	0.7023	0.4975	0.3621	0.8134	0.7253	0.0721	+15.81%, +45.78%, -80.08%
	Rocchio	0.7124	0.5346	0.2571	0.8503	0.7481	0.0683	+19.35%, +39.93%, -73.43%
	kNN	0.7341	0.5581	0.1423	0.8647	0.7524	0.0542	+17.79%, +34.81%, -56.44%

pute a document-concept matrix between documents and concepts as shown in Figure(4.6). Four standard document categorization techniques are chosen for testing the effect of the concepts on categorization quality: (1) Support Vector Machine (SVM), (2)Rocchio, (3) Naive Bayesian (NB), and (4) k-Nearest Neighbor (k-NN). These techniques are used as binary classifiers in which they recognize documents from one specific topic against all other topics. This setup was repeated for every topic.

In order to evaluate the quality of the text categorization, three widely used evaluation measures in document categorization and retrieval literatures are computed with 5-fold cross validation for each classifier. These measures are the Macro-averaged performance F1 measure (the harmonic mean of precision and recall), the Micro-averaged performance F1 measure, and the error rate.

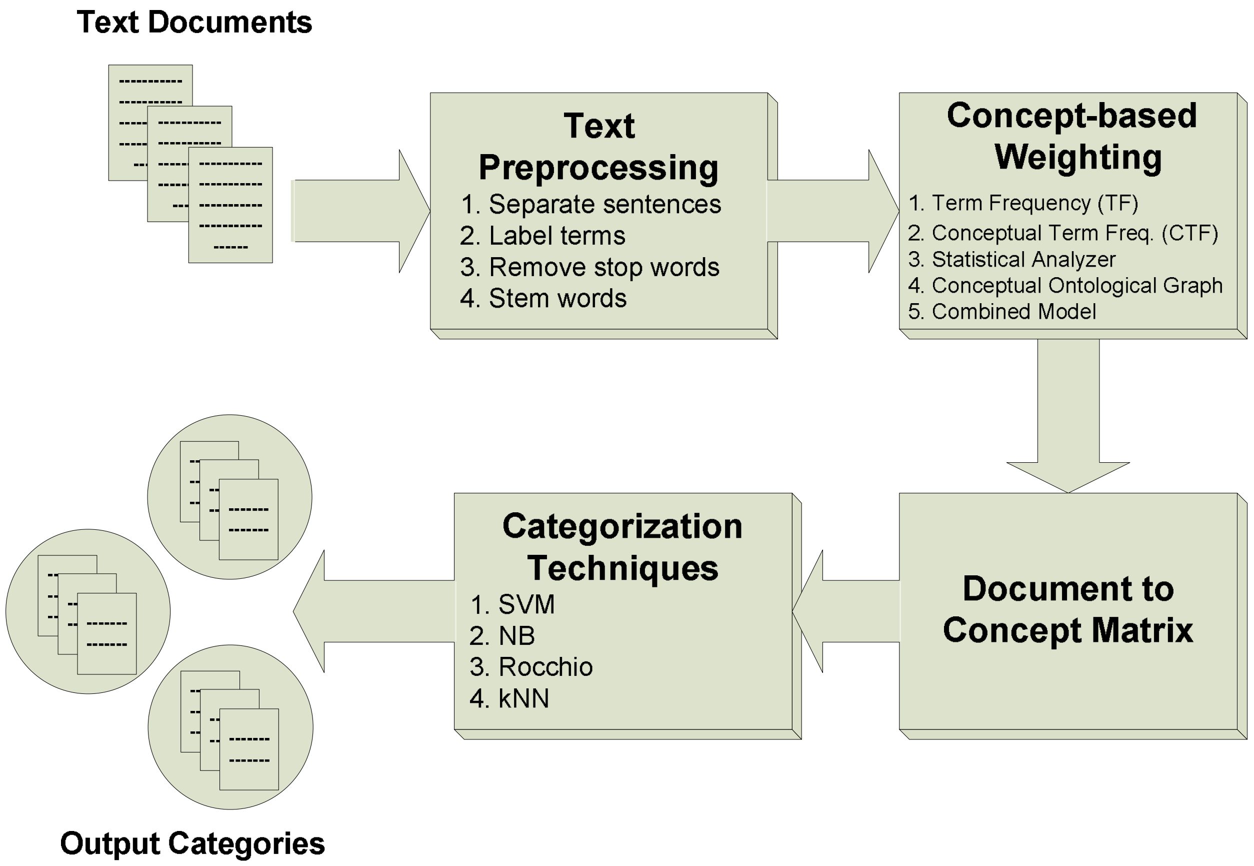

Figure 4.6: Concept-based Model For Text Categorization

Table 4.9: Text Categorization Improvement using Conceptual Ontological Graph ($weight_{COG}$)

DataSet		***Single-Term***			***Concept-based***			***Improvement***
		Micro Avg(F1)	Macro Avg(F1)	Avg Error	Micro Avg(F1)	Macro Avg(F1)	Avg Error	
Reuters	SVM	0.9023	0.7421	0.0871	0.9173	0.8362	0.0634	+1.66%, +12.68%, -27.21%
	NB	0.8132	0.6127	0.2754	0.8842	0.7581	0.0825	+8.73%, +23.73%, -74.94%
	Rocchio	0.8543	0.6513	0.1632	0.8963	0.7826	0.0716	+4.91%, +20.15%, -56.12%
	kNN	0.8736	0.6865	0.1074	0.9125	0.8043	0.0634	+4.45%, +17.15%, -40.96%
ACM	SVM	0.6851	0.4973	0.1782	0.7835	0.7163	0.1162	+14.36%, +44.03%, -34.79%
	NB	0.6134	0.4135	0.4215	0.7142	0.6542	0.1875	+16.43%, +58.21%, -55.51%
	Rocchio	0.6327	0.4826	0.2733	0.7356	0.6621	0.1246	+16.26%, +37.19%, -54.40%
	kNN	0.6543	0.4952	0.2103	0.7629	0.7063	0.1143	+16.59%, +17.95%, -45.64%
Brown	SVM	0.8537	0.6143	0.1134	0.9164	0.8326	0.0521	+7.34%, +35.53%, -54.05%
	NB	0.7864	0.5071	0.3257	0.8655	0.7754	0.0746	+10.05%, +52.90%, -77.09%
	Rocchio	0.8067	0.5728	0.2413	0.8943	0.8062	0.0653	+10.85%, +40.74%, -72.93%
	kNN	0.8291	0.5934	0.1256	0.9015	0.8174	0.0547	+ 8.73%, +37.74%, -40.19%
Newsgroups	SVM	0.7951	0.5923	0.1325	0.8614	0.7762	0.0752	+ 8.33%, +31.04%, -43.24%
	NB	0.7023	0.4975	0.3621	0.8053	0.7041	0.0874	+14.66%, +41.52%, -75.86%
	Rocchio	0.7124	0.5346	0.2571	0.8362	0.7365	0.0846	+17.37%, +37.76%, -67.09%
	kNN	0.7341	0.5581	0.1423	0.8451	0.7416	0.0835	+15.12%, +32.87%, -41.32%

Recall that in binary classification (relevant/not relevant), the following quantities are considered:

- p^+ = the number of relevant documents, classified as relevant.
- p^- = the number of relevant documents, classified as not relevant.
- n^- = the number of not relevant documents, classified as not relevant.
- n^+ = the number of not relevant documents, classified as relevant.

Obviously, the total number of documents N is equal to:

$$N = p^+ + n^+ + p^- + n^- \tag{4.6}$$

Table 4.10: Text Categorization Improvement using Concept-based Combined Model ($weight_{comb}$)

DataSet		***Single-Term***			***Concept-based***			***Improvement***
		Micro Avg(F1)	Macro Avg(F1)	Avg Error	Micro Avg(F1)	Macro Avg(F1)	Avg Error	
Reuters	SVM	0.9023	0.7421	0.0871	0.9512	0.8953	0.0121	+5.41%, +20.64%, -86.10%
	NB	0.8132	0.6127	0.2754	0.9073	0.8462	0.0342	+11.57%, +38.11%, -87.58%
	Rocchio	0.8543	0.6513	0.1632	0.9136	0.8574	0.0231	+6.94%, +31.64%, -85.84%
	kNN	0.8736	0.6865	0.1074	0.9457	0.8722	0.0122	+8.25%, +27.05%, -88.64%
ACM	SVM	0.6851	0.4973	0.1782	0.8975	0.8263	0.0532	+31.00%, +66.15%, -70.14%
	NB	0.6134	0.4135	0.4215	0.8423	0.7964	0.0641	+37.31%, +92.59%, -84.79%
	Rocchio	0.6327	0.4826	0.2733	0.8726	0.7935	0.0635	+37.91%, +64.42%, -76.76%
	kNN	0.6543	0.4952	0.2103	0.8901	0.8071	0.0542	+36.03%, +62.98%, -74.22%
Brown	SVM	0.8537	0.6143	0.1134	0.9352	0.8753	0.0211	+ 9.54%, +42.48%, -81.39%
	NB	0.7864	0.5071	0.3257	0.9067	0.8372	0.0341	+15.29%, +65.09%, -89.53%
	Rocchio	0.8067	0.5728	0.2413	0.9135	0.8465	0.0243	+13.23%, +47.78%, -89.92%
	kNN	0.8291	0.5934	0.1256	0.9274	0.8621	0.0213	+11.85%, +45.28%, -83.04%
Newsgroups	SVM	0.7951	0.5923	0.1325	0.9254	0.8573	0.0221	+16.38%, +44.74%, -83.32%
	NB	0.7023	0.4975	0.3621	0.8921	0.8026	0.0343	+27.02%, +61.32%, -90.52%
	Rocchio	0.7124	0.5346	0.2571	0.8935	0.8059	0.0324	+25.42%, +50.74%, -87.39%
	kNN	0.7341	0.5581	0.1423	0.9148	0.8524	0.0233	+24.61%, +52.73%, -83.62%

For the class of relevant documents:

$$Precision(P) = \frac{p^+}{p^+ + n^+} \tag{4.7}$$

$$Recall(R) = \frac{p^+}{p^+ + p^-} \tag{4.8}$$

The $F - measure_\alpha$ is defined as:

$$F_\alpha = \frac{(1+\alpha) * P * R}{(\alpha * P) + R} \tag{4.9}$$

The error rate is expressed by:

$$Error = \frac{n^+ + p^-}{N} \tag{4.10}$$

Generally, the Macro-averaged measure is determined by first computing the performance measures per category and then averaging these to compute the global mean. The Micro-averaged measure which is determined by first computing the totals p^+, p^-, n^+, and n^- for all categories and then use these totals to compute the performance measures. Micro-averaged gives equal weight to every document, while Macro-averaged gives equal weight to each category.

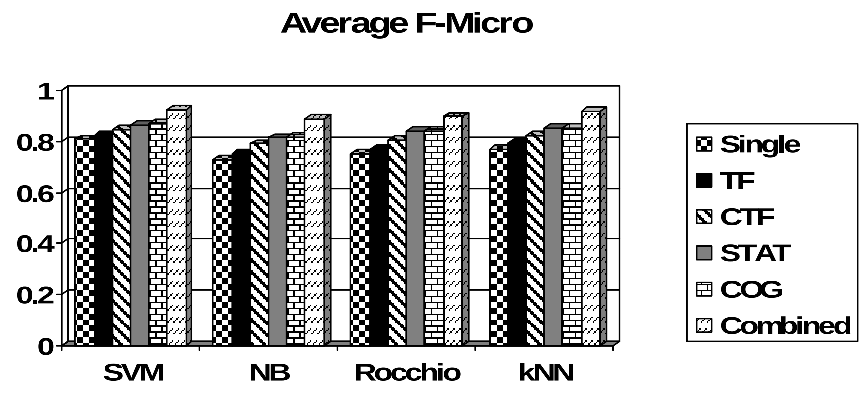

Figure 4.7: Categorization Improvements (F-Micro)

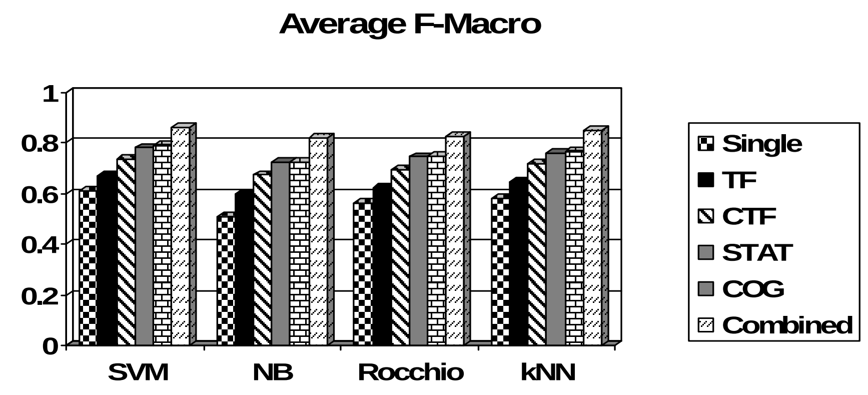

Figure 4.8: Categorization Improvements (F-Macro)

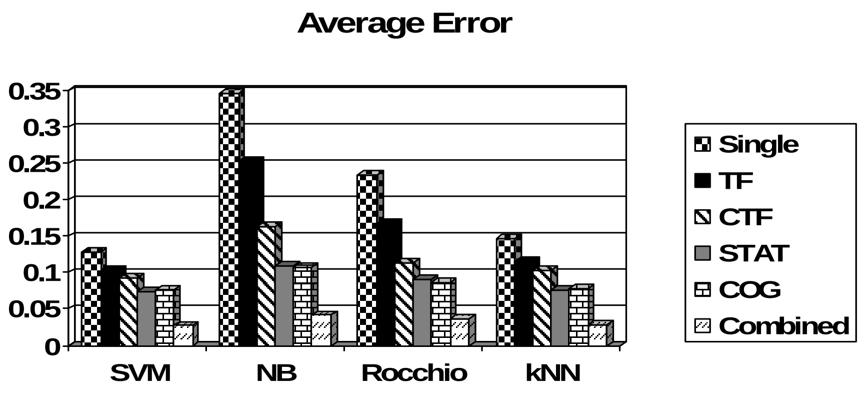

Figure 4.9: Categorization Improvements (Error)

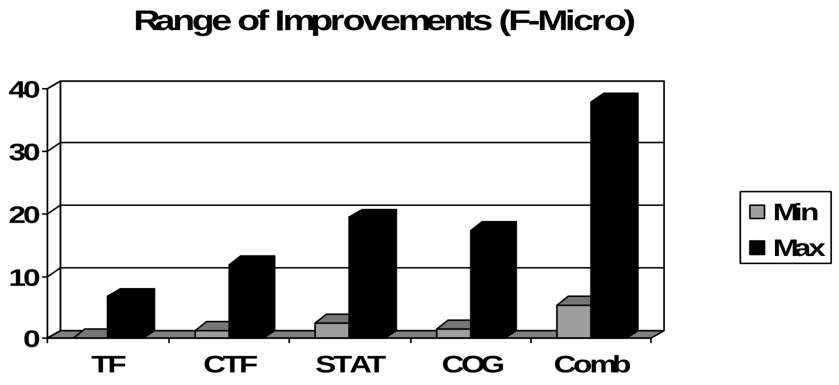

Figure 4.10: Categorization Range of Improvements (F-Micro)

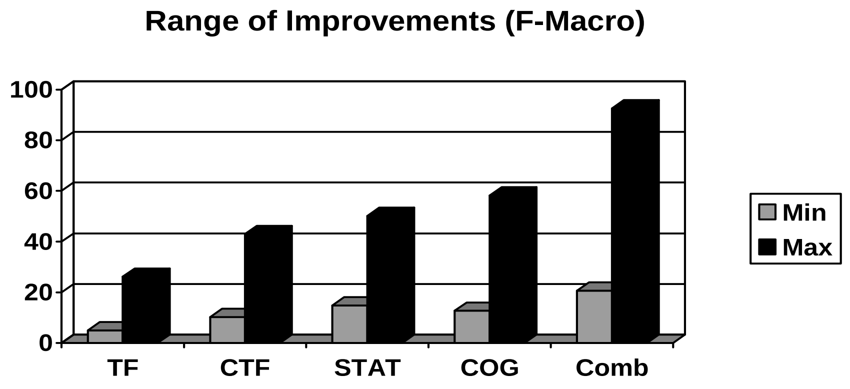

Figure 4.11: Categorization Range of Improvements (F-Macro)

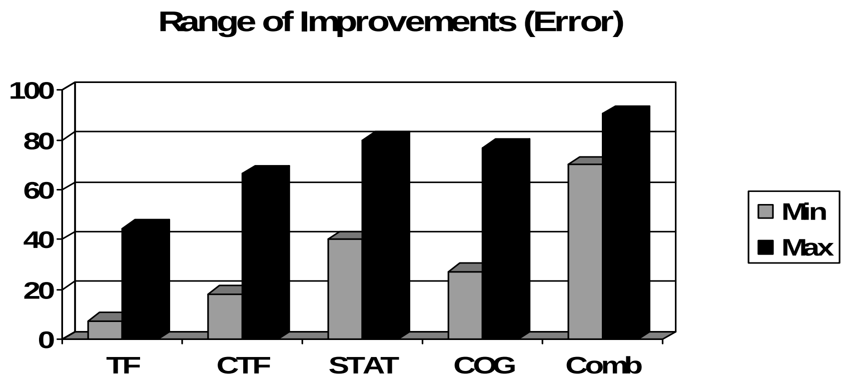

Figure 4.12: Categorization Range of Improvements (Error)

Table 4.11: Text Retrieval Improvement using Concept-based Term Frequency ($weight_{tf}$)

DataSet	***Search Engine***	***Baseline***			***Concept-based***			***Improvement***
		bpref	P(10)	MAP	bpref	P(10)	MAP	
TREC 7(351-400)	Lucene	0.2506	0.4860	0.2421	0.3287	0.5283	0.3265	+31.16%, +8.70%, +34.86%
	Terrier	0.2506	0.4860	0.2421	0.3594	0.5649	0.3561	+43.41%, +16.23%, +47.08%
TREC 8(401- 450)	Lucene	0.2823	0.4960	0.2841	0.3174	0.5437	0.3219	+12.43%, +9.61%, +13.30%
	Terrier	0.2823	0.4960	0.2841	0.3562	0.5612	0.3476	+26.17%, +13.14%, +22.35%
Reuters	Lucene	0.3411	0.5843	0.3361	0.4583	0.7724	0.4357	+34.35%, +32.19%, +29.63%
	Terrier	0.3272	0.5831	0.3152	0.4269	0.7586	0.4182	+30.47%, +30.09%, +32.67%
Cranfield	Lucene	0.3262	0.5471	0.3312	0.4136	0.7415	0.4283	+26.79%, +35.53%, +29.31%
	Terrier	0.3143	0.5253	0.3254	0.4597	0.7264	0.4019	+46.26%, +38.28%, +23.50%
NPL	Lucene	0.2357	0.4835	0.2246	0.3027	0.5927	0.2964	+28.42%, +22.58%, +31.96%
	Terrier	0.2241	0.4712	0.2135	0.2781	0.5618	0.2763	+24.09%, +19.22%, +29.41%
Med	Lucene	0.5062	0.7429	0.5183	0.5734	0.8473	0.5864	+13.27%, +14.05%, +13.13%
	Terrier	0.5134	0.7635	0.5294	0.5927	0.8591	0.5935	+15.44%, +12.52%, +12.10%
CISI	Lucene	0.2132	0.4537	0.2051	0.3086	0.5382	0.3164	+44.74%, +18.62%, +54.26%
	Terrier	0.2246	0.4612	0.2216	0.3259	0.5617	0.3375	+45.10%, +21.79%, +52.30%
LISA	Lucene	0.3472	0.5742	0.3561	0.4283	0.6741	0.4537	+23.35%, +17.39%, +27.40%
	Terrier	0.3435	0.5618	0.3458	0.4175	0.6493	0.4068	+21.54%, +15.57%, +17.64%
CACM	Lucene	0.4276	0.6217	0.4025	0.5361	0.7529	0.5147	+25.37%, +21.10%, +27.87%
	Terrier	0.4352	0.6359	0.4308	0.5492	0.7841	0.5386	+26.19%, +23.30%, +25.02%

Table 4.12: Text Retrieval Improvement using Conceptual Term Frequency ($weight_{ctf}$)

DataSet	***Search Engine***	***Baseline***			***Concept-based***			***Improvement***
		bpref	P(10)	MAP	bpref	P(10)	MAP	
TREC 7(351-400)	Lucene	0.2506	0.4860	0.2421	0.4193	0.6142	0.3742	+67.31%, +26.37%, +54.56%
	Terrier	0.2506	0.4860	0.2421	0.4275	0.6583	0.3958	+70.59%, +35.45%, +63.48%
TREC 8(401- 450)	Lucene	0.2823	0.4960	0.2841	0.4367	0.5972	0.4497	+54.69%, +20.40%, +58.28%
	Terrier	0.2823	0.4960	0.2841	0.4592	0.6148	0.4635	+62.66%, +23.95%, +63.14%
Reuters	Lucene	0.3411	0.5843	0.3361	0.5617	0.8574	0.5432	+64.67%, +46.73%, +61.6%
	Terrier	0.3272	0.5831	0.3152	0.5223	0.8261	0.5137	+59.62%, +41.67%, +62.97%
Cranfield	Lucene	0.3262	0.5471	0.3312	0.5379	0.8156	0.5362	+64.89%, +49.07%, +61.89%
	Terrier	0.3143	0.5253	0.3254	0.5126	0.7943	0.5284	+63.09%, +51.20%, +62.38%
NPL	Lucene	0.2357	0.4835	0.2246	0.3752	0.6742	0.3628	+59.18%, +39.44%, +61.53%
	Terrier	0.2241	0.4712	0.2135	0.3594	0.6358	0.3574	+60.37%, +34.93%, +67.40%
Med	Lucene	0.5062	0.7429	0.5183	0.6528	0.9036	0.6539	+28.96%, +21.63%, +26.16%
	Terrier	0.5134	0.7635	0.5294	0.6749	0.9125	0.6824	+31.45%, +19.51%, +28.90%
CISI	Lucene	0.2132	0.4537	0.2051	0.3952	0.6835	0.3928	+85.36%, +50.65%, +91.51%
	Terrier	0.2246	0.4612	0.2216	0.4258	0.6942	0.4156	+89.58%, +50.52%, +87.54%
LISA	Lucene	0.3472	0.5742	0.3561	0.5176	0.7928	0.5268	+49.07%, +38.07%, +47.93%
	Terrier	0.3435	0.5618	0.3458	0.4925	0.7853	0.5073	+43.37%, +39.78%, +46.70%
CACM	Lucene	0.4276	0.6217	0.4025	0.5736	0.8625	0.5638	+34.14%, +38.73%, +40.07%
	Terrier	0.4352	0.6359	0.4308	0.5924	0.8874	0.5872	+36.12%, +39.55%, +36.30%

4.4 Text Retrieval

The experimental setup consists of eight document collections. The first document collection contains 528,155 documents from the TREC document collection run on disks 4 and 5, not including the congressional record subcollection. The testing was performed using TREC 7(351-400) and TREC 8(401- 450) topics over the TREC collection. The second collection has 12,902 documents from the Reuters 21578 dataset. There are 9,603 documents in the training set and 3,299 documents in the test set. Out of the 5 category sets, the topic category set contains 135 categories, but only 95 categories have at least one document in the training set. These 95 categories were used in the experiments as queries. The third collection is the CRAN set of 1,398 aerodynamics abstracts with 225 queries from the Cranfield collection. The fourth collections is the LISA (Library and Information Science Abstracts) set of 5,872 abstracts with 35 queries. The fifth collection consists of 11,429 document titles with 93 queries from the NPL (also known as the VASWANI) set. The sixth collection is the Med (Medline) set of 1,033 articles with 30 queries from a medical journal. The seventh collection is the CISI set of 1,460 documents with 112 queries. The eighth collection

Table 4.13: Text Retrieval Improvement using Concept-based Statistical Analyzer ($weight_{stat}$)

DataSet	***Search Engine***	***Baseline***			***Concept-based***			***Improvement***
		bpref	P(10)	MAP	bpref	P(10)	MAP	
TREC 7(351-400)	Lucene	0.2506	0.4860	0.2421	0.4537	0.6914	0.4378	+81.04%, +42.26%, +80.83%
	Terrier	0.2506	0.4860	0.2421	0.4621	0.7083	0.4492	+84.39%, +45.74%, +85.54%
TREC 8(401- 450)	Lucene	0.2823	0.4960	0.2841	0.4755	0.6852	0.4761	+68.43%, +38.14%, +67.58%
	Terrier	0.2823	0.4960	0.2841	0.4867	0.6913	0.4852	+72.40%, +39.37%, +70.78%
Reuters	Lucene	0.3411	0.5843	0.3361	0.6034	0.9012	0.5873	+76.89%, +54.23%, +74.73%
	Terrier	0.3272	0.5831	0.3152	0.5862	0.8943	0.5624	+79.15%, +53.36%, +78.42%
Cranfield	Lucene	0.3262	0.5471	0.3312	0.5622	0.8975	0.5623	+72.34%, +64.04%, +69.77%
	Terrier	0.3143	0.5253	0.3254	0.5581	0.8749	0.5467	+77.56%, +66.55%, +68%
NPL	Lucene	0.2357	0.4835	0.2246	0.4371	0.7264	0.4362	+85.44%, +50.23%, +94.21%
	Terrier	0.2241	0.4712	0.2135	0.4289	0.7083	0.4275	+91.38%, +50.31%, >+100%
Med	Lucene	0.5062	0.7429	0.5183	0.7724	0.9521	0.7862	+52.58%, +28.15%, +51.68%
	Terrier	0.5134	0.7635	0.5294	0.7863	0.9574	0.7935	+53.15%, +25.39%, +49.88%
CISI	Lucene	0.2132	0.4537	0.2051	0.4652	0.7543	0.4621	>+100%, +66.25%, >+100%
	Terrier	0.2246	0.4612	0.2216	0.4831	0.7629	0.4857	>+100%, +65.41%, >+100%
LISA	Lucene	0.3472	0.5742	0.3561	0.5623	0.8921	0.5627	+61.95%, +55.36%, +58.01%
	Terrier	0.3435	0.5618	0.3458	0.5462	0.8765	0.5436	+59.01%, +56.01%, +57.2%
CACM	Lucene	0.4276	0.6217	0.4025	0.6473	0.9074	0.6286	+51.37%, +45.95%, +56.17%
	Terrier	0.4352	0.6359	0.4308	0.6582	0.9105	0.6491	+51.24%, +43.18%, +50.67%

is the CACM set of 3,204 titles and abstracts with 64 queries from the CACM journal.

For each dataset, stop words are removed from the concepts that are extracted by the proposed model. The extracted concepts are stemmed using the Porter stemmer algorithm [7]. These concepts are indexed as fields associated to their documents for the text retrieval purpose. These concepts are used to build concept-based index for each document. Two popular information retrieval systems are chosen for testing the effect on the search result between the concept-based index and the traditional index of a document content: Lucene search engine [73] and Terrier retrieval platform [74] as shown in Figure(4.13).

Most of the retrieval evaluation measures are derived in some way from recall and precision. Precision is the proportion of retrieved documents that are relevant, and recall is the proportion of relevant documents that are retrieved [33].

$$P = Precision(i, j) = \frac{N_r}{T} \tag{4.11}$$

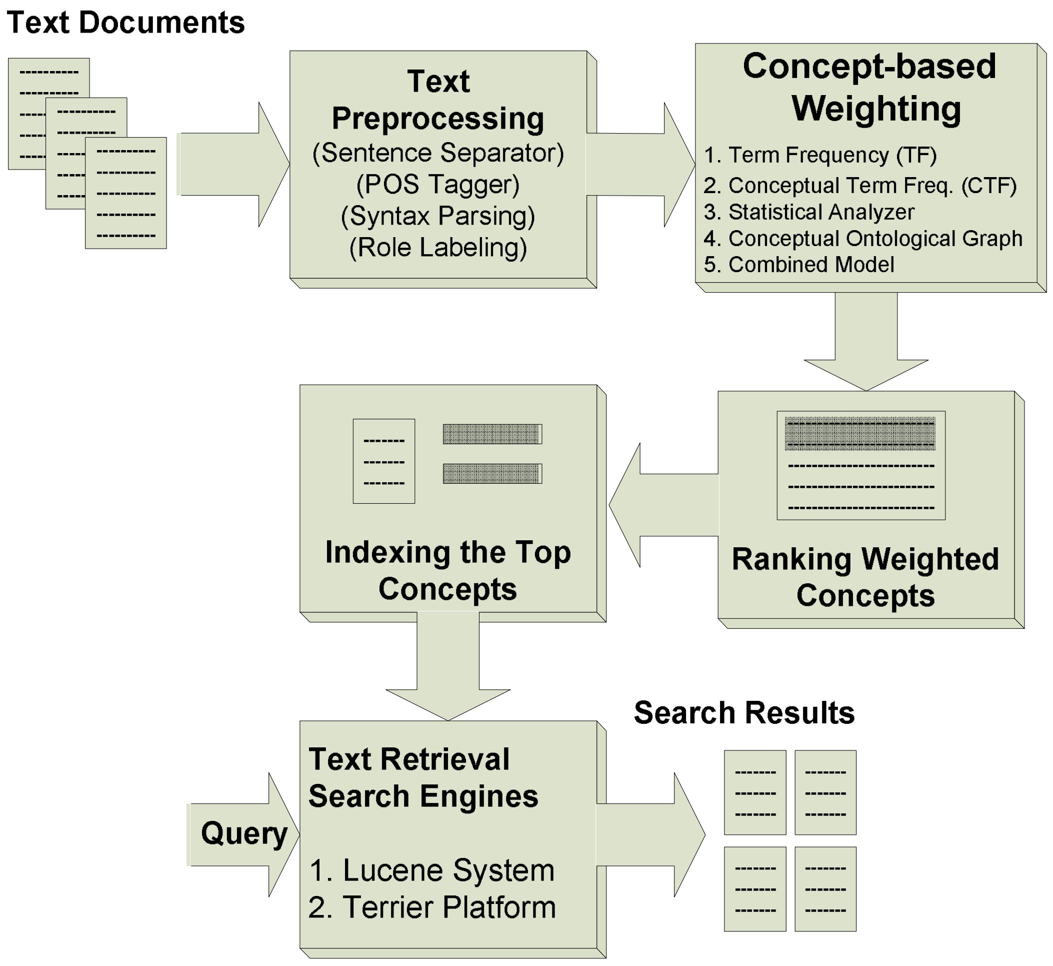

Figure 4.13: Concept-based Model For Text Retrieval

Table 4.14: Text Retrieval Improvement using Conceptual Ontological Graph ($weight_{COG}$)

DataSet	***Search Engine***	***Baseline***			***Concept-based***			***Improvement***
		bpref	P(10)	MAP	bpref	P(10)	MAP	
TREC 7(351-400)	Lucene	0.2506	0.4860	0.2421	0.4412	0.6725	0.4253	+76.05%, +38.37%, +75.67%
	Terrier	0.2506	0.4860	0.2421	0.4533	0.6937	0.4327	+80.88%, +42.73%, +78.72%
TREC 8(401- 450)	Lucene	0.2823	0.4960	0.2841	0.4816	0.7031	0.4852	+70.59%, +41.75%, +70.78%
	Terrier	0.2823	0.4960	0.2841	0.4924	0.7115	0.4973	+74.42%, +43.44%, +75.04%
Reuters	Lucene	0.3411	0.5843	0.3361	0.5837	0.8955	0.5468	+71.12%, +53.26%, +62.68%
	Terrier	0.3272	0.5831	0.3152	0.5514	0.8745	0.5157	+68.52%, +49.97%, +63.61%
Cranfield	Lucene	0.3262	0.5471	0.3312	0.5743	0.9023	0.5761	+76.05%, +64.92%, +73.94%
	Terrier	0.3143	0.5253	0.3254	0.5651	0.8875	0.5673	+79.8%, +68.95%, +74.33%
NPL	Lucene	0.2357	0.4835	0.2246	0.4582	0.7358	0.4517	+94.39%, +52.18%, >+100%
	Terrier	0.2241	0.4712	0.2135	0.4391	0.7264	0.4236	+95.93%, +54.15%, +98.40%
Med	Lucene	0.5062	0.7429	0.5183	0.7535	0.9463	0.7619	+48.85%, +27.37%, +46.99%
	Terrier	0.5134	0.7635	0.5294	0.7613	0.9518	0.7842	+48.28%, +24.66%, +48.12%
CISI	Lucene	0.2132	0.4537	0.2051	0.4316	0.7152	0.4305	>+100%, +57.63%, >+100%
	Terrier	0.2246	0.4612	0.2216	0.4527	0.7236	0.4417	>+100%, +56.89%, +99.32%
LISA	Lucene	0.3472	0.5742	0.3561	0.5837	0.9105	0.5886	+68.11%, +58.56%, +65.29%
	Terrier	0.3435	0.5618	0.3458	0.5741	0.9032	0.5793	+67.13%, +60.76%, +67.52%
CACM	Lucene	0.4276	0.6217	0.4025	0.6538	0.9228	0.6417	+52.89%, +48.43%, +59.42%
	Terrier	0.4352	0.6359	0.4308	0.6702	0.9345	0.6539	+53.99%, +46.95%, +51.78%

and

$$R = Recall(i, j) = \frac{N_r}{T_r} \tag{4.12}$$

where:

- N_r is the number of relevant records retrieved,
- T is the total number of irrelevant and relevant records retrieved, and
- T_r is the total number of relevant records in the database.

In order to evaluate the quality of the text retrieval, three widely used evaluation measures in information retrieval literature are adopted. These measures are the precision at 10 documents retrieved P(10) [34], the preference measure, bpref, which is a function of the number of times judged non-relevant documents are retrieved before relevant documents [34], and the mean uninterpolated average precision (MAP) [34].

Table 4.15: Text Retrieval Improvement using Concept-based Combined Model ($weight_{comb}$)

DataSet	***Search Engine***	***Baseline***			***Concept-based***			***Improvement***
		bpref	P(10)	MAP	bpref	P(10)	MAP	
TREC 7(351-400)	Lucene	0.2506	0.4860	0.2421	0.5312	0.7613	0.5263	>+100%, +56.64%, >+100%
	Terrier	0.2506	0.4860	0.2421	0.5461	0.7821	0.5312	>+100%, +60.92%, >+100%
TREC 8(401- 450)	Lucene	0.2823	0.4960	0.2841	0.5713	0.7932	0.5742	>+100%, +59.91%, >+100%
	Terrier	0.2823	0.4960	0.2841	0.5822	0.8041	0.5872	>+100%, +62.11%, >+100%
Reuters	Lucene	0.3411	0.5843	0.3361	0.6522	0.9115	0.6452	+91.20%, +55.99%, +91.96%
	Terrier	0.3272	0.5831	0.3152	0.6351	0.9042	0.6273	+94.10%, +55.06%, +99.01%
Cranfield	Lucene	0.3262	0.5471	0.3312	0.6831	0.9122	0.6851	>+100%, +66.73%, >+100%
	Terrier	0.3143	0.5253	0.3254	0.6752	0.8924	0.6792	>+100%, +69.88%, >+100%
NPL	Lucene	0.2357	0.4835	0.2246	0.5813	0.8723	0.5738	>+100%, +80.41%, >+100%
	Terrier	0.2241	0.4712	0.2135	0.5746	0.8514	0.5427	>+100%, +80.68%, >+100%
Med	Lucene	0.5062	0.7429	0.5183	0.8347	0.9572	0.8526	+64.89%, +28.84%, +64.49%
	Terrier	0.5134	0.7635	0.5294	0.8525	0.9643	0.8793	+66.04%, +26.29%, +66.09%
CISI	Lucene	0.2132	0.4537	0.2051	0.5643	0.8481	0.5627	>+100%, +86.92%, >+100%
	Terrier	0.2246	0.4612	0.2216	0.5827	0.8659	0.5814	>+100%, +87.74%, >+100%
LISA	Lucene	0.3472	0.5742	0.3561	0.6725	0.9423	0.6732	+93.69%, +64.1%, +89.04%
	Terrier	0.3435	0.5618	0.3458	0.6543	0.9215	0.6585	+90.48%, +64.02%, +90.42%
CACM	Lucene	0.4276	0.6217	0.4025	0.7512	0.9426	0.7351	+75.678%, +51.61%, +82.63%
	Terrier	0.4352	0.6359	0.4308	0.7635	0.9537	0.7624	+75.43%, +49.97%, +76.97%

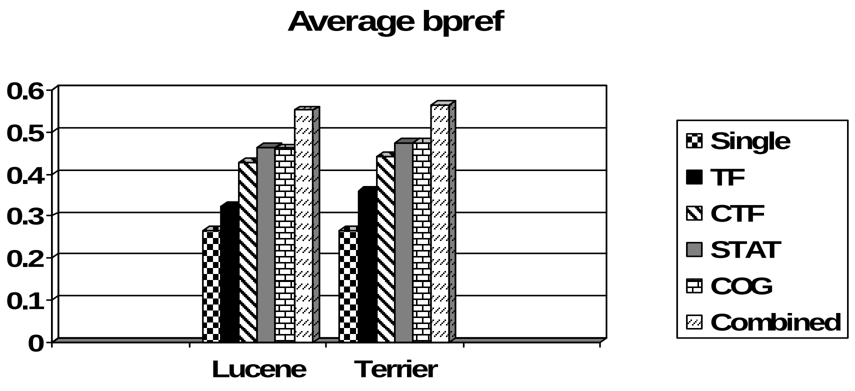

Figure 4.14: Retrieval Improvements (bpref)

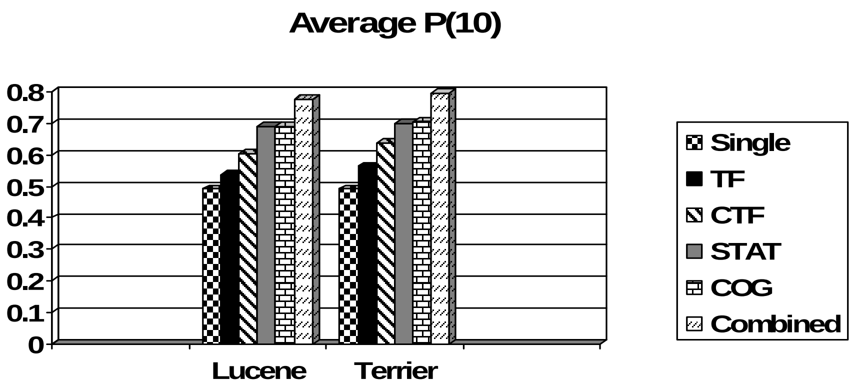

Figure 4.15: Retrieval Improvements (P10)

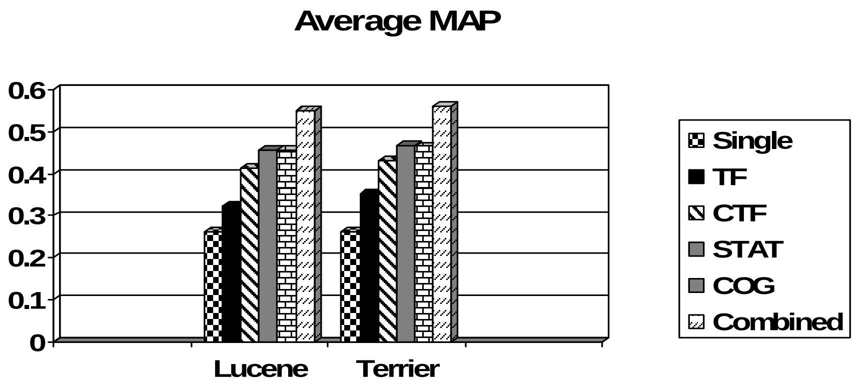

Figure 4.16: Retrieval Improvements (MAP)

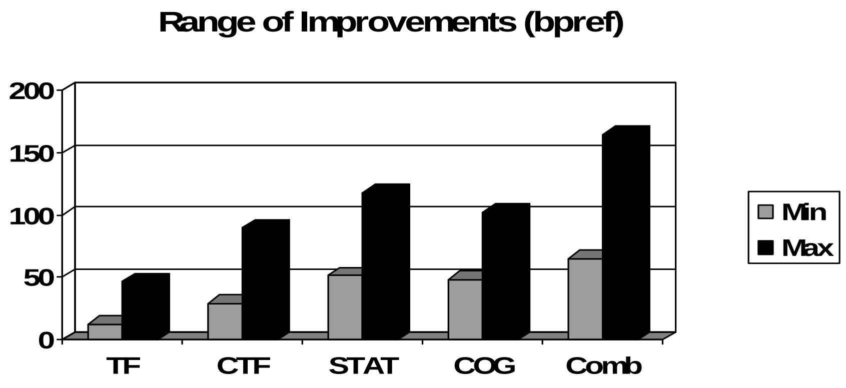

Figure 4.17: Retrieval Range of Improvements (bpref)

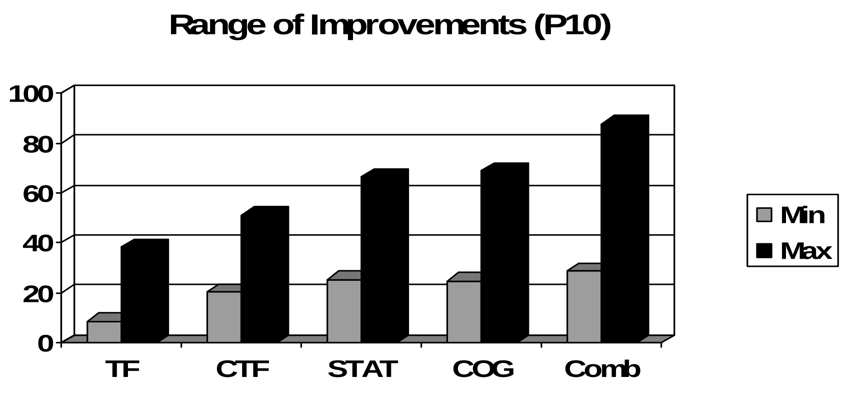

Figure 4.18: Retrieval Range of Improvements (P10)

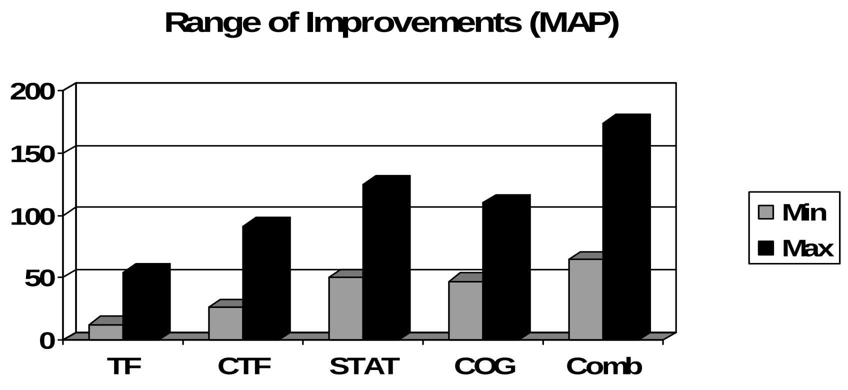

Figure 4.19: Retrieval Range of Improvements (MAP)

4.5 Discussion

In this section, we discuss the improvements of the evaluation measures achieved by the concept-based model compared to the traditional techniques in text clustering, categorization and retrieval.

4.5.1 Text Clustering Results

In text clustering experiments, the aim is to maximize the F-measure, and minimize the Entropy of clusters to achieve high quality clustering. As shown in Table(4.1) for the concept-based term frequency $weight_{tf}$ weighting, Table(4.2) for the conceptual term frequency $weight_{ctf}$ weighting, Table(4.3) for the concept-based statistical analyzer $weight_{stat}$, Table(4.4) for the COG representation $weight_{cog}$ weighting, and Table(4.5) for the combined model weighting, the ward and the complete linkages were used as the cluster distance measures for the HAC method since they tend to produce tight clusters with small diameter. A document-to-cluster similarity threshold of 0.3 was used in the single pass clustering method. A k of 5 and a cluster similarity threshold of 0.35 were used in the k-NN method. The parameters chosen for the different algorithms were the ones that produced best results.

For the concept-based term frequency ($weight_{tf}$), the percentage of improvement ranges from +6.31% to +67.94% increase in the F-measure quality, and -29.56% to -64.67% drop in Entropy (lower is better for Entropy).

For the conceptual term frequency ($weight_{ctf}$), the percentage of improvement ranges from +23.37% to (above 100%) increase in the F-measure quality, and -62.65% to -82.96% drop in Entropy.

For the concept-based statistical analyzer ($weight_{stat}$), the percentage of improvement ranges from +27.93% to (above 100%) increase in the F-measure quality, and -62.7% to -95.68% drop in Entropy.

For the conceptual ontological graph ($weight_{COG}$), the percentage of improvement ranges from +26.41% to (above 100%) increase in the F-measure quality, and -77.10% to -96.62% drop in Entropy.

For the combined model $weight_{comb}$ between the concept-based statistical analyzer and the COG representation , the percentage of improvement ranges from +29.73% to (above 100%) increase in the F-measure quality, and -89.27% to -98.33% drop in Entropy.

It is shown that the HAC clustering with the ward linkage has the best performance. Moreover, the HAC clustering with the complete linkage performance is close to the k-NN clustering performance. It is known that Single Pass clustering is very sensitive to noise; that is why it has the worst performance. However, when the concept-based similarity was introduced using the combined weighting scheme among the concept-based statistical analyzer and the COG representation the quality of clusters produced was pushed close to that produced by HAC and k-NN.

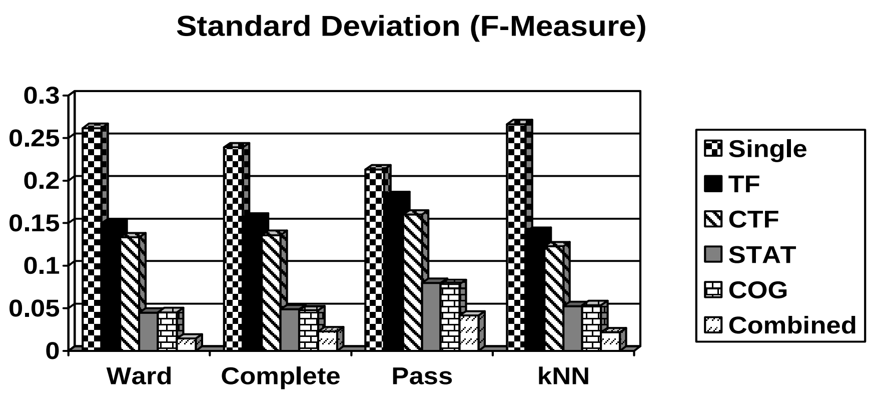

Figure 4.20: Standard Deviation (F-Measure)

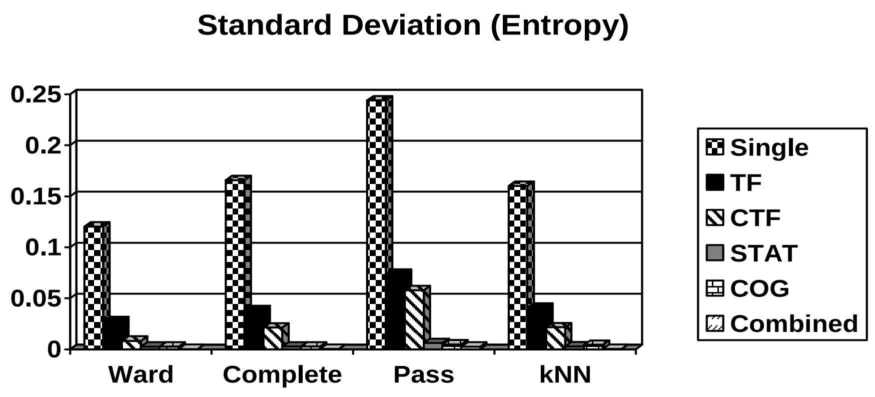

Figure 4.21: Standard Deviation (Entropy)

Figures (4.2,4.3) show the average of the F-Measure and Entropy over the entire datasets. Figures (4.2,4.3) illustrate the improvement of text clustering using (1) concept-based term frequency $weight_{tf}$, (2) conceptual term frequency $weight_{ctf}$, (3) concept-based statistical analyzer $weight_{stat}$, (4) COG representation $weight_{cog}$, and (5) combined model of the concept-based statistical analyzer and the COG representation. Figures (4.4,4.5) show the range of improvement of F-Measure and Entropy over the entire clustering techniques. To show the significance of the concept-based model in text clustering, Figures (4.20,4.21) show that the standard deviation is improved by using the concept-based model. Further details about the results of each clustering technique for each dataset can be found in the appendix.

It is illustrated that $weight_{tf}$ outperforms the single-weighting. In addition, the $weight_{ctf}$ weighting scheme is more accurate than the $weight_{tf}$ weighting when it comes to calculate the concept-based relations between documents. The combined model $weight_{combined}$ between the

$weight_{stat}$ and $weight_{cog}$ weighting schemes can accurately measure the importance of a concept at the sentence and document levels which leads to enhance the clustering quality substantially.

4.5.2 Text Categorization Results

In text categorization experiments, the intention is to maximize *Macro-averaged F1* and *Micro-averaged F1* and minimize the *error rate* measures to achieve high quality in text categorization.

The popular SVMlight implementation [75] is used with parameter C = 1000 (tradeoff between training error and margin). A k of 25 is used in the kNN method as illustrated in Tables(4.6,4.7, 4.8,4.9 and 4.10). The parameters chosen for the different algorithms were the ones that produced best results. It is illustrated that the entire concept-based weighting outperforms the single-weighting. Moreover, the $weight_{comb}$ has the best performance over the other concept-based weighings.

The results listed in Tables(4.6, 4.7, 4.8, 4.9, and 4.10) show the improvement on the categorization quality obtained by the (1) concept-based term frequency $weight_{tf}$, (2) conceptual term frequency $weight_{ctf}$, (3) concept-based statistical analyzer $weight_{stat}$, (4) COG representation $weight_{cog}$, and (5) combined model of the concept-based statistical analyzer and the COG representation.

For the concept-based term frequency ($weight_{tf}$), the percentage of improvement ranges from +0.12% to +6.77% increase (higher is better) in the Micro-averaged F1 quality and from +4.98% to +26.16% increase (higher is better) in the Macro-averaged F1 quality, and -7.29% to -44.73% drop (lower is better) in the error rate as shown in Table(4.6).

For the conceptual term frequency ($weight_{ctf}$), the percentage of improvement ranges from +1.32% to +11.84% increase (higher is better) in the Micro-averaged F1 quality and from +10.17% to +43.03% increase (higher is better) in the Macro-averaged F1 quality, and -18.1% to -66.4% drop (lower is better) in the error rate as shown in Table(4.7).

For the concept-based statistical analyzer ($weight_{stat}$), the percentage of improvement ranges from +2.56% to +19.35% increase (higher is better) in the Micro-averaged F1 quality and from +14.82% to +50.12% increase (higher is better) in the Macro-averaged F1 quality, and -39.95% to -80.08% drop (lower is better) in the error rate as shown in Table(4.8).

For the conceptual ontological graph ($weight_{COG}$), the percentage of improvement ranges from +12.68% to +58.21% increase in the Macro-averaged F1 quality and from +1.66% to 17.37+%

increase in the Micro-averaged F1 quality, and -27.21% to -77.09% drop (lower is better) in the error rate as shown in Table(4.9).

For the combined model $weight_{comb}$ between the concept-based statistical analyzer and the COG representation, the percentage of improvement ranges from +20.64% to +92.59% increase in the Macro-averaged F1 quality and from +5.41% to +37.91% increase in the Micro-averaged F1 quality, and -70.14% to -90.52% drop (lower is better) in the error rate as shown in Table(4.10).

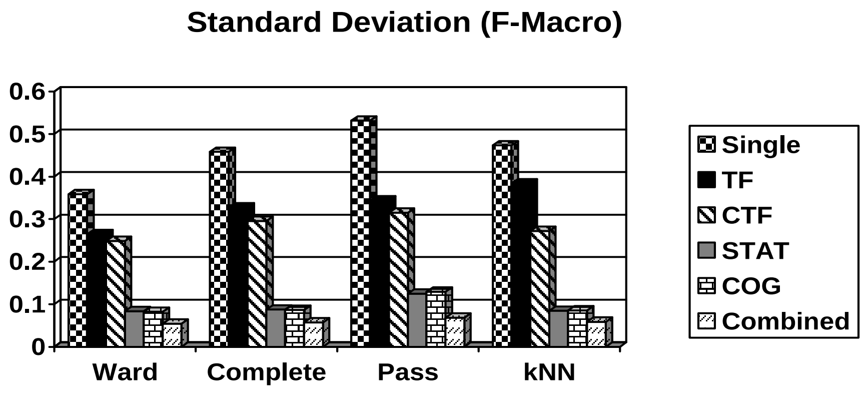

Figure 4.22: Standard Deviation (F-Macro)

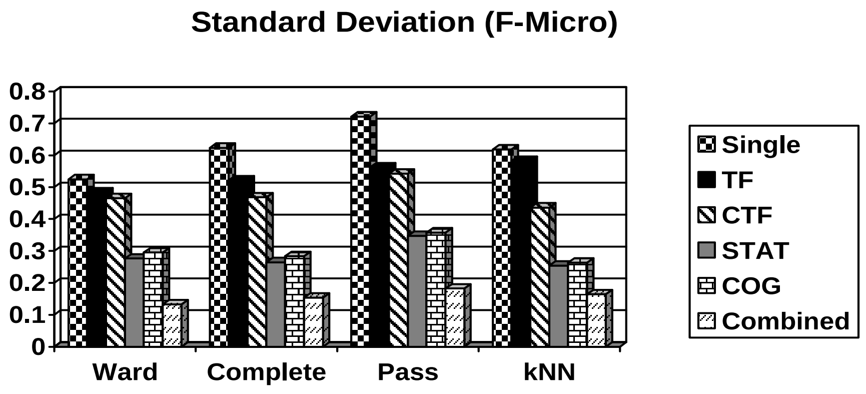

Figure 4.23: Standard Deviation (F-Micro)

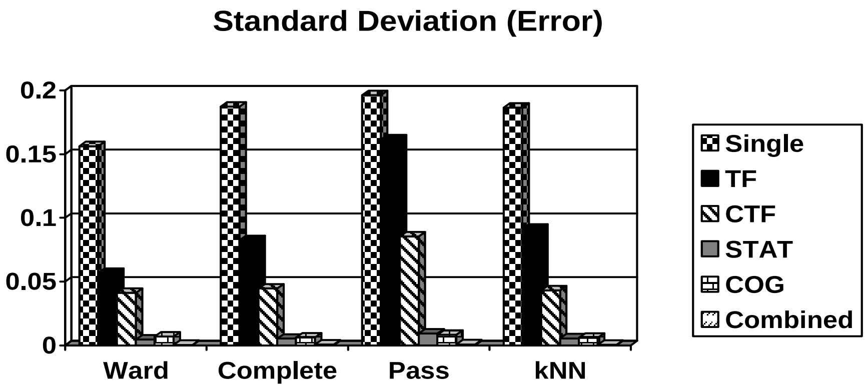

Figure 4.24: Standard Deviation (Error)

Figures (4.8,4.7,4.9) show the average of the F-Macro, F-Micro and error over the entire datasets. Figures (4.8,4.7,4.9) illustrate the improvement of text categorization using (1) concept-based term frequency $weight_{tf}$, (2) conceptual term frequency $weight_{ctf}$, (3) concept-based statistical analyzer $weight_{stat}$, (4) COG representation $weight_{cog}$, and (5) combined model of the concept-based statistical analyzer and the COG representation. Figures(4.11,4.10,4.12) show the range of improvement of F-Macro, F-Micro, and Error over the entire categorization techniques. To show the significance of the concept-based model in text categorization, Figures (4.22,4.23,4.24) show that the standard deviation is improved by using the concept-based model. Further details about the results of each categorization technique for each dataset can be found in the appendix.

4.5.3 Text Retrieval Results

In text retrieval experiments, the intention is to maximize the *bpref, P(10),* and *MAP* measures to achieve high quality text retrieval.

For the concept-based term frequency ($weight_{tf}$), the percentage of improvement ranges from +12.43% to +46.26% increase in the *bpref*, from +8.7% to +38.28% increase in the *P(10)*, and from +12.1% to +54.26% increase in the *MAP*, as shown in Table(4.11).

For the conceptual term frequency ($weight_{ctf}$), the percentage of improvement ranges from +28.96% to +89.58% increase in the *bpref*, from +20.4% to +51.2% increase in the *P(10)*, and from +26.16% to +91.51% increase in the *MAP*, as shown in Table(4.12).

For the concept-based statistical analyzer ($weight_{stat}$), the percentage of improvement ranges from +51.24% to (above 100%) increase in the *bpref*, from +25.39% to +66.55% increase in the *P(10)*, and from +49.88% to (above 100%) increase in the *MAP*, as shown in Table(4.13).

For the conceptual ontological graph ($weight_{COG}$), the percentage of improvement ranges from +48.28% to (above 100%) increase in the *bpref*, from +24.66% to +68.95% increase in the *P(10)*, and from +46.99% to (above 100%) increase in the MAP, as shown in Table(4.14).

For the combined model $weight_{comb}$ between the concept-based statistical analyzer and the COG representation, the percentage of improvement ranges from +64.89% to (above 100%) increase in the *bpref*, from +28.84% to +87.74% increase in the *P(10)*, and from +64.49% to (above 100%) increase in the MAP, as shown in Table(4.5).

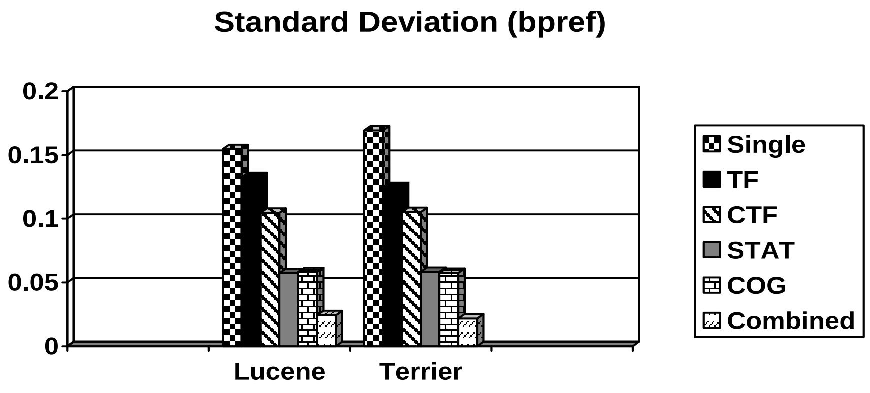

Figure 4.25: Standard Deviation (bpref)

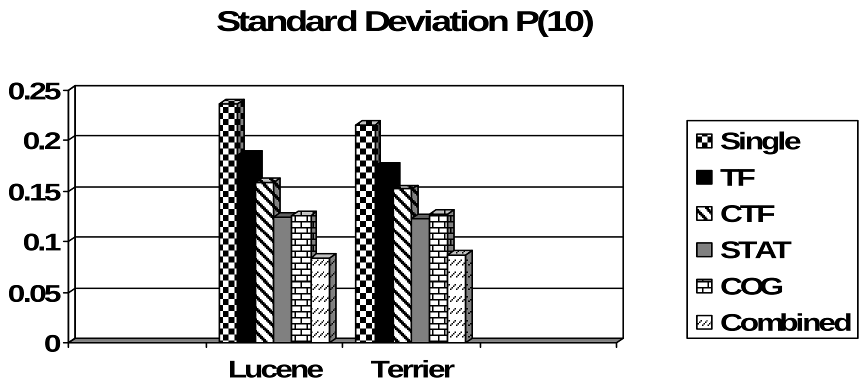

Figure 4.26: Standard Deviation (P10)

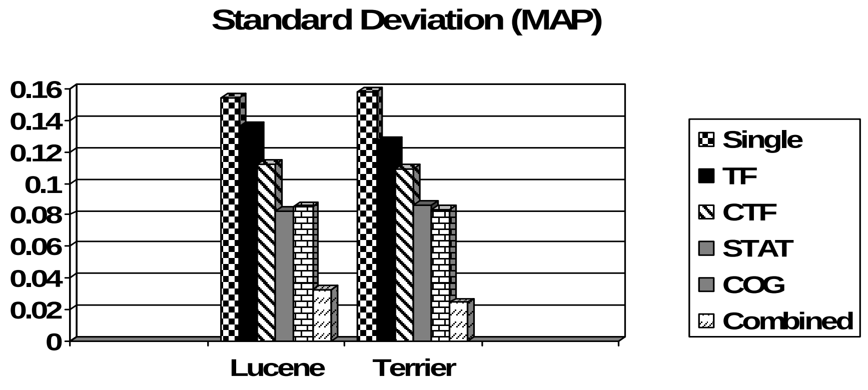

Figure 4.27: Standard Deviation (MAP)

Figures (4.14,4.15,4.16) show the average of the bpref, P10 and MAP over the entire datasets. Figures (4.14,4.15,4.16) show that both Lucene and Terrier retrieval systems provides higher relevance score for the concept-based index than the score of the regular document index by Lucene and Terrier search engines using (1) concept-based term frequency $weight_{tf}$, (2) conceptual term frequency $weight_{ctf}$, (3) concept-based statistical analyzer $weight_{stat}$, (4) COG representation $weight_{cog}$, and (5) combined model of the concept-based statistical analyzer and the COG representation.

Figures((4.17,4.18,4.19)) show the range of improvements of the concept-based model over the text retrieval engines. To show the significance of the concept-based model in text retrieval, Figures (4.25,4.26,4.27) show that the standard deviation is improved by using the concept-based model. Further details about the results of each retrieval technique for each dataset can be found in the appendix.

4.5.4 Observations and Conclusions

As shown in Figures(from A.1 to A.26) in tect clustering, categorization and retrieval, there is a common pattern that the $weight_{tf}$ outperforms that single-term weighting. The $weight_{ctf}$ outperforms $weight_{tf}$. The $weight_{stat}$ and $weight_{cog}$ outperforms $weight_{ctf}$. The $weight_{stat}$ and $weight_{cog}$ outperforms each other interchangeably. The $weight_{comb}$ outperforms the $weight_{stat}$ and $weight_{cog}$.

The $weight_{tf}$ outperforms single-term weighting because the $weight_{tf}$ captures the important terms in a document instead of using the entire terms. However, the $weight_{tf}$ lacks the importance of a term on the sentence level. Thus, the $weight_{ctf}$ is more accurate than the $weight_tf$. The $weight_{stat}$ and $weight_{cog}$ outperforms $weight_{ctf}$ and $weight_{tf}$ due to the fact that both $weight_{stat}$

and $weight_{cog}$ combines between the term frequency and conceptual term frequency that reflects the term analysis on the sentence and document levels.

In text clustering and categorization as shown in Figures(from A.1 to A.20), the weighting of the concept-based statistical analyzer outperforms the weighting of the COG representation in the cases of the Reuters and Newsgroups datasets. On the other hand, the COG representation weighting outperforms the concept-based statistical analyzer in the cases of the ACM and the Brown datasets. This is due to the fact that sentences in Reuters and Newsgroups are shorter (with respect to number of words) than sentences in the ACM and the Brown datasets. Statistical analyzer performs better than the COG representation in short sentences as their writers tend to associate more than one verb to the same term. This association increases the frequency of the semantic roles to the same term captured by the statistical analyzer. On the other hand, the conceptual ontological graph outperforms the statistical analyzer in long sentences as their writers tend to mention the most important term in the sentence once to emphasize on the term. In this case, The COG representation captures the important term based on the incoming and the outgoing links to its node in the COG representation. The weighting ($weight_{combined}$) of both the concept-based statistical analyzer and the COG representation outperforms the weighting of each one individually as it combines between both weighings.

4.6 Summary

Extensive comparisons between $weight_{tf}$, $weight_{ctf}$, $weight_{stat}$, $weight_{cog}$, and $weight_{combind}$ in text clustering, categorization, and retrieval are presented in this chapter. It is observed that using concepts in the text clustering, categorization, and retrieval rather than regular terms, improves the quality substantially. The reason behind this observation is that concepts are less sensitive to noise when it comes to calculating similarity between (1) document and document as in text clustering, (2) document and category as in text categorization, and (3) query and relevant document as in text retrieval. This is due to the fact that these concepts are originally extracted by the semantic role labeler and analyzed using two different techniques with respect to the sentence, document and corpus levels. Thus, the matching among these concepts is less likely to be found when it come to calculating concept similarity. The results produced by the proposed concept-based model in text clustering, categorization, and retrieval have higher quality than those produced by traditional techniques.

Chapter 5

Conclusions and Future Research

5.1 Conclusions

Most of recent text mining systems consider only the presence or the absence of keywords in text. Statistical analysis of word frequencies is not sufficient for representing the meaning of text. Concept-based mining is the area targeting the semantics of text rather than word frequencies.

The main contribution of this work lies in the proposed concept-based model which captures and represents the semantics in text based on concepts. The concept-based model discovers structured knowledge to be utilized in several applications. The new concept-based model is proposed to improve the text clustering, categorization, and retrieval qualities. By exploiting the semantic structure of the sentences in documents, a better text clustering, categorization, and retrieval results are achieved.

The new concept-based model composed of four components. The first component is the concept-based statistical analyzer which analyzes the semantic structure of each sentence to capture the sentence concepts using the conceptual term frequency ctf measure. The second component is the conceptual ontological graph (COG). This representation captures the structure of the sentence semantics represented in the COG hierarchical levels. Such a representation allows choosing concepts based on their weights which represent the contribution of each concept to the meaning of the sentence. This leads to perform concept matching and weighting calculations in each document in a very robust and accurate way. The third component is the concept extractor which combines the weights of concepts extracted by the concept-based statistical analyzer and the conceptual ontological graph into one top concept list. The extracted top concepts are used to build (1) document to document similarity matrix for the purpose of text clustering (2) standard normalized feature vectors using the standard vector space model (VSM) for the purpose of text categorization and (3) concept-based index for the purpose of text retrieval. The fourth component is the concept based similarity measure which is capable of perform an accurate calculation of pair-wise documents. This allows performing concept matching and concept-based similarity calculations among documents in a very robust and accurate way in text clustering. The quality of the text clustering, categorization, and retrieval results achieved by the proposed model surpasses that of traditional weighting approaches significantly.

5.2 Limitation of the Approach

There are number of limitations of the concept-based model. One limitation is that an English sentence has to be grammatically correct. If an English sentence is not grammatically correct, then the semantic role labeler may not have an output. Another limitation is that if an English sentence is extremely short. For example, if a sentence contains only three words. i.e. "Mike plays football". then the subject, verb, and the object will have the same weighting. Though, this is never happen in regular documents as sentences tend to be much longer.

5.3 Future Work

There are a number of suggestions to extend this work. One direction is to link the presented work to web document clustering, categorization, and retrieval. The intention is to apply the same approach on web documents. In this case, a text extraction process is required before applying the proposed approach to web documents. This direction can open new horizons in many business applications. For example, clustering stock markets of the users web blogs based on the sentence meaning can lead to know which stock to buy.

Another future direction is to investigate the usage of WordNet to extract the synonyms, hypernyms, and hyponyms and their effect on document clustering, categorization, and retrieval results, compared to that of traditional methods. In this case, the model analyzes terms and their corresponding synonyms and/or hypernyms on the sentence and document levels. Thus, if two documents contain different words and these words are semantically related, the model can measure the semantic-based similarity between the two documents.

Another area for possible future direction is to perform inference in first order logic on the concepts extracted by the COG representation. At the inference level, the approach takes full advantage of the structure of the proposed representation and makes inferences based on predicate logic. Inference can acquire the basis to understand the meaning of text and deduce new knowledge from already specified knowledge.

5.4 List of Publications

The work in this thesis has resulted in a number of publications, as well as patents, posters and software demos, which are listed below.

Book Chapters

1. Shady Shehata, Fakhri Karray, Mohamed Kamel, "Concept-based Mining Model", Dynamic and Advanced Data Mining for Progressing Technological Development: Innovations and Systemic Approaches, by IGI Publishing (formerly called "Idea Group Publishing"), IRM Press, Information Science Publishing, CyberTech Publishing, and Information Science Reference. 2007.

2. Shady Shehata, Fakhri Karray, Mohamed Kamel, "Concept-based Text Clustering", Evolving Application Domains of Data Warehousing and Mining: Trends and Solutions, by IGI Publishing (formerly called "Idea Group Publishing"), IRM Press, Information Science Publishing, CyberTech Publishing, and Information Science Reference. 2007.

Journal Articles - Submitted

3. Shady Shehata, Fakhri Karray, Mohamed Kamel, "An Efficient Concept-based Mining Model for Enhancing Text Clustering", IEEE Transactions on Knowledge and Data Engineering (TKDE).

4. Shady Shehata, Fakhri Karray, Mohamed Kamel, "An Efficient Model For Enhancing Text Classification Using Sentence Semantics", Special Issue of Computational Intelligence Journal. **(ADMA08 paper has been selected)**

5. Shady Shehata, Fakhri Karray, Mohamed Kamel, "An Efficient Concept-based Method for Semantic Text Analysis", Computational Linguistics Journal.

6. Shady Shehata, Fakhri Karray, Mohamed Kamel, ”An Efficient Concept-based Retrieval Model For Enhancing Search Engine Quality”, Knowledge and Information Systems Journal (KAIS).

Journal Articles - In Preparation

7. Shady Shehata, Fakhri Karray, Mohamed Kamel, ”Enhancing Text Categorization Using Concept-based Model”, ACM Transactions on Knowledge Discovery From Data (TKDD).

Conference Proceedings

8. Shady Shehata, Fakhri Karray, Mohamed Kamel, ”A Concept-based Model for Enhancing Text Categorization”, 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), USA 2007, pp. 629-637. [**Full paper in research track with acceptance rate: less than 8%**]

9. Shady Shehata, Fakhri Karray, Mohamed Kamel, ”Enhancing Text Clustering Using Concept-based Mining Model”, IEEE International Conference on Data Mining (ICDM), Hong Kong 2006, pp. 1043-1048. [**Full paper in research track with acceptance rate: less than 10%**]

10. Shady Shehata, Fakhri Karray, Mohamed Kamel, ”Enhancing Search Engine Quality Using Concept-based Text Retrieval”, IEEE/WIC/ACM International Conference on Web Intelligence (WI), USA 2007, pp. 26-32. [**Acceptance rate: 17%**]

11. Shady Shehata, Fakhri Karray, Mohamed Kamel, ”Enhancing Text Classification Using Sentence Semantics”, Advanced Data Mining and Applications (ADMA), 2008, pp. 87-98. [**Acceptance rate: 26%**] **Best Paper Award Nomination**

12. Shady Shehata, Fakhri Karray, Mohamed Kamel, ”Enhancing Text Retrieval Performance Using Conceptual Ontological Graph”, Workshop on Ontology Mining and Knowledge Discovery from Semi-Structured Documents (MSD), IEEE International Conference on Data Mining, ICDM, Hong Kong 2006, pp. 39-44. [**Acceptance rate: 30%**]

13. Shady Shehata, Fakhri Karray, Mohamed Kamel, ”A Concept-based Graph Representation for Enhancing Text Categorization”, Text Mining Workshop (TMW09), SIAM International Conference on Data Mining (SDM), 2009.

14. Shady H. Shehata, Fakhri Karray, Mohamed Kamel, "Concept-based Mining Model for Learning Objects", LORNET Scientific Conference on Learning Systems of the Future: Integrating Knowledge and Services (I2LOR06), Montreal, Canada 2006.

15. Christopher Brooks, Scott Bateman, Wengang Liu, Gordon McCalla, Jim Greer, Dragan Gaevic, Timmy Eap, Griff Richards, Khaled Hammouda, Shady H. Shehata, Mohamed Kamel, Fakhri Karray, Jelena Jovanovic, "Issues and Directions with Educational Metadata", LORNET Scientific Conference on Learning Systems of the Future: Integrating Knowledge and Services (I2LOR06), Montreal, Canada 2006.

16. Shady H. Shehata, Fakhri Karray, Mohamed Kamel, "Concept Mining using Conceptual Ontological Graph (COG)", LORNET Scientific Conference on Portals and Services for Knowledge Management and Learning on the Semantic Web (I2LOR05), Montreal, Canada 2005.

17. Shady H. Shehata, Fakhri Karray, Mohamed Kamel, Anas Vaqar and Hazem Shehata. "A Framework for Ontology Construction from Text Documents", International Conference of e-Learning Applications, 2005.

18. Shady H. Shehata, Jan Bakus, Fakhri Karray and Mohamed Kamel, "The Effect of Verb Argument Structure on Document Classification", International Conference in Machine Intelligence (ACIDCA-ICMI), 2005.

19. Yu Sun, Fakhri Karray, Shady H. Shehata, Otman Basir, Mohamed Kamel and Jiping Sun, "Measures of Fuzzy Event for Determination of Semantic Meaning", International Conference in Machine Intelligence (ACIDCA-ICMI), 2005.

20. Shady H. Shehata, Fakhri Karray, Mohamed Kamel, "Multi-Agent Framework for Enhancing Ontology based on Fuzzy Inferencing", LORNET Scientific Conference on Towards the Educational Semantic Web (I2LOR - 04), Montreal, Canada, November 18th and 19th, 2004.

Posters

- Shady Shehata, Fakhri Karray, Mohamed Kamel, "Enhancing Text Categorization, Retrieval and Clustering Using Concept-based Model", 4th annual LORNET Scientific Conference I2LOR-07 in Montreal (Nov 4-7), 2007. **First Position Award For Best Poster**

- Shady Shehata, Fakhri Karray, Mohamed Kamel, "A Concept-based Model for Enhancing Text Categorization", 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), USA 2007.
- Shady Shehata, Fakhri Karray, Mohamed Kamel, "Concept-based Mining Model: Extracting Concepts from Documents", Theme of Physical Science, Math & Technology at the Graduate Student Research Conference April, 2007. **Best Poster Presentation Award**
- Shady Shehata, Fakhri Karray, Mohamed Kamel, "Enhancing Text Clustering Using Concept-based Mining Model", IEEE International Conference on Data Mining, ICDM, Hong Kong 2006.
- Shady H. Shehata, Fakhri Karray, Mohamed Kamel, "Concept-based Mining Model for Learning Objects", LORNET Scientific Conference on Learning Systems of the Future: Integrating Knowledge and Services (I2LOR06), Montreal, Canada 2006.

Demos

- Shady Shehata, Fakhri Karray, Mohamed Kamel, "Web-Based Text Mining System", 4th annual LORNET Scientific Conference I2LOR-07 in Montreal (Nov 4-7), 2007. **Third Position Award For Best Demo**
- Shady Shehata, Fakhri Karray, Mohamed Kamel, "Enhancing Text Clustering Using Concept-based Mining Model", IEEE International Conference on Data Mining, ICDM, Hong Kong 2006.
- Shady H. Shehata, Fakhri Karray, Mohamed Kamel, "Concept-based Mining Model for Learning Objects", LORNET Scientific Conference on Learning Systems of the Future: Integrating Knowledge and Services (I2LOR06), Montreal, Canada 2006.
- Shady H. Shehata, Fakhri Karray, Mohamed Kamel, "Conceptual Search Engine", LORNET Scientific Conference on E-Learning for the Future from Content to Services (I2LOR05), Vancouver, Canada 2005.

Appendix A

Appendix

A.1 Text Clustering Results

The results illustrated in Figures(A.1, A.2, A.3, A.4, A.5, A.6, A.7, and A.8) show the improvement on the clustering quality obtained by the concept-based model for the different datasets and clustering techniques.

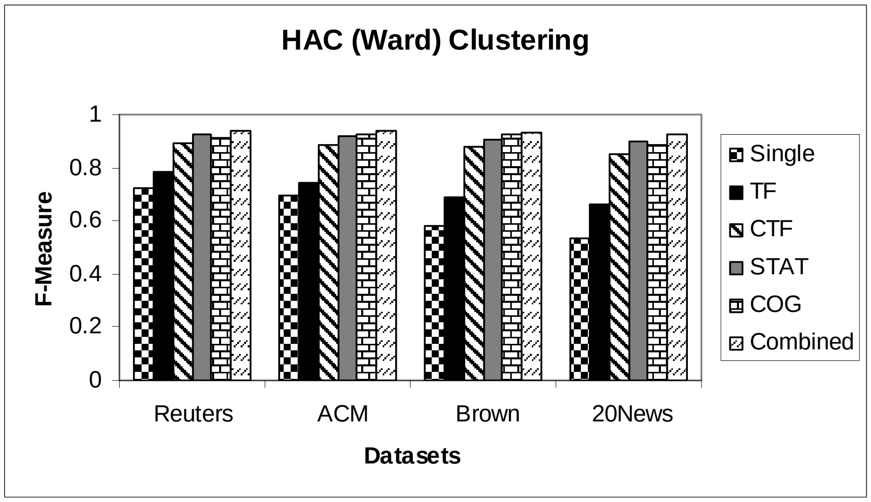

Figure A.1: HAC-Ward Clustering (F-Measure)

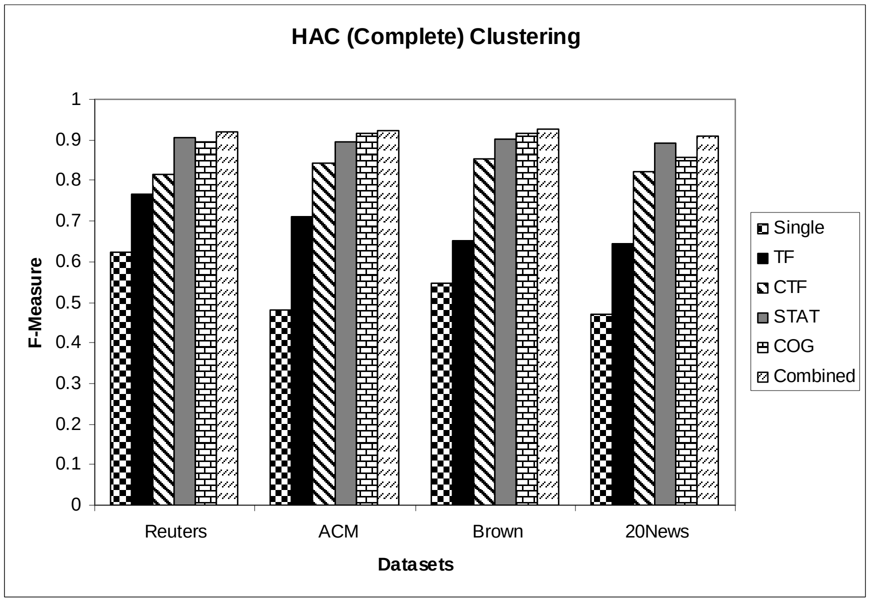

Figure A.2: HAC-Complete Clustering (F-Measure)

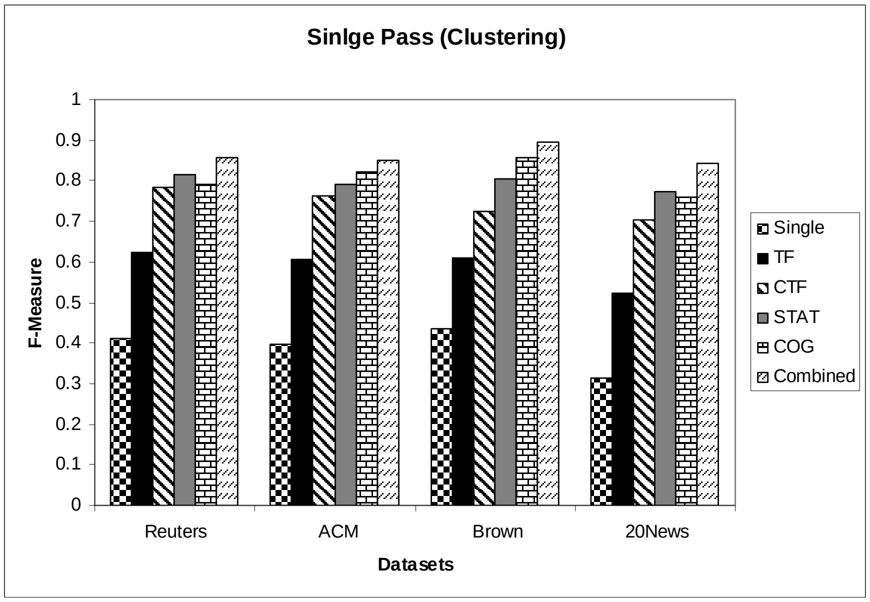

Figure A.3: Single Pass Clustering (F-Measure)

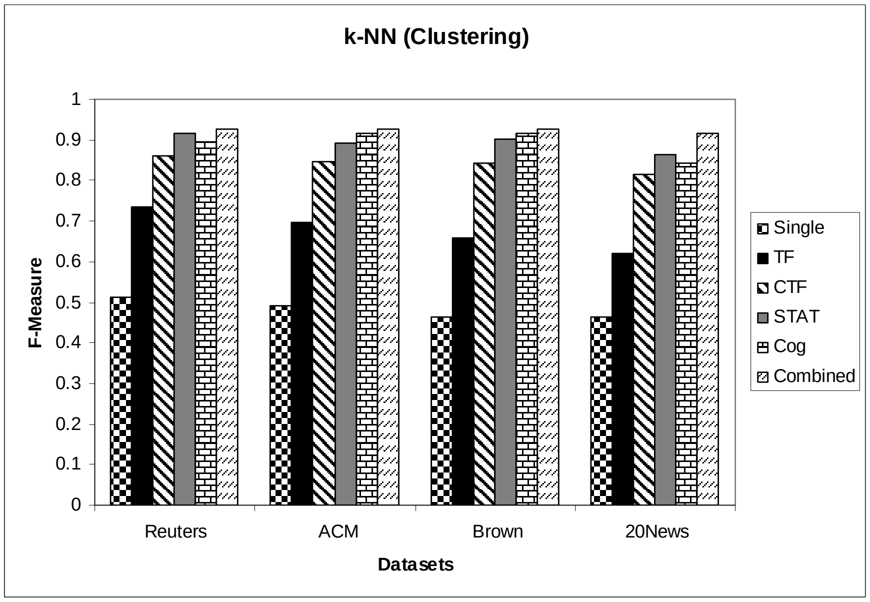

Figure A.4: kNN Clustering (F-Measure)

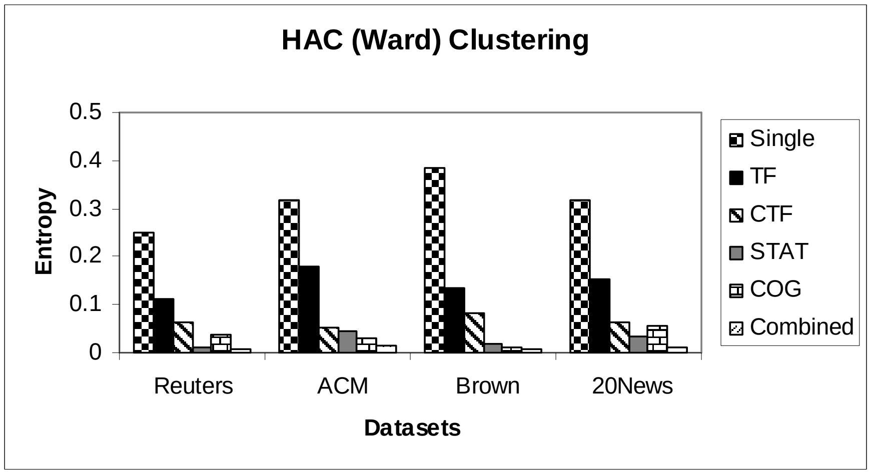

Figure A.5: HAC-Ward Clustering (Entropy)

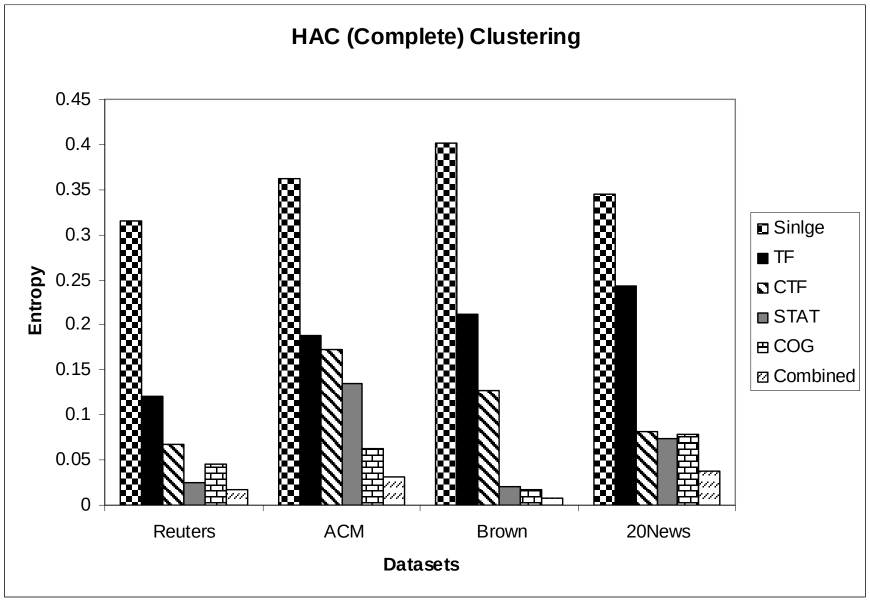

Figure A.6: HAC-Complete Clustering (Entropy)

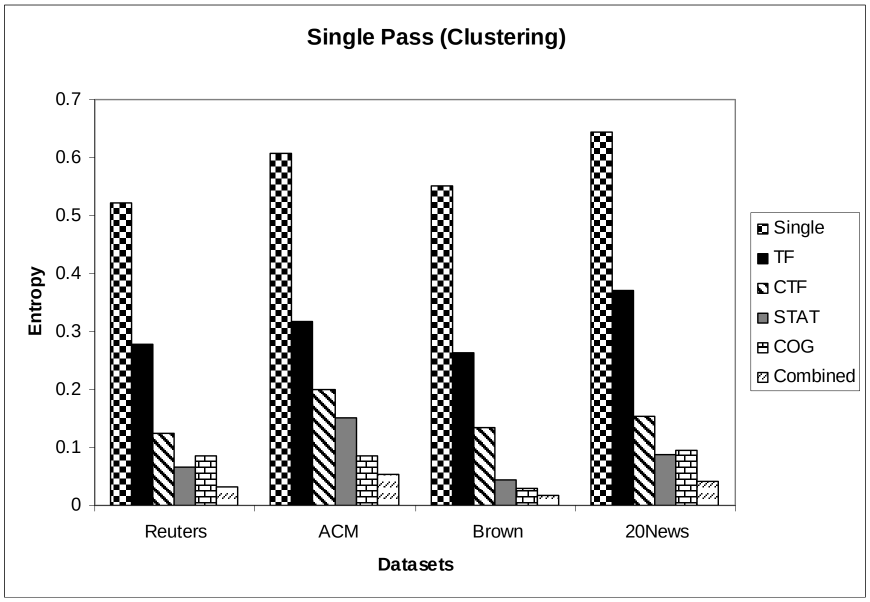

Figure A.7: Single Pass Clustering (Entropy)

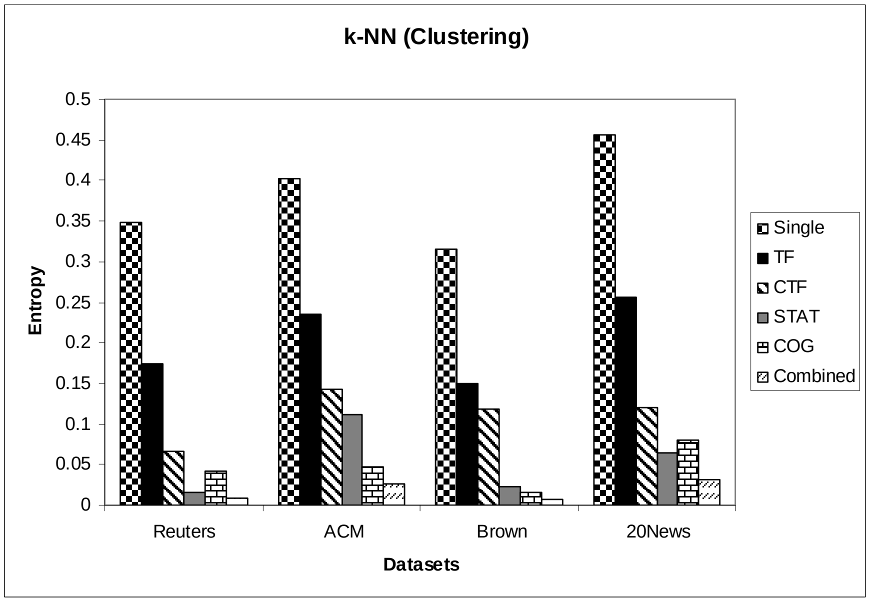

Figure A.8: kNN Clustering (Entropy)

A.2 Text Categorization Results

The results illustrated in Figures(A.9, A.10, A.11, A.12, A.13, A.14, A.15, A.16, A.17, A.18, A.19, and A.20) show the improvement on the categorization quality obtained by the concept-based model for the different datasets and categorization techniques.

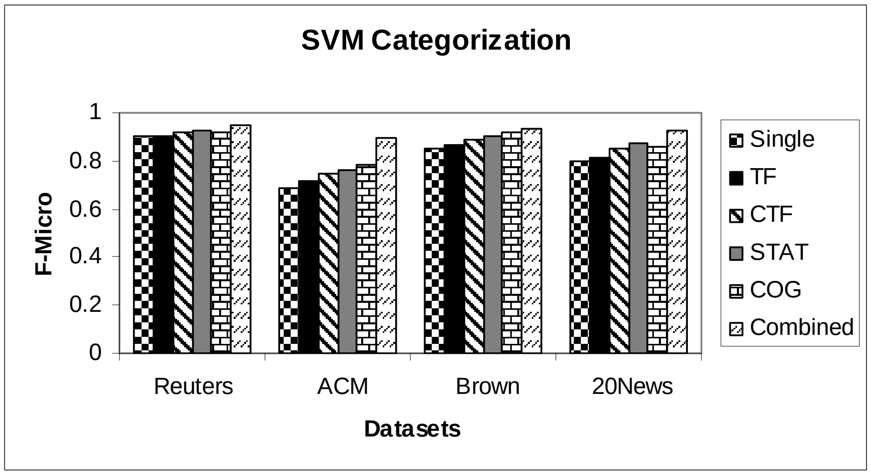

Figure A.9: SVM Categorization (F-Micro)

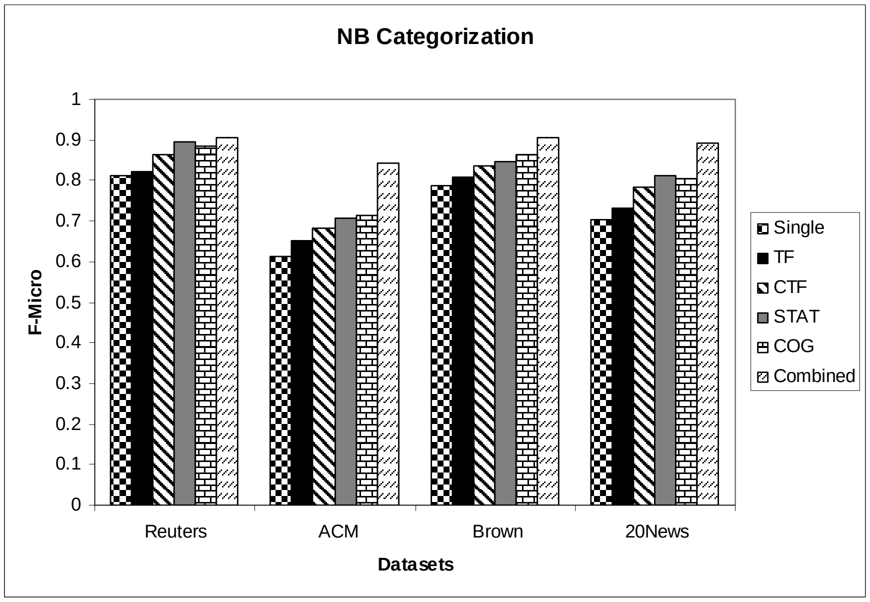

Figure A.10: NB Categorization (F-Micro)

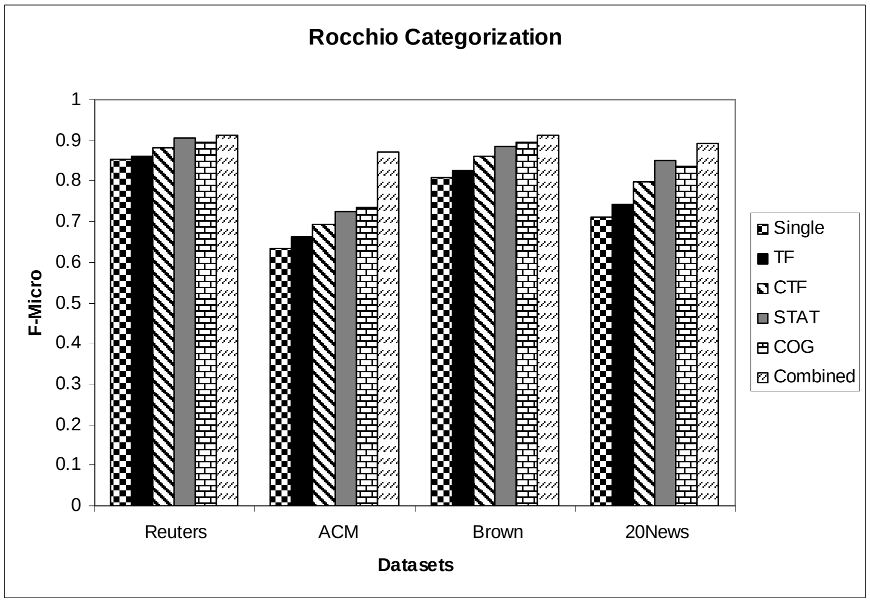

Figure A.11: Rocchio Categorization (F-Micro)

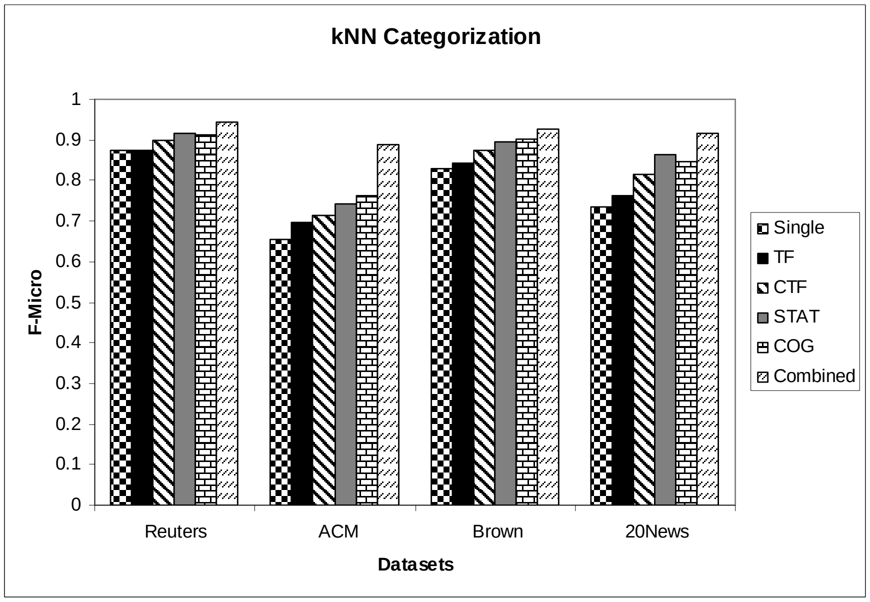

Figure A.12: kNN Categorization (F-Micro)

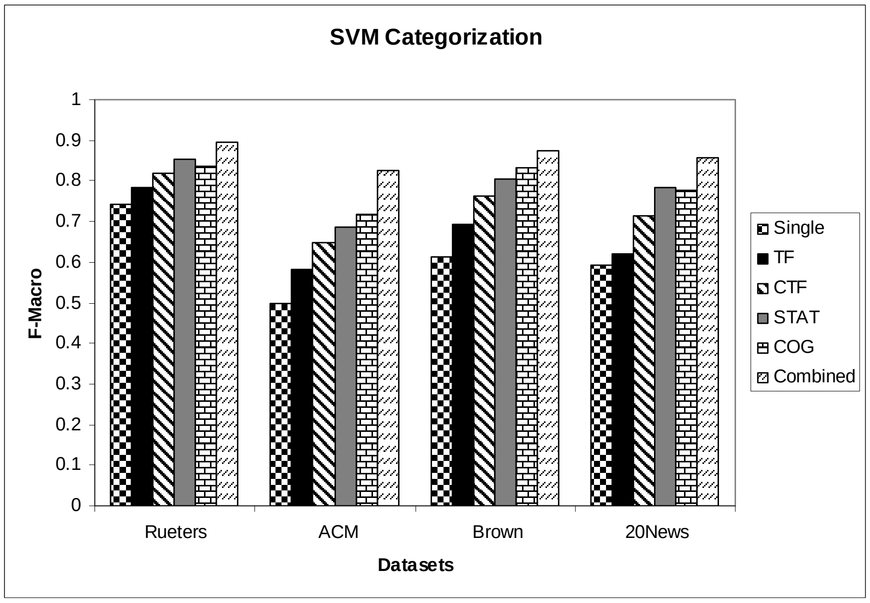

Figure A.13: SVM Categorization (F-Macro)

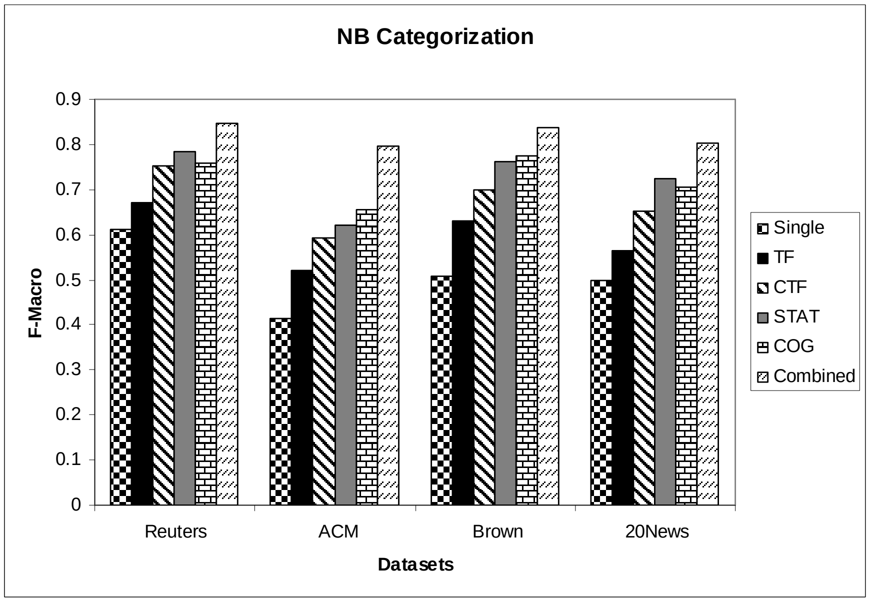

Figure A.14: NB Categorization (F-Macro)

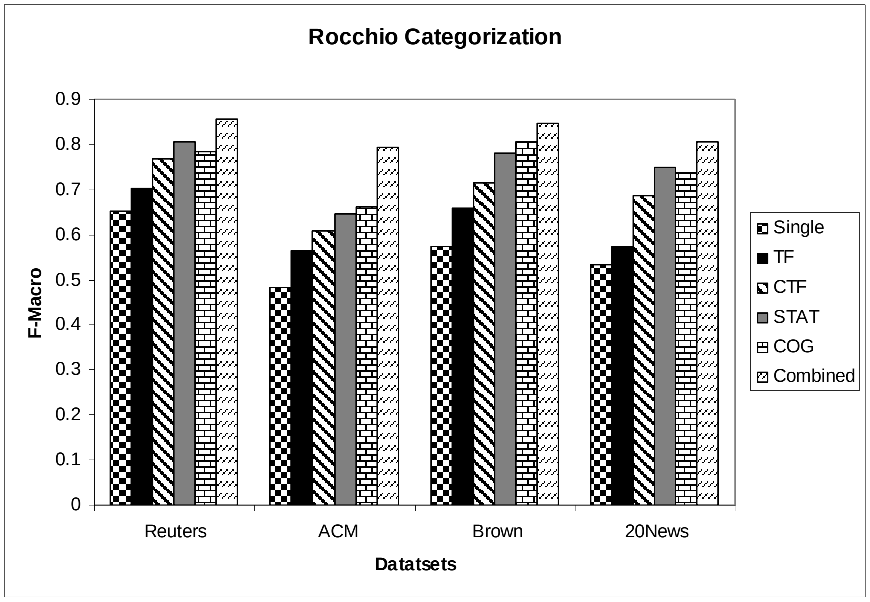

Figure A.15: Rocchio Categorization (F-Macro)

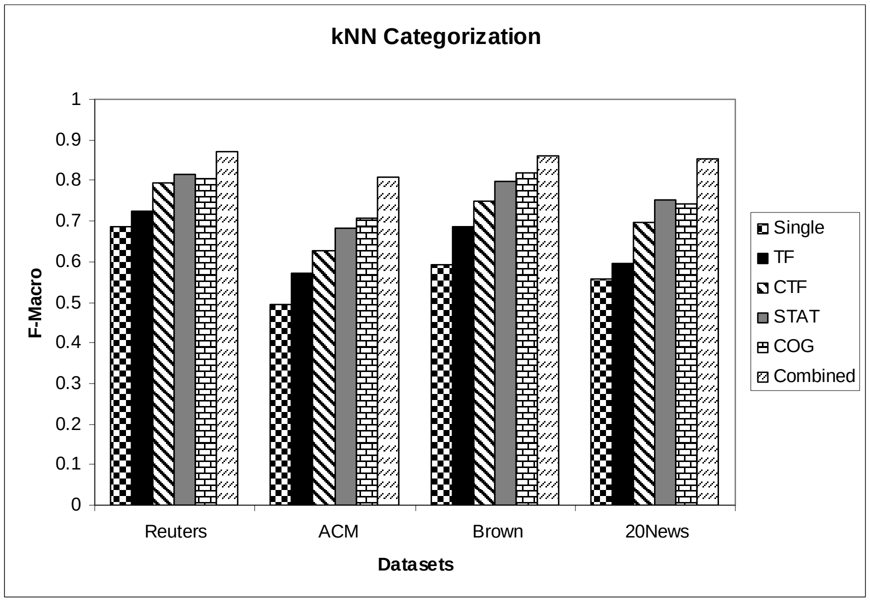

Figure A.16: kNN Categorization (F-Macro)

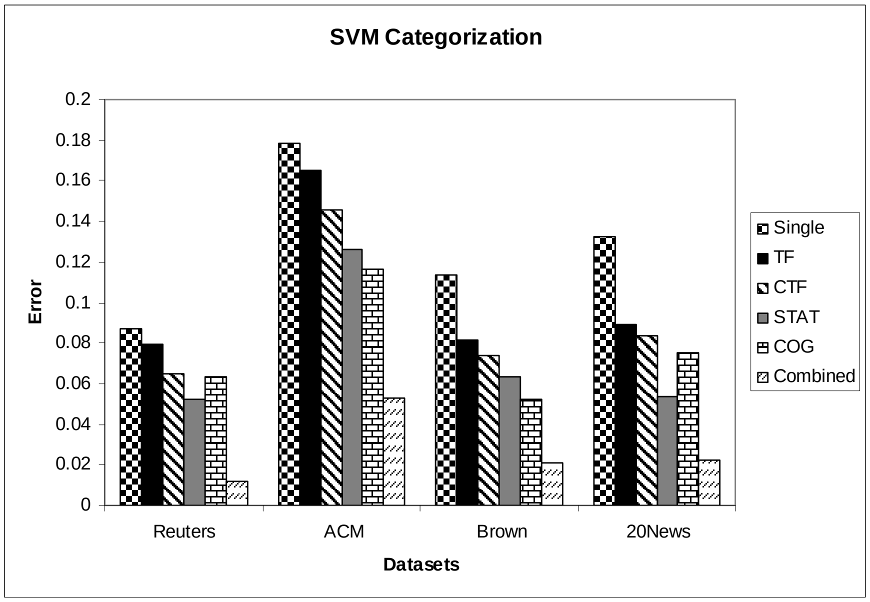

Figure A.17: SVM Categorization (Error)

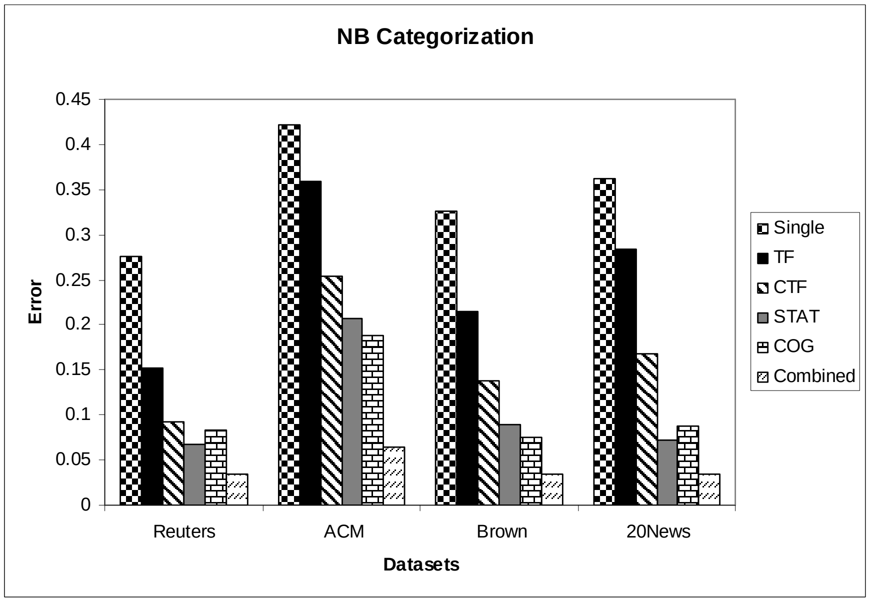

Figure A.18: NB Categorization (Error)

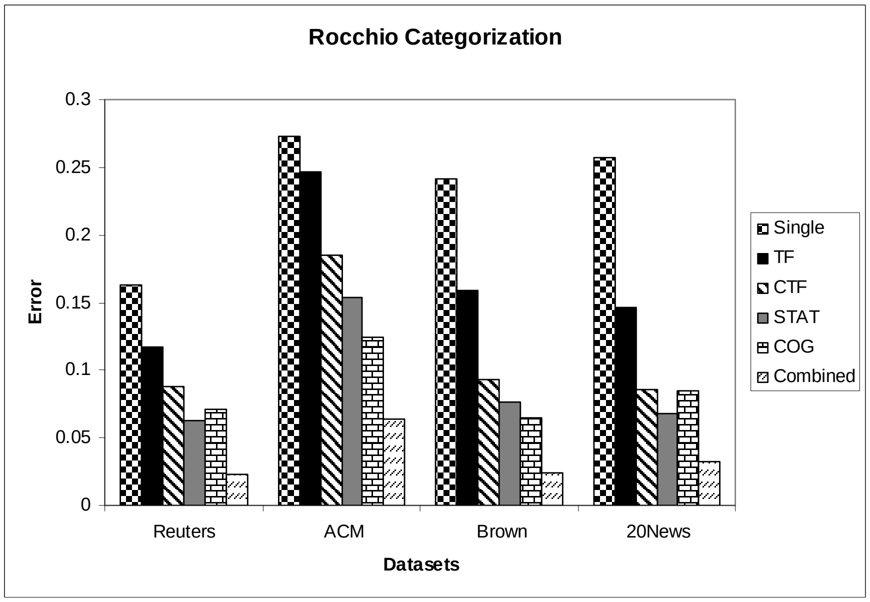

Figure A.19: Rocchio Categorization (Error)

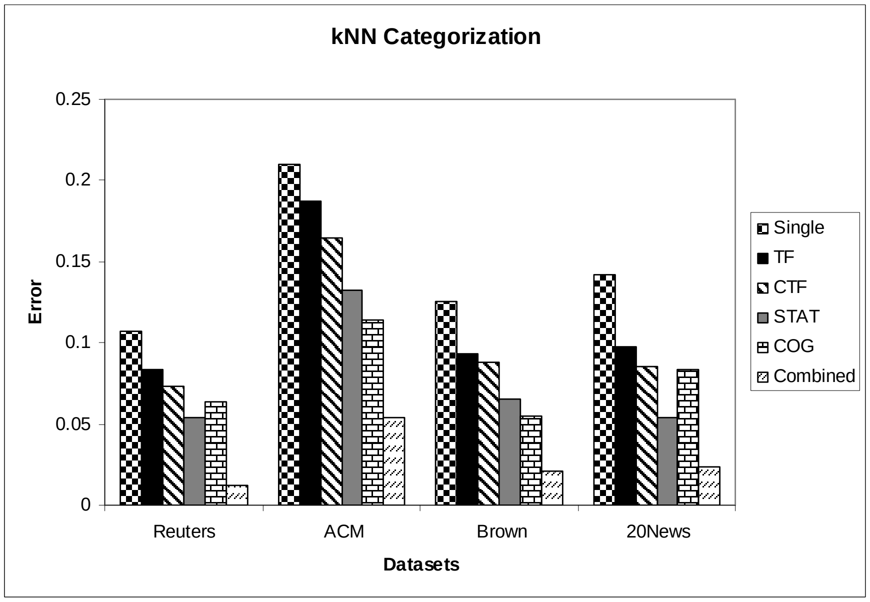

Figure A.20: kNN Categorization (Error)

A.3 Text Retrieval Results

The results illustrated in Figures(A.23, A.24, A.21, A.22, A.25, and A.26), show the improvement on the retrieval quality obtained by the concept-based model for the different datasets and retrieval engines.

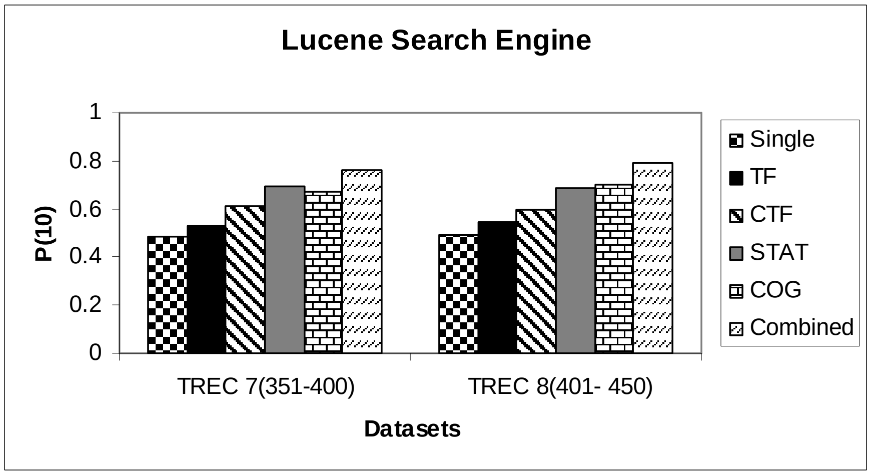

Figure A.21: Lucene Search Engine (P10)

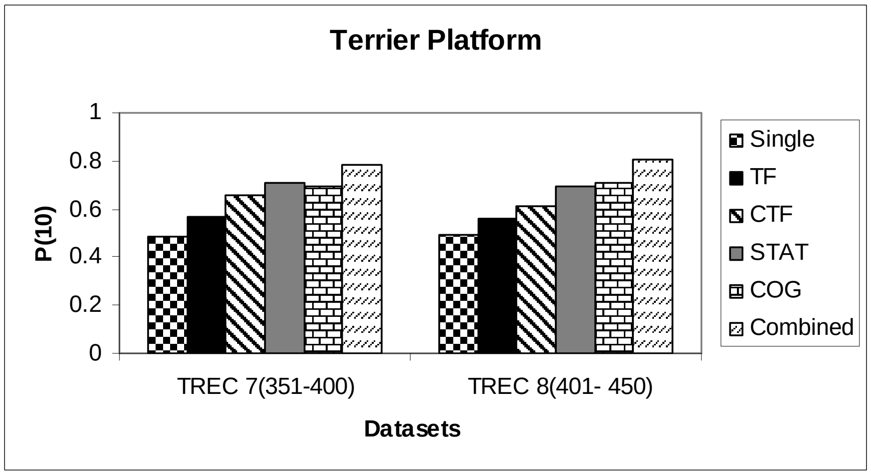

Figure A.22: Terrier Search Platform (P10)

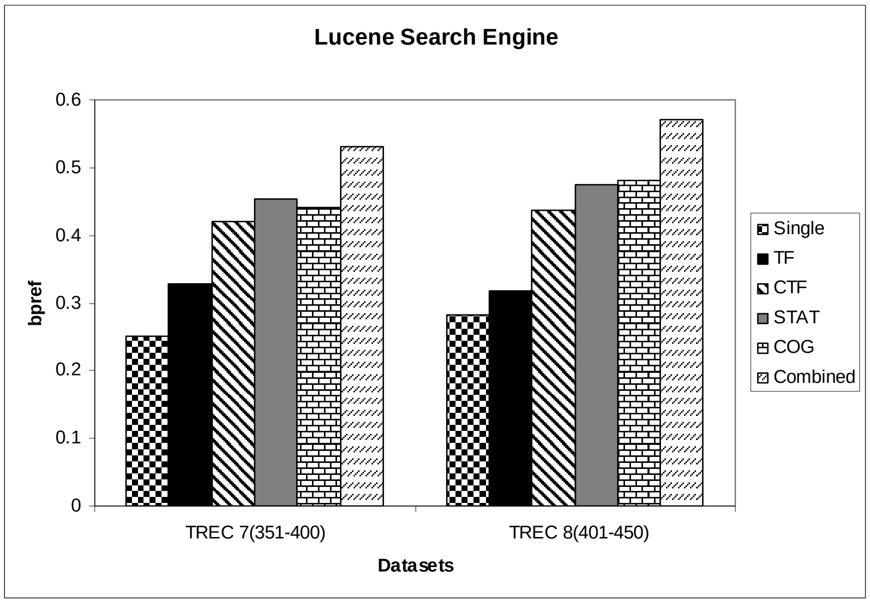

Figure A.23: Lucene Search Engine (bpref)

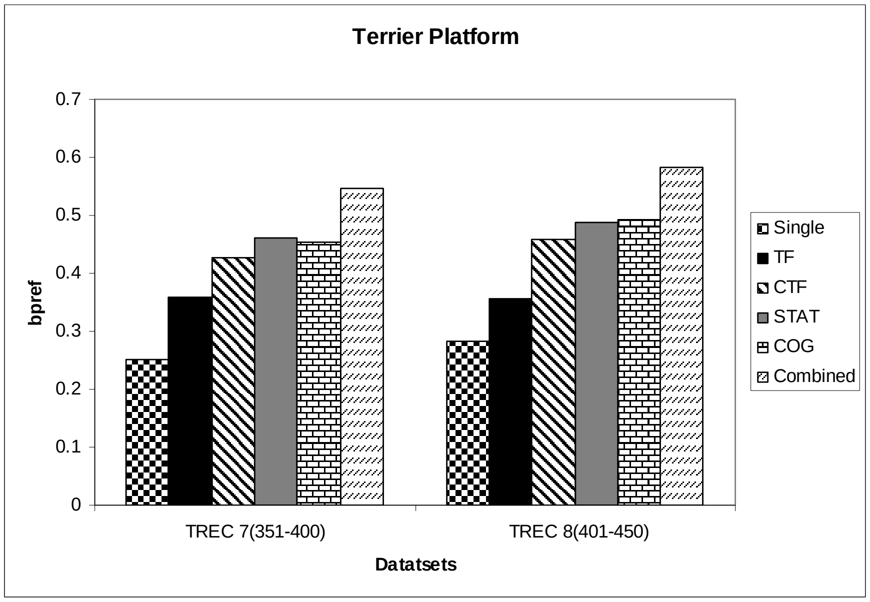

Figure A.24: Terrier Search Platform (bpref)

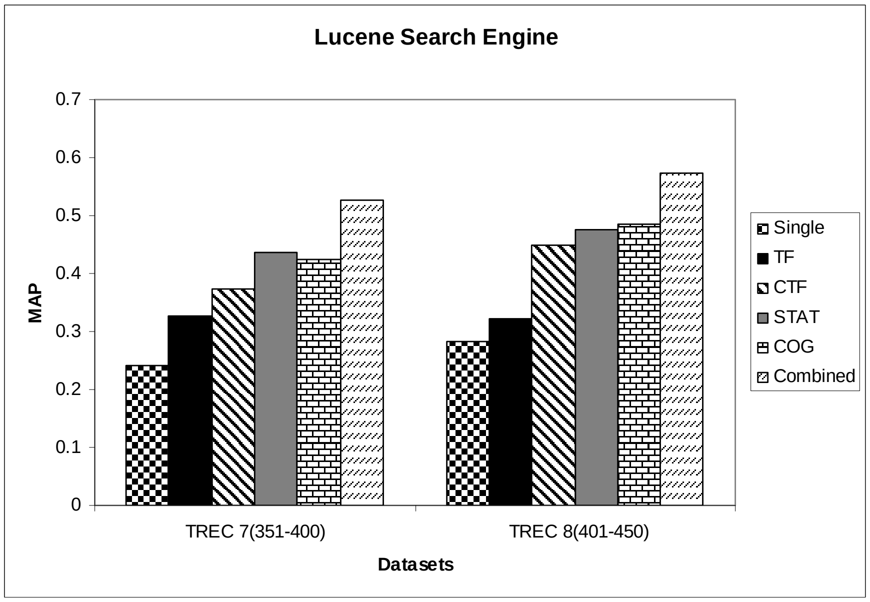

Figure A.25: Lucene Search Engine (MAP)

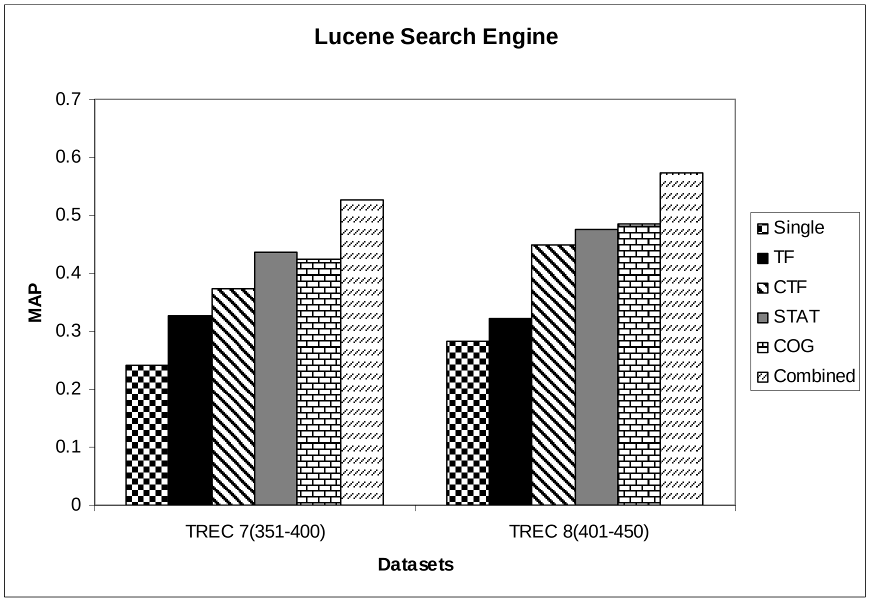

Figure A.26: Terrier Search Platform (MAP)

Bibliography

[1] M. Lynch, "e-business analytics, Indepth Report," 20 November 2000.

[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web." Scientific American, 2001.

[3] R. Feldman and I. Dagan, "Knowledge discovery in textual databases (kdt)," in *First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, Montreal, Canada, 1995, pp. 112–117.

[4] G. Salton, A. Wong, and C. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, November 1975.

[5] C. Manning. and H. Schutze, *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

[6] C. Y. Suen, "N-gram statistics for natural language understanding and text processing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 164–172, April 1979.

[7] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, July 1975.

[8] M. Steinbach, G. Karypis, and V. Kumar, "Scatter/gather:a cluster-based approach to browsing large document collections," in *16th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993.

[9] ——, "A comparison of document clustering techniques," in *Knowledge Discovery and Data Mining (KDD) Workshop on TextMining*, August 2000.

[10] K. Cios, W. Pedrycs, and R. Swiniarski, "Data mining methods for knowledge discovery." Boston: Kluwer Academic Publishers, 1998.

[11] C. J. van Rijsbergen, *Information Retrieval.* London, second edition.: Buttersworth, 1979.

[12] G. Kowalski, *Information Retrieval Systems Theory and Implementation.* Kluwer Academic Publishers, 1997.

[13] C. Buckley and A. F. Lewit, "Optimization of inverted vector searches," in *SIGIR*, 1985, pp. 97–110.

[14] D. R. Cutting, J. O. Pedersen, D. Karger, and J. W. Tukey, "Scatter/gather: A cluster-based approach to browsing large document collections," in *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1992, pp. 318–329.

[15] O. Zamir, O. Etzioni, O. Madani, and R. M. Karp, "Fast and intuitive clustering of web documents," in *KDD*, 1997, pp. 287–290.

[16] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," in *ICML*, D. H. Fisher, Ed. Morgan Kaufmann, 1997, pp. 170–178.

[17] C. Aggarwal, S. Gates, and P. Yu, "On the merits of bilding categorization systems by supervised clustering," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, S. Chaudhuri and D. Madigan, Eds. N.Y.: ACM Press, Aug. 15–18 1999, pp. 352–356.

[18] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data.* Englewood Cliffs: Prentice Hall, 1988.

[19] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, ser. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons Inc., 1990.

[20] K. J. Cios, W. Pedrycz, and R. W. Swiniarski, "Data mining methods for knowledge discovery," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1533–1534, 1998.

[21] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* IEEE Computer Society Press, 1991.

[22] D. R. Hill, "A vector clustering technique," in *FID-IFIP , Samuelson(ed), N-H 1968*, 1967.

[23] C. E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948, continued 27(4):623-656, October 1948.

[24] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering," in *KDD*, 1999, pp. 16–22.

[25] Z. Xu, X. Xu, K. Yu, and V. Tresp, "A hybrid relevance-feedback approach to text retrieval," in *ECIR*, ser. Lecture Notes in Computer Science, F. Sebastiani, Ed., vol. 2633. Springer, 2003, pp. 281–293.

[26] P. R. Halmos, *Naive Set Theory.* New York: Springer, 1974.

[27] J. E. Goin, "Classification bias of the K-nearest neighbor algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 3, pp. 379–381, May 1984.

[28] V. Vapnik, *Statistical Learning Theory.* Wiley, 1998.

[29] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[30] A. F. Smeaton, "An overview of information retrieval," in *Information Retrieval and Hypertext*, M. Agosti and A. F. Smeaton, Eds. Dordrecht, NL: Kluwer Academic Publishers, 1997, pp. 3–25.

[31] S. Robertson, S. Walker, and M. Beaulieu, "Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive track."

[32] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau, "Okapi at TREC," in *Text REtrieval Conference*, 1992, pp. 21–30.

[33] D. Hull, "Using statistical testing in the evaluation of retrieval experiments," in *Proceedings of Special Interest Group on Information Retrieval (ACM SIGIR)*, 1993.

[34] C. Buckley and E. M. Voorhees, "Retrieval evaluation with incomplete information," in *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval.* New York, NY, USA: ACM Press, 2004, pp. 25–32. [Online]. Available: http://portal.acm.org/citation.cfm?id=1009000

[35] J. F. Allen, *Natural Language Understanding*, second edition ed. Redwood City: CA: Benjamin/Cummings, 1994.

[36] D. Jurafsky and J. H. Martin, *Speech and Language Processing.* Prentice Hall Inc., 2000.

[37] C. Fillmore, "The case for case," in *Universals in Linguistic Theory*, E. Bach and R. Harms, Eds. Holt and Rinehart and Winston, 1968, pp. 1–88.

[38] L. Levin, "Operations on lexical forms: Unaccusative rules in germanic languages," Ph.D. dissertation, MIT and Department of Linguistics and Philosophy, 1985b.

[39] L. Beth, *English Verb Classes and Alternations A Preliminary Investigation.* The University of Chicago Press, 1993.

[40] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *Proceedings of the 1998 COLING-ACL Conference*, Montreal, Canada, 1998, pp. 86–90.

[41] K. Paul and P. Martha, "From treebank to propbank," in *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, Spain, 2002.

[42] P. Kingsbury, M. Palmer, and M. Marcus, "Adding semantic annotation to the penn treebank," in *Proceedings of the Human Language Technology Conference (HLT'02)*, 2002.

[43] K. Kipper, H. Dang, and M. Palmer, "Class-based construction of a verb lexicon," in *AAAI-2000 Seventeenth National Conference on Artificial Intelligence*, Austin, TX, 2000.

[44] P. Kingsbury and K. Kipper, "Deriving verb-meaning clusters from syntactic structure," in *HLT/NAACL Workshop on Text Meaning*, 2003.

[45] R. Al-Halimi, R. C. Berwick, J. F. M. Burg, M. Chodorow, C. Fellbaum, *et al.*, *WordNet: An Electronic Lexical Database*, C. Fellbaum, Ed. Cambridge: MIT Press, 1998.

[46] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Computational Linguistics*, vol. 28, no. 3, 2002.

[47] M. Collins, "Head- driven statistical model for natural language parsing," Ph.D. dissertation, University of Pennsylvania, 1999.

[48] S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin, and D. Jurafsky, "Shallow semantic parsing using support vector machines," in *the Proceedings of the Human Language Technology/North American Association for Computational Linguistics (HLT/NAACL-2004)*, Boston, 2004.

[49] M. Surdeanu, "Using predicate argument structures for information extraction," in *ACL*, Sapporo, Japan, 2003.

[50] E. Schapire, “The boosting approach to machine learning,” in *Proceedings of the MSRI Workshop on Nonlinear Estimation and Classification*, Berkeley, CA, 2002.

[51] A. Cynthia, R. Levy, and C. D. Manning, “A generative model for semantic role labeling,” in *European Conference on Machine Learning (ECML)*, 2003, pp. 397–408.

[52] K. Hacioglu, S. Pradhan, W. Ward, J. H. Martin, and D. Jurafsky, “Semantic role labeling by tagging syntactic chunks,” in *the Proceedings of the Eighth Conference on Natural Language Learning (CoNLL-2004)*, Boston, 2004.

[53] M. King, “Evaluating natural language processing systems,” *Commun. ACM*, vol. 39, no. 1, pp. 73–79, 1996.

[54] J. F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* Pacific Grove, CA: Brooks Cole Publishing Co., August 1999.

[55] T. R. Gruber, “A translation approach to portable ontologies,” *Journal of Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[56] ——, “Toward principles for the design of ontologies used for knowledge sharing,” in *workshop on Formal Ontology*, March 1993.

[57] S. U. O. W. G. S. WG), *OpenCyc Project*, IEEE.

[58] G. Frank, A. Farquhar, R. Fikes, V. Heiberg, and S. Zamler, *The World Fact Book Knowledge Base Project*, Stanford University.

[59] *NLM’s Unified Medical Language Project*, National Library of Medicine, 1999.

[60] D. Brickley and R. V. Guha, *RDF Vocabulary Description Language 1.0: RDF Schema*, World Wide Web Consortium (W3C) Recommendation, 2004.

[61] M. Paolucci, O. Shehory, and K. Sycara, “Interleaving planning and execution in a multiagent team planning environment,” *Electronic Transactions of Artificial Intelligence*, 2001.

[62] D. Fensel, F. Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider, “Oil: An ontology infrastructure for the semantic web,” *IEEE Intelligent Systems*, 2001.

[63] S. Bechhofer, C. Goble, and I. Horrocks, “Daml+oil is not enough,” *First Semantic Web Working Symposium (SWWS’01)*, 2001.

[64] M. Dean *et al.*, *OWL Web Ontology Language Reference*, World Wide Web Consortium (W3C) Recommendation, 2003.

[65] M. Viezzer, "Ontologies and problem-solving methods and ontology learning," in *European Conference on Artificial Intelligence (ECAI)*, August 2000.

[66] N. Houser, D. D. Roberts, and J. V. Evra, *Studies in the Logic of Charles Sanders Peirce.* Bloomington: Indiana University Press, 1997.

[67] M. Minsky, *Semantic Information Processing.* Cambridge, MA: MIT Press, 1968.

[68] P. Kingsbury and M. Palmer, "Propbank: the next level of treebank," in *Proceedings of Treebanks and Lexical Theories*, 2003.

[69] A. Strehl, J. Ghosh, and R. Mooney, "Impact of similarity measures on web-page clustering," in *Proceedings of 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI)*, 2000, pp. 58–64.

[70] W. Francis and H. Kucera, *Manual of information to accompany a standard corpus of present-day edited american english, for use with digital computers*, 1964.

[71] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, N.J., 1988.

[72] S. Y. Lu and K. S. Fu, "A sentence to sentence clustering procedure for pattern analysis," *IEEE Transactions on Systems Mans and Cybernetics*, vol. 8, pp. 381–389, 1978.

[73] "Apache jakarta lucene search engine (version 1.3), http://lucene.apache.org/."

[74] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma, "Terrier: A High Performance and Scalable Information Retrieval Platform," in *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.

[75] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *Proceedings of ECML-98, 10th European Conference on Machine Learning*, C. Nédellec and C. Rouveirol, Eds., no. 1398. Chemnitz, DE: Springer Verlag, Heidelberg, DE, 1998, pp. 137–142.