2-1-2004

Text categorization based on phrase indexing

Hany Seif Habib
The American University in Cairo AUC

The American University in Cairo
School of Sciences and Engineering

Text Categorization Based on Phrase Indexing

A Master Thesis Submitted by

Hany Seif Habib
B.Sc. in Computer Science

Under Supervision of

Prof. Dr. Amr Goneid

To

Computer Science Department

In Partial Fulfillment of the requirements for the degree of
Master of Science in Computer Science

November 2003

2003/61

The American University in Cairo
Computer Science Department
Master Program

November 2003

Text Categorization Based on Phrase Indexing

M.Sc. Thesis
Submitted by
Hany Seif Habib

Supervised by
Prof. Dr. Amr Goneid

Approvals:

Committee Chair / Advisor: **Dr. Amr Goneid**
Affiliation [redacted]

Committee Reader / Examiner: **Dr. Amr El Kadi**
Affiliation [redacted]

Committee Reader / Examiner: **Dr. A. Khalil**
Affiliation [redacted]

Committee Reader / Examiner: **Dr. Mokhtar Boshra**
Affiliation [redacted]

Department Chair: **Dr. N. Mikhail** [redacted]
Date:

Dean: **Dr. Fadel Assabghy**
Date: January 28, 2004 [redacted]

Acknowledgement

First, I would like to acknowledge my supervisor Dr. Amr Goneid for his continuos guidance and support throughout my work in this thesis. He has been very helpful to me directing me towards the correct and best methodologies to be used in my research.

I would like also to acknowledge Dr. Amr El Kadi and Dr. Amr El Abbadi for their constructive support in my thesis presentation. Their comments and directions were very helpful to me in continuing research in a solid and precise direction.

I would like to acknowledge all Computer Science Department instructors and chairman for their continuos support for me as well as all other students during my study period to be able to fulfill all program requirements successfully.

I would like also to acknowledge The American University in Cairo and its entire staff members and professors for their persistence and efforts in maintaining a high standard of quality of education.

Finally I would like to acknowledge my wife and parents for their continuos support and encouragement. You were of great help to me facilitating the time and environment to work on this thesis and the whole master program.

Thank you all very much.

Abstract

This thesis investigates Information Retrieval (IR) field of research with focus on Text Categorization (TC) area. We study the available techniques and models used for classification of text documents to a predefined set of categories. We use a subset of Reuters – 21578 test collection for our research. We use the first 150 documents for training and the following 100 for testing.

We use pre-processing steps such as parsing, stop-word removal, and stemming using porter stemmer. We identify all possible phrases in a document during the pre-processing stage. We use only adjacent phrases of size two. We learn frequency information during the pre-processing stage for documents of the training set. We construct the index lists after applying a feature reduction function on extracted features (terms and phrases). We assign weights to relate index features to categories. The weights and index lists are then used to classify test set documents. Categorization results are compared with relevance judgements to evaluate the performance of our categorization methodology.

We propose a feature reduction technique to reduce the number of initially extracted features by selecting features of high categorization quality. The result of the feature reduction process is a set of index-features. This function uses feature frequency and feature document frequency combined with a new feature category frequency technique. Coefficients of the proposed feature reduction formula control the output of the formula allowing more or less features to be selected. We try several coefficient combinations and achieve over 98% reduction for terms and over 99% reduction for phrases and at the same time achieving high precision and recall values.

Our primary goal is to provide a method that achieves high precision categorization based on phrases. We want to prove that phrases can be used alone as an independent highly precise classifier. Our secondary goal is to propose other term-based techniques that perform at least as good as term-classifiers with less complexity. We propose a classifier based on the existence of category names in a document. This classifier considers a document relevant to a category if category name exists in the document. We will also propose title classifier that gives higher weight to index terms found in a document title. We also used the same phrase classifier concept on terms to obtain categorization results based on terms only to be

able to compare our proposed techniques with term categorization using the same testing environment.

We evaluate precision and recall for each technique individually to be able to compare them together and also study the behavior of combining these techniques and the resulting effect on the total system performance.

Our phrase classifier achieves average of 89% precision and 35% recall. Other researches based on phrase categorization achieve much less precision. For example, a research on statistical and syntactic phrases achieves an average of 54% precision at recall of 30% and a maximum of 85% at 0% recall level [19].

Using category-term classifier alone achieves 67% precision and 37% recall. Using title classifier independently achieves average of 26% precision and 46% recall. Categorization based on term-classifier only achieves averages of 12% precision and 89% recall.

Keywords

Information Retrieval, Text Categorization, Test Collection, Precision, Recall, Stop-words, Stemming, Term, Phrase, Statistical Phrase, Relevance.

Table of Contents

List of Tables

List of Figures and Graphs

Chapter 1

Introduction

Chapter 1
Introduction

We are living in the information era. Digital media, the Internet and other communications technological advances made it possible for many people and organizations to share their information and knowledge. Modern computing and network technology made it possible to organize and store large amounts of data and pass them around the world with minimal effort [5]. Since the availability of information is no longer a problem, sorting out this information and finding out what is relevant and what is not to a user need became a complex problem that troubled information technology researchers.

Information is available in many forms such as text documents, video, voice, pictures, etc. However, the text form of information is still considered as the most dominating form of them all. The field that is interested in finding information, storing them in an indexed form for easy retrieval, retrieving the relevant documents to a user interest, and presenting them is known as Information Retrieval. Information Retrieval field of research has many practical applications that made it very important for many researchers and corporations to dedicate their effort, time, and money to achieve advances in this field. These practical applications include digital libraries, search engines, question answering, summarization, agents, recommendation systems, automatic organization, cross-language retrieval, data mining, multi-database search, and knowledge management.

Information Retrieval is considered to be one of the best-organized fields of research. Pioneers working in this field established a standard base for others to work on. They created what they called test collections. A test collection is a set of documents that are stored in a collection specific structure and format that is available for researchers to use as a test data. What is important about these collections is that they include a human indexed information that would make it possible for researchers to compare their results to the actual correct results.

Since the most dominating form of information is text documents, it is our main concern in this thesis. We also chose English text documents to be our area of research, since available test collections are in English text and it is the main language used for many researchers, so we will be able to compare results to others.

1.1 Information Retrieval:

Information Retrieval (IR) is generally concerned with representation, storage, organization of, and access to information items [23]. The aim of the process is to provide an easy access to users and effective and efficient retrieval for related information to a user need. On the other hand information representation is not an easy task. A user need might be a sentence such as: " Find all documents containing information related to football teams in Egypt that have won any tournament during the past 20 years and have scored at least 50 goals in each tournament. For documents to be relevant, they must include players and coaches names, location of the matches played, and the results of the matches." Most if not all search engines will not be able to understand such need and retrieve the correct documents. Therefore, an easier representation to the user need is required. This representation depends on the system used; however, the main concept is to represent the user need by a few words (keywords) that better represent or describe the user need. This representation is known as a user query. A search engine or IR system would typically have the information stored and indexed in some way. It will process the user query matching it with its indexes and retrieve the most relevant documents to the user.

Therefore we can say that an IR system basically consists of two main parts. The first part processes the documents, stores them, and indexes them for easy retrieval later on, and we call this part text document classification. A branch of IR that is concerned only with this area of research is known as text categorization (TC). The second part processes the user query and finds related information to that query and retrieves them to the user. Some IR systems just match the query with whatever information it contains in its indexes. This usually happens in static environments such as a library. This process is known as ad-hoc search problem. The other type of search is known as routing problem where the system will use documents classification for new documents dynamically added to the system and retrieve what it thinks relevant to the user query. This usually happens in more dynamic environments such as newsgroups.

In the Information Retrieval field of research there are many keywords and definitions that are important for us to understand before we proceed because they will be used excessively in our discussion.

A document collection is a set of documents grouped together and stored in a specific way to that collection. A library is a document collection where books and magazines are its documents. Other examples include journals, encyclopedias, and newsgroups. A text collection is a document collection where all documents stored in the collection are in text format. A test collection as previously mentioned is a set of documents stored in a specific format where each document has information that relates it to categories it represents. Some of these test collections also contain standard queries and list of the relevant documents for each query. These test collections are used in IR research where researchers can compare their results with the ones stored within the test collection.

A training set is the set of documents used for training an IR system to be able to understand how it should classify documents fed into it later on and how could it match queries to classified documents. A test set is the set of documents fed into an IR system to test its ability to classify documents. Usually both training and testing sets are obtained from test collections where results of both are available beforehand. The results for a training set are used to train the system while the results of the test set are used as benchmarking to measure the system performance.

A relevant document is a document that contains relevant information to the user need represented by a query. The target of an IR system is to retrieve all relevant documents to the user that match his query. For TC a document is considered relevant if it contains relevant information to one or more categories. A category is a main topic that documents represent or to which they are relevant. Text categorization maps text documents to a pre-defined set of categories. Usually defining the categories is a human-handled process. However, there are other research areas interested in finding out more categories and adding them to their list of defined categories.

1.2 Information vs. Data Retrieval:

A Database system is an example of Data Retrieval systems. These systems mainly are concerned with documents that contain keywords in user queries. In most cases this does not satisfy user information needs. Information Retrieval is on the other hand concerned with retrieving documents containing information about the user subject represented by his query [23]. Therefore, for a data retrieval system it is a

views can be obtained for a document starting by full-text view to the least set of words (index terms) that can represent the document. The full-text view is the most complete logical view, but using it implies high computational costs. The human-generated index-terms view is the most concise view, but using it might lead to poor quality retrieval [23].

The following diagram by Yates and Neto [23] (figure 1) shows different processing stages for a document and example logical view obtained after each stage. It also shows that any stage is optional and can be ignored, but if processed, it produces a different logical view of the same document.

The document starts by containing text and structure. The structure recognition stage recognizes the structure of the document such as chapters, references, titles, etc. The output of that stage is the structure view and the text that is fed into a later stage. The later stage recognizes the accents, spacing, etc. and produces the full-text view. Other following stages include stop-words removal stage which removes stop words from the text such as "a", "the", etc. These words represent a large percentage of document words and do not imply any meaning related to any specific document since they are included in all other documents belonging to all other categories. Noun grouping stage removes verbs and adjectives and only keeps noun words. Stemming stage converts all words into their linguistic roots, thus obtaining a less set of terms. The final stage is the indexing stage, which indexes the terms, and obtains the index-terms logical view. This later stage could be manually or automatically done.

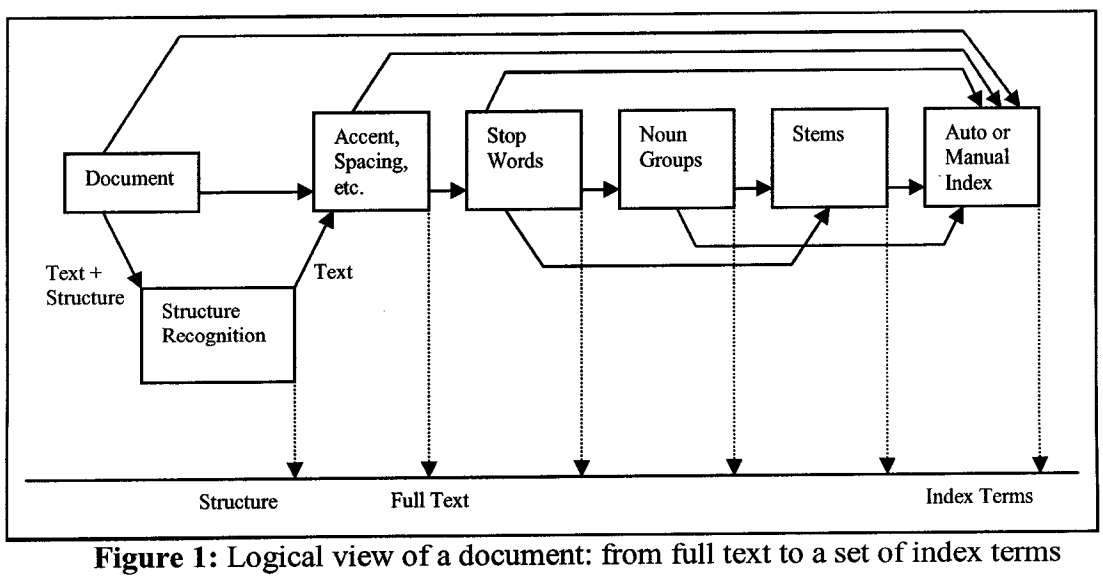

Figure 1: Logical view of a document: from full text to a set of index terms

The main target of any or all of these stages is to obtain the smallest set of words that better represent the document. Since the smallest linguistic entity that carry meaningful information is the word, it is considered as the basic element used in IR field. Other entities such as phrases and sentences carry more meaningful information but they require a lot more linguistic processing to be able to use them. Therefore, most IR researches and systems depend on words in their categorization and retrieval techniques. Basically they count on word frequency information [5].

1.4 Applications of Information Retrieval and Research Areas:

IR systems do not achieve high quality effective retrieval because of one or more of three main reasons. First, the low quality of automatic understanding of a user need. Second, a low quality of automatic understanding of documents' text. Third, a low quality of matching between the user needs and documents' contents [21]. Researchers in IR field are trying to improve the quality of IR systems by improving quality in these three directions in many areas. Moreover, many applications for IR field are waiting on these improvements to start building new systems or improve existing ones to make use of these advanced researches in this very important and hot field.

1.4.1 Research Areas and Applications:

There are many research areas in IR field. The core IR field is concerned with retrieving all documents in a collection that are related to a certain topic or query. Many research areas originated from this main scope. For example, finding answers to a closed set of questions is the focus of question answering research area. Following is a brief of some of the most important research areas in the IR field and some of their applications. They do not cover all areas, but only what we think as the hottest ones.

In the following section we'll describe briefly other important applications of the IR field that were not discussed in this section.

1.4.1.1 Integrated Solutions

A very important application for text-based IR tools is using them as a part of an integration solution that integrates other applications together. This helps a lot in solving organization's information management problems. Other text-based tools such as routing and extraction, multimedia tools, and scanned documents tools combined with text-based IR tools provide a big part of the integration solution to be used by organizations. It became one of the most important aspects of developing a common platform to integrate Database management and IR systems. Many applications today can make use of an effective integration between these two systems together with other multimedia capability [35].

1.4.1.2 Distributed Information Retrieval.

The demand for text retrieval systems that can work in distributed, wide area network environment has greatly increased since the World Wide Web evolved and a huge increase in the use of the Internet has become evident. The amount of information available on the net is huge and it is not possible to store everything in one Database system. Other applications such as Lotus Notes facilitated a rapid creation tool for a Database distributed throughout an organization [35]. Such advances dictated a new requirement to the IR field. The research area works on this issue is known as Distributed IR.

A distributed IR system would typically work on hundreds of thousands of Databases. The first problem it would face would be selecting the databases in which it will perform the search process. The second problem will be gathering all the retrieved documents and ranking them to generate the result set for the user. It can not depend on the ranking provided by each Database, because each Database might have its special ranking scale, and because the user is expecting a uniform output from the system. All internal processing should be hidden to the user as if he is getting his answers from only one Database [35].

1.4.1.3 Efficient and Flexible Indexing and Retrieval

Many different aspects of the system may have impact on its efficiency. Almost all users interested in IR field are concerned with efficiency measures such as query response time and indexing speed. Since full-text Databases are being used

now, efficiency became more important than ever. New algorithms are designed to increase indexing and query processing speeds regularly. Other techniques such as text compression are used to decrease the storage overheads an I/O times. Of course real-time text-based multi-user systems are very concerned about this issue; moreover, they are concerned with concurrency control, update, and recovery strategies [35].

1.4.1.4 Magic

Magic is the way users would regard an IR system that retrieves documents containing information about information needed by the user but not containing words used by the user in his query to describe his needs. Information needs are usually described using different vocabulary. The words in the query might not exist in the documents that contain the needed information. The process of query expansion by an IR system provides a solution for this problem. Vocabulary expansion can result from transforming the document and query representation, as with Latent Semantic Indexing technique. Another technique being used is using automatic thesaurus built by corpus analysis [35].

1.4.1.5 Interfaces and Browsing

The system interface is the major and maybe the only way a user can interact with the system. Therefore, it is one of the most important factors involved in evaluating a system. IR systems retrieval and routing algorithms became more complex; hence, an IR system interface became more complex than before to be able to provide the system with the needed requirements and then report the result to the user. The interface must be easy to deal with and user friendly. It must support many functions in an easy way such as query formulation, presentation of retrieved information, feedback, and browsing [35]. There has not been much effort done in interfaces of IR systems, but lately many people are interested in this area and produce some work and more work is yet to come.

1.4.1.6 Routing and Filtering

Information routing is also known as information filtering, or information clipping. These are synonyms describing the same process of identifying information requirements stored by users in profiles rather than queries and sending the identified document to the interested users. A typical system that uses routing techniques would have many users and one or more profiles for each user. A profile is a long-term more complex expression than a query, but still it is a way of describing a specific information need by the user [35]. These systems are usually dynamic and new data arrive to them regularly. A typical system using routing techniques will sort and index arriving information and will compare them with users' profiles and will send the matching information to users. This is typically used to retrieve information from news feeds. Effectiveness and efficiency are very important in routing systems and they are measured in different ways than IR systems.

1.4.1.7 Effective Retrieval

Effective retrieval techniques have been the core of IR research for many years. Many effectiveness measures have been proposed, but the most important and most widely used ones are precision and recall.

Users are always interested in techniques that achieve significant improvements and avoid occasional major mistakes rather than only achieving few percentages of improvements in precision and recall. The user regards a technique that results in reliable few improvements as a better one than another technique that is not reliable and achieves higher improvements. A known technique for its contribution in improvements, but it is a main source for bad mistakes is stemming. New techniques are being developed to improve stemming and others are trying to integrate stemming in other stages such as in query processing rather than the document indexing stage [35].

1.4.1.8 Multimedia Retrieval

This area of research is interested in retrieving information from video, sound, and image Databases without text descriptions. The benefits perceived from this field are very interesting to industry. However, there are not many general-purpose solutions for that. Solutions for this field may include retrieval of photographs of

faces, or generating pictures of fabric in specific color shades [35]. Multimedia indexing and retrieval has totally different techniques than text-based indexing and retrieval, because it deals with different formats. However, there are retrieval models that are applicable on all formats, so they are applied in multimedia retrieval as well as retrieval of text-based information.

1.4.1.9 Information Extraction

According to Croft [35], information extraction was primarily developed in the ARPA (Advanced Research Projects Agency) MUCs Message Understanding Conferences. This field is of main interest to the government as well as companies. Systems built in this area extract certain types of information from information Databases or data streams such as news feeds. For example, if you are interested in extracting information about football players moving from a team to another, the system will extract the player name, the names of two teams involved in the deal, the date of movement, and the price paid from the buying team to the selling one.

These systems are of great importance to government agencies. They are also of great importance to companies and organizations as well as individuals. For general use, they are concerned with extracting places' names, people names, organization names, dates, and numbers. Research invested in this area is more focused on reducing the efforts needed to develop new applications rather than improving the existing ones. This is due to the current state of the existing applications that require considerable investments to modify or build them again [35].

1.4.1.10 Relevance Feedback

Government agencies and industry see relevance feedback as a useful feature for IR systems. This technique retrieves a set of documents matching the user query and the user would select those documents that are most relevant to his query. The system would then use the user's feedback to generate a new more advanced and powerful query and enhance its retrieval.

Relevance feedback selects the best features (words or phrases) from the relevant documents, gives them weights and uses them again in a new query fed into the system for another search iteration. Some practical difficulties delayed the adoption of this technique in many IR systems. The most important difficulty is that

most research and testing done in this area is based on small size test collections of abstract-length document. Another problem is that we can not anticipate the behavior of users when requested to give their feedback. The main idea here is to consider user feedback by selecting the most relevant set of documents to be able to enhance the query. Users may select only one document that might not even be strongly relevant. Of course this would result in less effective retrieval. These factors made this technique an unpredictable one for use in practical life [35]. More research is being done to improve this technique such as automatic relevance feedback.

1.4.1.11 Text Summarization

This area of research can be thought of as producing a table of contents to a book. A table of content of a book makes the user browse the contents of the book through a logical view of much less size than the full-text view. A typical document will not include a table of contents; therefore, text summarization builds this table of contents for each document in a collection [5].

Text summarization is not interested in identifying or learning what a document is mainly talking about or something specific about the document as much as it is concerned in generating a smaller scale logical view of the document that describes the contents of the document.

A contribution in this area [5] is using cluster analysis to group documents and describe clusters. A user has the option to select a subset of clusters for further evaluation and new representation is built according to the selected set of clusters.

1.4.1.12 Text Categorization

This area of research is concerned with labeling documents of a collection with labels that would be used during search as keywords for fast retrieval. This is a very useful technique for organizing large collections of documents. It was usually done manually by humans assigning these labels to documents, but an automatic text classifier will save a lot of time and money.

If we think of a single query as a topic or category and an IR system is retrieving all relevant documents to that category, we'll see that the whole IR problem is actually a Text Categorization (TC) problem. Since these two are very similar, most techniques used for Text Categorization were helpful in solving the basic classical IR

problem. The techniques used to solve this problem are mainly based on machine learning and statistical categorization [5].

TC area of research is the main focus of the present work. In the remaining chapters of this thesis we will be explaining IR in general since TC is considered an IR that retrieves relevant documents to multiple queries. Therefore, we find it useful to understand techniques used for both.

1.4.1.13 Search Engines

One of the most widely used applications that make use of Information Retrieval research is the development of search engines. Millions use search engines on the Internet every day. A search engine might use a TC model to construct its index and a retrieval model to retrieve the result set to the user. Text operations are applied on both documents' text and user information requests. TC model is applied to categorize documents and construct the index. Query operations might be applied to the user request later on to prepare the query for the search and rank operation. The engine will use a retrieval model to match the user query with its index and retrieve the result set that better match the query. A ranking process ranks the retrieved documents according to the degree of relevance to the query.

1.4.2 Other Information Retrieval Applications:

There are many other applications for Information Retrieval. We only explained the most important ones and the hottest in research area. This sub-section briefly mentions other important areas of research and applications. These were not described in more details in this introduction because they do not add much value to this introduction focused at providing a good basis for IR field and text document categorization sub-field, which is the main focus of this thesis.

These applications include data and information mining, multi-lingual IR also known as cross-language retrieval, question answering systems, recommendation systems, automatic organization, and knowledge management.

1.5 Motivation, Thesis Objectives and Document Organization:

We have seen in this chapter the importance of Information retrieval field to government agencies, companies and organizations, as well as to individuals. For a large collection of data such as that available in a library many tasks need to be done in a better way than before since the amount of data has been multiplied several times. Tasks such as library catalogue and general administration have been successfully automated; however, having an effective easy to use Information Retrieval system has unfortunately not been achieved [3]. This example applies in many fields where effective IR systems are needed. This demand has created the need for an effective IR system and encouraged many researchers to work in many directions in this field. Some of them are interested in other areas such as efficiency and user interface of an IR system, but most of the researchers were more interested in the effectiveness of the categorization and retrieval processes. Therefore, categorization effectiveness became our main interest and this thesis proposes techniques to improve categorization precision for text documents.

The main objective of this thesis is to propose and investigate a methodology for text categorization based on phrase indexing techniques using a statistical approach that achieves high precision. Not many researchers investigated text categorization based on phrases and most researches have resulted in similar conclusions that phrases can be used to enhance the performance, but not as a primary categorization tool. These results discouraged some researchers from pursuing further research on phrases, but encouraged others to investigate phrases more thoroughly. We believe that since a phrase carries more meaning than a term, it should be a better classifier than a term if employed in an efficient way. We will use a subset of the Reuters-21578 test collection. It is a standard collection for text categorization research [20]. We will propose a feature reduction technique based on threshold concepts to filter out low-quality terms and phrases and construct reasonable size index tables. We will also propose and investigate four categorization techniques, the most important of which and the basic focus of this thesis is based on phrases. Other techniques will be based on terms and are proposed to compare their performance to the phrase-based classifier. We will also study the combination of several techniques together and assess the resulting effectiveness. Out first goal is to prove that phrase-

based categorization can achieve high precision. The second goal is to suggest other techniques that might be used to enhance effectiveness of the primary classifier based on phrases and at the same time achieve better performance than the most widely used term classifiers with less computational complexity.

This thesis consists of two parts. The first part (chapters 1 and 2) provided in this chapter an introduction to the IR and TC fields and their research areas and related applications. Chapter (2) provides a detailed survey for Information Retrieval models and methods used for Categorization and feature reduction techniques.

The second part of the thesis (chapters 3, 4, and 5) describes our proposed methodology in detail. In chapter (3) we describe our categorization methodology in abstract level and explain how phrases are used for categorization. Chapter (4) explains the pre-processing techniques used for the categorization process. It also explains how we propose to reduce the number of index phrases and index terms used for categorization and how the system learns information that will help it classify documents. Chapter (5) explains in details our proposed categorization techniques and analyzes results achieved by individual techniques. We also study the combination of different techniques together and the effect of this combination to the categorization effectiveness precisely in terms of categorization precision. We also compare our results with results of other researches. The last chapter (Chapter 6) concludes methodology and achieved results and suggests future work building on our results.

Chapter 2

Literature Survey

must be fed into another stage to decide on relevance. The first problem is considered to be much easier than the second one.

In general, in the 1960's retrieval and indexing techniques have witnessed some basic advances. Followed by new models in the 1970's for probabilistic and vector space models. Also clustering, relevance feedback, and large on-line boolean information services were presented in the same decade. Natural Language Processing (NLP) and IR started on the 1980's and started to be used in Expert systems. Off-the-shelf IR systems started also in this period. While in 1990's and 2000's many serious changes and enhancements occurred in this direction. Large scale, full text IR and filtering experiments were implemented and systems have been built. Dominance of ranking, interfaces and browsing, multimedia, multilingual, and many web-based search engines were developed and used. Machine learning techniques and question answering techniques were implemented [15].

As we mentioned before, TC is very similar to IR, but categorizing documents to more than one predefined category or topic instead of a single user query. Research has been conducted on IR more than TC. Most techniques and models used are applicable for both fields. In this chapter we will survey IR and TC together because of the great similarity between both. We believe that we can make use of IR techniques in TC research.

In the following sections, we'll describe traditional methods and early developments in the field of text and information retrieval then move forward into more advanced techniques. The reasons for describing the traditional methods are to cover almost all origins of the current techniques, gain knowledge of these methods as background information for newer developments. They also include learning the basis of almost all-newer developments that provided extensions and/or variations to these basic methods to become what they are [21].

2.2 Early Developments:

Developments in the field of IR go back to thousands of years ago. A table of contents or an index is a simple early example of the early developments in IR. Manual indexes used to be constructed to provide easy access to contents of information items [23]. Libraries as main sources of information have been among the first users of manual then automatic IR tools to help users easily access their

information needs. They started by manual catalogue cards passing by more advances searching tools by subject, keyword, etc. and reaching the today's tools of graphical interfaces, electronic forms, etc. [23]. With the advances of computer and digital technology, automatic indexing techniques have been developed to provide indexes for large amounts of data collections. We are only concerned in this thesis in computer-related techniques and models.

2.2.1 Full Text Scanning

This approach is the most forward keyword matching technique. It mainly compares search expression-string(s) with the document by scanning all document text trying to find a match. If a match is found, then this document is included in the result set. If the search expression consists of several strings, a sub-test will be conducted for each string and the final result is a Boolean expression of the combined sub-tests results. The search time for this technique is linear to the document length, while the number of states of the automation may be exponential to the size of the regular search expression [2].

The basic technique compares the string with the document by scanning the document from the beginning to the end searching for a match for the first character of the search string, followed by comparing the rest of the characters. If any character mismatch occurs, it goes back to the first matching character and starts the matching process again from the next character. Variations, enhancements and more efficient techniques are developed but mostly based on search expressions of string nature. This technique is very easy to implement, but also very slow [2].

Better techniques were developed to enhance the processing speed for full-text search methods. A better technique uses a preprocessing part that detects the recurring sequences of letters so that when a mismatch occurs, a shift of more than one character is applied saving a significant number of comparisons. Another fast technique applies the comparison of the letters from right to left. If a mismatch occurs, it shifts with the whole number of the search string. Other techniques used automata theory or bit-encoding schemes for simultaneous multi-search algorithm [2].

The main advantage of the full-text search method is that it needs no storage overhead. Minimal effort for insertions and updates is another advantage. On the other hand, bad response time is the main disadvantage [2].

2.2.2 Signature Files

Signature files is a very interesting approach and attracted many researchers. It uses hashing on the document words and superimposed coding to produce a smaller size signature file for the document. This file is a bit string or signature of the original file. Comparing this smaller file is much easier and faster than comparing the original long document. One approach applied a stage of stop list removal and word stemming to reduce the amount of words processed and used a numeric procedure instead of the look-up table as a hashing function [2].

Other techniques suggested using consecutive letters "n-grams" as input to the hashing function, or using equi-frequent text segments instead of the n-grams. Another suggested not to use superimposed coding and having the signature file as the concatenation of all words' signatures [2].

Another interesting technique uses one-level signature file. It has interesting details, two of which may be used for text retrieval. It stored all first bits of all signatures consecutively first, then the second bits, and so on. This technique reduced the I/O time of retrieval. He also suggested that more frequently appearing words should be treated in a special way, and that of course affects the creation of the signature file [2].

Two-level signature files, trees of signatures, and partitioning based on signatures were among many techniques developed to improve search speed. A mechanical device based on edge-notched cards and needles attracted many interests. Techniques developed based on signature files and the edge-notched machine to minimize the false-drops probability [2].

The main advantages of this method are its easiness of implementation, efficiency in handling insertions, the ability to handle queries on parts of words, the tolerance to handle spelling and typing mistakes, the ability to handle growing files, and can be easily transformed for parallel processing. On the other hand, the main disadvantage is the bad response time when the file is large [2].

2.2.3 Inversion

This method is used by most commercial packages for its fast retrieval and easy implementation. It constructs an index file for all keywords of all documents. For each keyword in the table, it keeps pointers to the documents referring to this keyword. Other techniques used two-level index tables where they stored words sharing the first letters in the fist level and pointers to the second level where these words are separated and each of them holds pointers to containing documents.

Developments and challenges based on this method include the proposed hybrid methods and algorithms to grow the posting list adaptively since few of the words in the vocabulary of the posting list appear a lot, while most of the words appear only once or twice. Since the index tables might be huge, other techniques were developed to achieve fast insertions incrementally. Other compression techniques are proposed to handle the huge index sizes [2].

The main advantages of Inversion are the relative easiness of implementation, the speed, and the high elasticity in handling synonyms. On the other hand, the index huge size, index updating or re-organizing time and the cost of merging lists in a dynamic environment are the main disadvantages [2].

2.3 Information Retrieval Processes:

Information Retrieval is composed of two processes. The first is the information indexing process, while the second is the information retrieval process which is composed of two sub-processes: the user needs representation sub-process, and the information-needs matching sub-process. This is shown and explained in the following figure by Yates and Neto [23] (Fig. 2), It is necessary to define the information Database before the retrieval process can be initiated. This first process starts by feeding the system with information or text documents, text operations that will be applied, and indexing model. The system applies text operations and indexing techniques to generate an index for the information as shown in the left half of the diagram. The second process, which is the retrieval process as shown in the right half of the diagram, starts by the user feeding his information need to the system. Usually this information need is represented by a kind of query of text strings using a visual interface. The system applies the same text operations to the query as it did with the

documents. The system might apply query operations to generate another query form fed into the retrieval engine. The system then matches the query with the index using the retrieval engine and retrieves the matched documents. It performs a ranking stage for the retrieved documents to sort the result set to the user with a relevance scale. The ranked result set is then displayed to the user through the visual interface. The user might select a subset of the result set and indicate that it is relevant to his information needs and start another feedback-initiated retrieval process. The second retrieval process quality depends on the quality of feedback given to the system represented by the selected subset of relevant documents. The system then uses the feedback to re-formulate the old query to a new one that represents the user needs in a better way and feed it to its retrieval engine initiating a new search and match cycle.

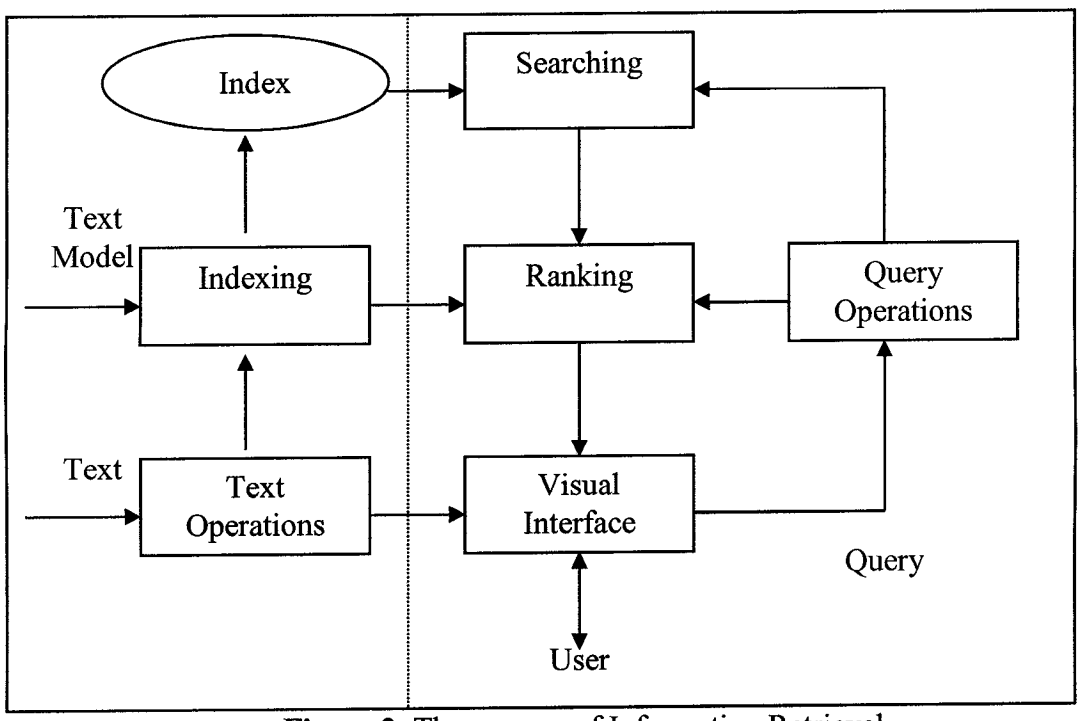

Figure 2: The process of Information Retrieval

This is a typical process description for Information Retrieval. Techniques, operations, methods, and models may vary, but the core process remains as explained above. The quality of the retrieved set is dependent on many factors. It depends on the text operations applied, indexing models used, index formulation quality, information need representation, query operations, search and match engine techniques, and the ranking method applied. If relevance feedback is used, it adds another factor, which is the quality of feedback from the user that is taken as input for next retrieval processes.

2.4 Information Retrieval Evaluation:

Evaluation of an Information Retrieval system has been always a complex area in IR. Many factors are considered in the evaluation process of an IR system but the most important are effectiveness and efficiency of a system. Effectiveness has many measures, but the most widely used are precision and recall. Efficiency is measured in many ways such as response time and storage overhead needed by the system.

Other evaluation factors include the ability of the system to assist the user in formulating his query, the presentation of the documents in the result set, and the interface appeal to users.

Since the most important evaluation factor is the system effectiveness, we will only focus on this factor in details. The following sub-sections will describe test collections used for IR and TC systems evaluations, the effectiveness evaluation performance measures, and briefly describe other evaluation methods for system effectiveness. Effectiveness for IR and TC measures the ability of the system to retrieve the largest number of relevant documents and the least number of non-relevant documents.

2.4.1 Test Collections

A Test collection, as mentioned before, is a group of text documents, a group of queries, and a list that links relevant document to these queries also called as relevance judgment. Collections and relevance judgments are very important in training and testing IR systems. They provide a standard benchmark so we can compare systems to each other and to the optimal results. An effective IR system should either select all documents relevant to any query or rank them as more relevant than documents that are less relevant and use threshold to finally obtain the result set.

Many test collections have been developed for the purpose of testing IR systems [5]. These collections vary in the number of documents they contain, the style and contents of the documents, length and specificity of the query, number of relevant documents, and documents' length. Each of these variations among collections affect very much the techniques used for classification and retrieval. The ultimate goal of IR

and TC research is to have a system that behaves well in all environments. Therefore, until we find a system that is reliable and offers guaranteed effectiveness in response to different test collection, research will be always an open playground for researchers to develop and apply new techniques and models.

The problem with test collections is that they represent only a subset of what a user might need and represent in his query. The documents in the collections are also a small subset of what is there in the real world. Researchers are in debate of how good a system will be in real life even if it has been proven to be good with test collections [5]. However, having a standard benchmark to measure systems against each other and against a known result is definitely a great advantage of having test collections.

A part of this problem is automatically handled if the set of documents and queries in the collections represent an unbiased sample of real life data and queries. The assessment of relevance is very much related to individual users as indicated by Salton [14] and referenced by Hull [5]. Therefore, any retrieved set might be judged with a different relevance score from user to another. This problem has no obvious good solution.

Examples of these test collections include Reuters-21578, Lisa, NPL, MED, and CRAN collections. A huge collection of two gigabytes of documents was also developed by Text REtrieval Conference (TREC) [30] to be used in the research area of IR. It contains a diversity of document sizes, topics, and styles to provide a better testing environment for many researchers.

2.4.2 Text REtrieval Conference (TREC)

The Text Retrieval Conference (TREC) [30] was established in 1991 to evaluate large-scale IR systems and models. There are many tracks of research that are being evaluated in IR by TREC. This conference is considered to be the most well known IR evaluation setting. It provides information, work done by researchers, results, and test collections.

2.4.3 Effectiveness Performance Measures

The output of an IR or TC process is a set of documents ranked according to their relevance. However, for most test collections, documents are defined to be either relevant or irrelevant. This ranking is important because IR or TC process is not a definite process due its complexity, so documents with higher-ranking scores are most probably more relevant than others with lower ranking scores to the information need or pre-defined topic.

2.4.3.1 Precision and Recall

Precision is known as the proportion of retrieved document that is relevant to the total number of retrieved documents.

$$\text{Pr}\,ecision = \frac{\text{Re}\,trieved\ \text{Re}\,levant}{Total\ \text{Re}\,trieved} = \frac{r}{n} \qquad (Eq.2.1)$$

Where r = the number of retrieved documents that are relevant and n is the total number of retrieved documents.

Recall is the proportion of retrieved document that is relevant to the total number of relevant documents.

$$\text{Re}\,call = \frac{\text{Re}\,trieved\,\text{Re}\,levant}{Total\,\text{Re}\,levant} = \frac{r}{R} \qquad (Eq.2.2)$$

Where r = the number of retrieved documents that are relevant and R is the total number of relevant documents.

Both precision and recall are usually normalized to take values between zero and one or in the form of percentage. This helps in obtaining uniform values in different testing environments or at different DCV (Document Cut-off Values). DCV is defined as the number of documents assumed that the user would like to examine out of the result set.

Precision and Recall are known to have an inverse relation with each other. As the number of examined documents increase, recall values increase and precision values tend to decrease [5].

2.4.3.2 Averaging Methods

We can draw an evaluation curve for each query. If we average all these curves we can obtain an overall evaluation of the system. If we plot precision or recall against DCV values, we can take the averages easily at the same DCV for each query. However if we plot precision vs. recall, it won't be that easy to average values since recall values might differ from query to another depending on the number of relevant documents. The Macro-evaluation strategy takes the average precision at the same recall value for all queries and repeat that for different fixed levels of recall values to obtain an average curve of precision at a single scale of fixed levels of recall values [5].

2.4.3.3 Evaluation Assumptions

A standard technique for comparing scores to tell when a method is better than the other is fixing one variable and calculating the score of the other using the different methods. Usually the fixed variable is called the control or the independent variable, while the other variable is called the response or the dependant variable. The Macro-evaluation technique uses recall as the control variable and precision as the response variable. While averaging at fixed DCV uses the number of documents examined as the control variable and precision or recall as the response variables [5].

2.4.4 Other Evaluation Methods

Many evaluation methods have been proposed other than precision and recall but are not widely used in evaluations of systems; however, all the other evaluation techniques are good to use. One can choose whatever evaluation technique he wishes to use to evaluate his method. These other techniques include modeling document scores by two normal distributions, one for relevant and one for irrelevant documents. Another evaluation technique is called Expected Search Length (ESL). This technique measures the expected number of documents that must be examined to satisfy the user's needs [5].

On the other hand, many researchers are not satisfied by the current evaluation techniques. Mainly because they consider all relevant documents of equal importance

which is not always the real case. Also because two relevant documents are assumed to contain twice as much information as a single relevant document.

2.5 Lexical Analysis and Pre-Processing Steps:

Most documents are structured to sections such as an introduction, a body, and a conclusion. This structure is important for the reader to better understand the document content in less time. Each of these sections will contain words of different natures: verbs, nouns, adjectives, articles, etc. Not all of these words are of significant meaning by itself, but combined with other words in a context they make sense and compose a building block in the total meaning of the document. The more words are combined together, the better we understand the document contents. Of all the words in the document, we will find some words of greater significant to the topic the document is talking about. On the other hand, other words such as articles are of no significance at all to the document by themselves, but in the context they contribute to the collective meaning of the other words.

Researchers analyzed this natural process and are trying to develop methods and algorithms to make a computer better understand the document in the same way. Therefore, researches have been done in this area and resulted into a set of methods that enable an IR or TC system of analyzing the document in the way mentioned above and obtain a set of raw data to be processed later by the categorization or retrieval model. These lexical analysis methods are considered pre-processing steps for an IR or TC system [5][15]. In this section we will explain these methods and describe the advantages and disadvantages of each.

2.5.1 Parsing:

Parsing recognizes the structure of the document. A parser would read the document from tip to toe trying to recognize major sections of the document, such as title, introduction, chapters, conclusion, references, etc. This method is very useful and we think that applying different techniques to different sections might be useful. On the other hand, recognizing the structure of an unstructured document is not an easy task.

2.5.2 Token finding:

This process is also called word extraction. A token is simply a stream of characters that has a collective meaning [15]. Token finding attempts to recognize tokens in the document such as numbers, alphanumeric words, spacing, punctuation, and tokens of special meaning. This step is important because remaining words in the document will be only alphabetical words, thus ready for further analysis applicable only on alphabetical strings. At the same time numbers and alphanumeric words can be handled by other techniques since they are recognized in this step. For some languages this is easy because there is a distinct separator between words. Other languages such as Chinese need words' segmentation first [5].

2.5.3 Stop Words Removal:

There are many words that are repeated many times in any document and in all documents. These words by themselves when taken out of their context do not bear a meaning. These words when used for indexing increase the index size dramatically. Therefore their removal does not affect the meaning of the document and decrease the index size dramatically, thus decreasing the search time as well. Most of these words are only used to represent relations between the other content-bearing words in the document. We call these words "stop-words". Examples of stop words are "in", "a", and "the". Stop-words list may vary from an algorithm to another. They are also context-dependant. A certain word for some context might be considered as a stop word, while in another context it will be important as an index term. In general stop-words lists range between 8 and 500 words [5][15]. A sample list of these words is available in appendix A of this thesis. This list is used for our research purpose.

Therefore, a stop word removal step removes all these non-content-bearing words from the text stream preparing for indexing. A disadvantage of this step is that it loses the relations between the words, so the complete meaning of the text might be unclear. Another problem is falling into the trap of removing content-bearing information such as removing "a" from "Vitamin A". A good solution for the first problem is to extract the relations before removing stop-words. The second problem is more complex and researchers are trying to find good solutions for it.

2.5.4 Stemming:

Different morphological variants of the same word might exist in the same document or in other documents. For efficient indexing as well as effective categorization, we need to treat all these variants as one word. The process of stemming obtains the root form for all variants of the same word, also called "stem". A stemmer may consist of a set of rules and / or dictionaries. These rules remove or add prefixes or suffixes to the word variant to reach the word root [15].

The simplest stemmer is the plural to singular stemmer, It removes the "s" suffix from plural words to obtain the singular form of the word. Most stemmers have been developed using a set of rules that are applied in sequence to obtain the word root. Many experiments have been conducted to measure the effect of stemming to categorization process. The results show that using a stemmer will reduce the index size with 20 – 50% for small collections. Experiments also show that the effect tend to decrease with larger collections. However, all stemmers do not negatively affect the IR process performance or efficiency [5].

On the other hand, sometimes a certain word might be meaningful to the context with its prefix or suffix, while it will be of no meaning at all when stemmed. This might result into misleading indexing. This is the only problem with stemmers, but it has a low occurrence frequency, so most probably it is neglected because reduction in index size makes us ignore this low probable deficiency [5][15].

An example of known stemmers is Porter stemmer [22]. This algorithm is a set of rules that are applied in sequence to a word to obtain its root. It depends on the consonants and vowels sequences. The stemmer measures the variable m for this form $[C](VC)^m[V]$ where C is a sequence of consonants and V is a sequence of vowels including "y". Examples for m are (m=0) in "tree" and (m=1) in "trees" or "trouble" and (m=2) appears in "troubles" or "private". The algorithm applies a set of steps in sequence to transfer the word from its old form to its new form [15]. For example, the word "sensitivity" will be stemmed in sequence (sensitivity ➔ sensitiviti ➔ sensitive) to reach its root "sensitive". A complete description of Porter Stemmer is available in this thesis appendix B.

Many variations and enhancements have been developed to Porter stemmer. Other stemmers also have been developed to achieve the same goal such as KSTEM that combines a set of rules with a list of words [15]. Actually most stemmers are seen

as recall enhancement techniques. The most important consideration for a good stemmer is to achieve high recall value without deteriorating precision [34].

2.5.5 Weighting:

All previous steps only analyze the document to be ready for further stages. We can not yet measure how relevant a document is to a certain topic or query. Assigning weights to documents is the step that will make that measurement phase possible. In previously explained pre-processing steps we establish the base for this weighting process by assigning weights to single document features such as words or phrases. A document relevance weight most probably will depend on a combination of features' weights. Different weighting techniques have been developed to measure the relevance of a certain document to a certain topic or query. A document weight might be calculated using several weighting techniques. The term frequency and term inverse document frequency are two of the most important weighting techniques. In most weighting techniques certain normalization process is required to obtain uniform results across different documents and collections. In the following sub-sections we will describe some of the well-known techniques used in document weighting.

2.5.5.1 Term Weighting:

The first category of weighting for a document is term weighting. It includes calculation methods based on individual terms.

2.5.5.1.1 Term Frequency (TF):

This model considers the words that appear in a document more frequent than others are more important in describing the document and thus take more weights. This model has many ways to calculate the TF for any word in the document. Usually TF values need normalization that can be based on maximum term frequency or could include document length component. In large collections, logarithms are used to smooth the numbers obtained. In general for a term t(i) in document d(j) the term frequency is known to be [28]:

TF(i,j) = the number of occurrences of t(i) in d(j). *(Eq. 2.3)*

2.5.5.1.2 Term Inverse Document Frequency (IDF):

Also called Collection Frequency. It is inversely proportional to the number of documents in the collection in which the term appears. A term that appears in all documents is not of great significant to a certain document and vice versa. This measurement counts the number of documents in the collection in which a term occurs. IDF for term t(i) appeared in df(i) documents in total of N documents in the collection is [28][15]:

$$IDF = \log (N / df) + 1, \text{ or} \qquad \textit{(Eq. 2.4)}$$
$$IDF = \log N - \log df$$

Logarithmic function is used for normalization and can be used with any convenient base. One useful implementation uses base two to the power 0.1 to achieve resulting values in the range of –32K to 32K [28]

2.5.5.1.3 Term Weighting Normalization:

A term that occurs 5 times in a 50-terms document is apparent to be more of more significant than in a 5000-terms document. Therefore, one of the methods to detect such a fact and take it into consideration while weighting the document terms is document length normalization. The length of the document is calculated based on its number of terms and used to normalize terms' frequencies. For a document d(j) the document length is [28]:

DL (j) = the total number of all terms occurrences in the document

We can also normalize the measure by using the length of an average document, so the Normalized DL is [28]:

$$NDL (j) = DL (j) / \text{Average DL for all documents.} \qquad \textit{(Eq. 2.5)}$$

Another term normalization method is to use the maximum term frequency. For example in TC, the maximum term frequency for certain category will represent a

normalization factor for the term frequency in each document that would better describe how important this term to the matched category.

2.5.5.1.4 Combining Term Weighting Techniques:

Better results can be achieved by combining described techniques: normalized TF and IDF for all matched terms. This is known as the TF.IDF technique. For matching documents against queries or topics, this will give one score that will better describe the relevance. There are many variations for the combining formula. This formula is among the best ones that was proven by TREC to be highly effective [28]:

For one term t(i) and one document d(j)

$$Combined\,Weight\,(i,j) = \frac{[TF(i,j) * IDF(i) * (K1+1)]}{[K1 * ((1-b) + (b * (NDL(j)))) + TF(i,j)]} \qquad (Eq.2.6)$$

where K1 and b are tuning constants. K1 is used to modify the extent of influence of TF. K1 = 0 eliminates the influence while K1 = 2 is an appropriate value for heterogeneous collections of full text as tested by TREC. Larger values will increase the influence of TF. On the other hand b ranges from 0 to 1 and it modifies the effect of document length. If b = 1, documents are assumed long because of repetitiveness, while b = 0, documents are assumed long because they are multi-topic. TREC have found that b = 0.75 is helpful. This formula ensures that the effect of term frequency is not too strong. It also ensures that the combined weight for a term occurring once in an average length document is only the IDF value [28].

The Document total Weight will be simply the summation of all terms' combined weights. These terms might be only the query terms when matching with a query or the selected terms (or all terms) of the document for text categorization problems.

2.5.5.1.5 Term Weighting Techniques Problems:

Strzalkowski [31] indicated that standard TF.IDF weighting may be inappropriate for mixed term sets, consisting of ordinary concepts, proper names, and phrases. One of the reasons for that is that it favors terms that occur more frequently in a document which is good for general-type queries such as "find all you know

about topic X". Another reason is that it assigns low weights for less frequent, but highly specific terms such as people names, which is often decisive for relevance. The third reason is that it does not address the inter-term dependencies when a phrase and its constituent terms appear in the document. Techniques have been developed to solve these problems. The first two were tackled in TREC-2 and solutions have been proposed including special weighting for phrases and considering the top T highest IDF values to cover specific terms [31].

2.5.5.2 Phrase Weighting:

A more meaningful information item in the document will be a phrase. A phrase is group of 2 or more words that together have a collective meaning. Phrase indexing and weighting is more complex than term indexing and weighting. Many factors affect phrase indexing and weighting. One factor to be considered is the number of terms to be used to construct a phrase. Another factor is the maximum distance between terms that is allowed to consider these terms as a single phrase.

Phrase indexing and weighting is the core of this thesis. This thesis mainly focuses on phrase-based categorization; therefore it will be discussed in more details in a complete section in Chapter 3.

2.6 Information Retrieval Models:

Information Retrieval models describe how the computational process works to rank documents and information requests and how the comparison is performed. In doing so, a model uses some variables including queries, documents, terms, relevance judgments, users, information needs, etc. There are many models developed for this purpose. Some of them are based on the statistical data gathered in the pre-processing steps described before and some others are based on Natural Language Processing or Artificial Intelligence or combination of many ways. Some models process one search process and return the result set to the user, while others use this initial result set to obtain a feedback from the user on what is relevant and what is not and run another search iteration to enhance results. Other models increase the terms of the query by introducing more terms to increase matching probability [28]. In the following sub-sections we will describe some of the most important models with more focus on

statistical based models because we will depend on statistical analysis in our proposed categorization method.

2.6.1 Boolean Model:

It is also called Exact Match model. It uses precise query criteria to identify if a document matches or not. Boolean operators are used for query construction. Any document is examined and either matches the query or not. The result is a set of documents with no specific order. Most commercial IR systems rely on such strategy because it's fast and return a result set of high precision [2][5][15]. A known example is WESTLAW. It has been successful in serving professional and legal market since 1974 [15].

Although this model is widely used by commercial systems, it has some disadvantages. These disadvantages are summarized below [5][15]:

- Query formulation is very hard.
- Same vocabulary must be used for indexing and query.
- No ranking for document importance.
- No concept of importance.
- No control over the size of the result set.
- Search result is extremely sensitive to the choice of query terms.

Enhancements to the original model were developed to overcome these disadvantages. Enhancing the interface and using a graphical representation for results of set-operations between query parts show users the components of his query and the effect when using each part separately. The user becomes able to decide which terms are more important for his query than others. Also good interfaces used to help users in the query formulation process. Another enhancement uses a structured dictionary for index terms and the user can use that concept to limit or enlarge his query result. Adding a ranking for the retrieved documents also helps to sort the result in term of importance. An easy way to do that is to give a rank equal to the number of query terms found in each document. Thus higher rank will mean more importance. To overcome the problem of terms occurring by coincidence, proximity variable was introduced for query phrase terms [5].

2.6.2 Best Match Model:

Best Match model is considered a variation of the Exact Match model. It uses almost the same concepts, but using free text in query formulation and the result is ranked in descending order of relevance. It is generally more effective than Exact Match, but was not tested on large collections, so it is not clear if performance will scale with collection size. Similar to Exact match, it does not understand natural language.

WESTLAW supported Best Match in 1992. Query is formed using boolean and proximity operators are combined with restrictions, term expansion and truncation characters, and some document structure fields [2][15].

2.6.3 Vector Space Model:

Vector space model represents both documents and queries as vectors in a multi-dimensional space. A document vector is composed of elements representing query-terms in the document. This representation can be boolean telling if the term exists in the document or not, or the term frequency or other more complex calculations. The angle between document and query vectors or the inner product of them estimates how relevant is the document to the query.

$$Q(i) = (q_{i1}, q_{i2}, \ldots, q_{in}) \qquad \textit{(Eq. 2.7)}$$

$$D(j) = (d_{j1}, d_{j2}, \ldots, d_{jn}) \qquad \textit{(Eq. 2.8)}$$

Where q_{ik} is a value of term k in query Q (i) and document D (j) [5].

Vector Space Model (VSM) solves most problems of the Boolean Model. It accepts free text queries, provides weighting for terms, and generates result set ordered by similarity weights. However it does not deal with term dependence. It has proven better results than Boolean Model in many experiments. The model also deals with short and long queries similarly and with no extra complications. Having result set of ordered documents according to weights also makes it possible to control the size of the result set as per user preference [5][15].

A slight variation from the original VSM uses concepts instead of terms as the vector elements. Concepts are harder to identify, but they provide orthogonal vectors.

Terms are easier to determine but are not completely orthogonal. Having orthogonal or linearly independent vectors achieves more powerful classification and better performance [15].

Representing documents and query as vectors allow easy understanding for what is called relevance feedback. Relevance Feedback methodology will be discussed later in this chapter. By applying addition of the weighted vectors of relevant document and subtraction of the vectors of the irrelevant documents to the query vector and obtain a new query vector and apply the matching process again, effectiveness has been proven to be enhanced [2].

2.6.4 Clustering:

The basic idea about clustering is to group related items together in a cluster and treat all items in the cluster in a similar way. Clustering can work on documents or terms. We can group documents that are more probable to be relevant to a certain query together in one cluster. We can also group terms that co-occur in multiple documents in one cluster. Synonyms also are always grouped in clusters. This is very useful for automatic thesaurus generation and dimensionality reduction in IR and TC systems. This seems very promising, but Salton [2] states that term-grouping algorithms effectiveness are in doubt.

Clustering in general involves two processes: cluster generation and cluster search. Cluster generation is usually implemented by using VSM to represent each document by a vector in multi-dimensional space and manual or automatic indexing process is conducted. Usually stop word removal for common words, word stemming, and dictionary for synonyms are used before indexing to generate clusters [13]. Simple automatic indexing was proven by Salton [12] to behave at least as good as manual indexing as Faltoutsos and Oard said [2]. A t-dimensional vector represents each document where t is the number of permissible index terms or concepts. A term that is absent from the document is represented by 0 or negative value [38] while the existence of the term is represented by 1 or a positive number representing the term weight. Many weighting functions have been proposed to calculate the term weight such as TF, IDF, and Term specificity. The next step in cluster generation is critical. It should be efficient and theoretically sound. This classification step should be stable under growth, independent of the initial order of the documents, and small errors in

document description should lead to only small changes in partitioning [3]. Clusters hierarchy generation on the other hand is an easy task that usually accelerates retrieval. Methods for clusters' hierarchy generation have been proposed including using document similarity matrix, nearest neighbor criteria, and minimum spanning tree [2].

The second process of clustering is searching in the cluster. Cluster searching is much easier than cluster generation. Query vector is compared to vectors representing cluster centroids. A cluster-to-query similarity function has to be used to decide which clusters are more similar to the query. These more similar clusters are processed first. A commonly used function is the cosine function [13]. Other method by Yu and Luk [2] used binary cluster vectors to calculate the expected number of qualifying documents in each cluster and suggest continuing in clusters where satisfying number of qualifying documents exist. Pattern recognition methods are also applied to derive a linear discriminating function to work as cluster-to-query similarity function [37] [2].

2.6.5 Relevance Feedback:

Relevance feedback as mentioned before is simply applying the search and match process again with new input from the user describing the initial result set components relevance to the initial query to obtain a new query for the new search and match process [13]. In general Relevance Feedback systems accept a query from the user and process the initial search and return to the user a set of documents that qualify as relevant to the user query. Usually these documents are sorted as the system thinks of their relevance to the user need represented by the query. The user examines the top relevant documents and identifies for the system if they are relevant or not. The system uses the user's judgement to enhance the query and search again in the collection. The process iterates as many as the user wishes to refine the result set as much as possible. This approach is known to achieve good effectiveness and also known for its easiness of use for the users since all what a user would do is to select the relevant documents and apply the search again until he is satisfied. Since it is an interactive process, the system speedy response is important. It should return the result set quickly to the user [5].

As previously mentioned VSM is ideal for relevance feedback application. Rochio [13] has developed one of the most successful strategies for relevance feedback based on VSM. His strategy improves performance by 20 – 80 % depending on the collection. He added a weighted-sum of relevant document and subtracted a weighted-sum of the irrelevant document from the query. The formula he used is [5]:

$$Q_{rfb} = Q_o + \alpha \sum_{i \in rel} \frac{D_i}{|D_i|} - \beta \sum_{j \notin rel} \frac{D_j}{|D_j|} \qquad (Eq.2.9)$$

Values for the formula coefficients and the number of documents a user may examine before the next iteration are among the most important factors tuning the relevance feedback process performance. Moreover, it was found that selecting and adding the most important T terms by their weights to the query is more effective than using all the terms from relevant documents. This is because some terms in the documents may not be of good relevance to the topic of the query, thus resulting in the danger of over-fitting [5].

Relevance feedback can be also applied using a Probabilistic Retrieval Model instead of the VSM. Probabilistic Model has two classes of documents for each query: relevant class and irrelevant class. An extension to this model added a "don't care" class as well. Given a document D it measures the probability of having this document belongs to the relevant class. If the probability is higher than the probability of not being relevant, it considers this document relevant else it is irrelevant [15]. Probabilistic Model like VSM does not consider the dependence structure that is known to exist between terms. It is based on using binary model of word occurrence and it is not clear if it will generalize well if word frequency information is included. Experiments have been made using Poisson model but results were not better than the basic VSM [5].

Combining Clustering with Relevance Feedback seems like a good approach. If initial query results in documents in more than one cluster and the user relevance feedback selects the relevant documents from one cluster, this makes it easier for the next iteration to retrieve documents from the more relevant cluster to the users query.

2.6.6 Semantic Information Models:

A different set of approaches for IR is to use the semantic information. These approaches uses information used by other models described above and add to them new semantic information extracted from the documents. The other models achieve good effectiveness, but these new models attempt to reach better performance. There are mainly three classes that use semantic information and they are described in the following sub-sections [2].

1. Models using parsing, syntactic information and natural language processing.
2. Latent Semantic Indexing (LSI).
3. Neural Network Models.

2.6.6.1 Natural Language Processing (NLP):

Enhancing IR systems performance is the goal of NLP. They match semantic contents of documents to queries. NLP techniques achieves good performance levels in TREC, but not as it was claimed and expected for them. In all cases and techniques of NLP, the first step is automatic syntactic analysis. At the same time,. Techniques such as stop-word list removal and phrase indexing are considered as a kind of NLP techniques. Stop list removal removes low-content-bearing words while phrase indexing deals with phrases which have greater semantic content than terms. Phrase indexing can be done on many levels. Croft et Al. [36] suggested using a coarse parser to detect sentences and using these sentences for indexing instead of terms. Other researchers suggested grouping keywords to achieve higher precision/recall values. Others used a skimming parser to turn documents into case frames instead of simple keyword systems. Others used document vectors for comparisons followed by section vectors, then paragraph then sentence vectors [2].

2.6.6.2 Latent Semantic Indexing (LSI):

LSI is an enhancement for VSM. LSI constructs a term-document matrix where each value V_{ij} represents the number of occurrences of the term i in document

j. The model then computes the Singular Value Decomposition SVD to eliminate the small singular values. The result is a singular vector and singular value matrices. They are used to map term frequency vectors for documents and queries into a sub-space where the semantic relationship from the term-document matrix exist and term usage variation is suppressed. The next step is to rank documents in order of their decreasing similarity to query. In order to do that the model uses the inner product of the vectors to calculate the cosine similarity measure [2]. Since LSI is used to reduce the number of terms taken into consideration in matching, it is considered as a dimensionality reduction technique as well. More details for LSI will be discussed in the coming section.

2.6.6.3 Neural Networks Models:

In general neural network techniques use spreading activation method. The common technique is to use an automatic or manual thesaurus to create one node in a hidden layer corresponding to each concept in the thesaurus. Many research and experiments have been done in this track and reasonable performances have been achieved [2]. Another approach was proposed by Mandl [32] to apply LSI using a neural back-propagation network. Using LSI as a preprocessing stage for the neural network enables the transformation between two representation schemes.

2.7 Feature Selection and Dimension Reduction:

Features could be terms or phrases or other variables that an IR or TC system would use to formulate its classification rules. One may also think of features as the individual terms, which are resulted from a full-text scan process. Feature selection might be selecting the more important terms reducing the size of the features instead of using all terms in the documents full-text logical view, or it might be selecting other features than terms.

2.7.1 Term Selection Techniques:

Many approaches have been used to select feature terms. We have previously described the Document Frequency (DF) in the explanation of Inverse Document

Frequency (IDF) technique as a preprocessing step. It is also used to limit the number of terms (features) to be selected for classification. In this approach rare terms are assumed to have little influence on the classification process. They are also thought of as being non-informative for category prediction. Therefore, terms with low DF values are ignored, thus reducing the size of the vocabulary and enhancing performance [39].

Another approach is Information Gain (IG). IG measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. We compute the IG for each term in the training set and remove those terms of IG values less than a certain threshold. Mutual Information (MI) is another approach for feature selection and reduction of term variables. MI considers a two-way contingency table of a term t and a category c. The model also defines A as the number of times t and c co-occur, B as the number of times t occurred without c, and C as the number of times c occurred without t, and N as the total number of documents. MI is then calculated as

$$MI(t,c) = \log \frac{\Pr obability\ of\ co-occurrence\ of\ t\ and\ c}{\Pr obability\ of\ t\ \ *\ \ \Pr obability\ of\ c} \qquad (Eq.2.10)$$

Which is approximated by this equation

$$MI(t,c) = \log \frac{A*N}{(A+C)*(A+B)} \qquad (Eq.2.11)$$

If t and c are independent, MI (t, c) will be equal to zero. To select a term we combine all category specific scores of a term to calculate the average and maximum Mutual Information [39].

Other term feature selection techniques include $?^2$ statistics and Term Strength (TS) techniques. $?^2$ measures the lack of independence between t and c and can be compared to the $?^2$ distribution with one degree of freedom to judge extremeness. On the other hand, Term Strength estimates term importance based on how commonly a term is likely to appear in closely related document. Document similarities of the training set is calculated and two documents are considered similar if their similarity value is higher than a set threshold. TS is calculated based on the conditional

probability that a term appears in the second half of related documents given it first appeared in the first half [39].

Most IR and TC systems and models are based on term frequency information; however, term-based models suffer three major problems. There are many terms that are synonyms to each other, a single term might have several meanings, and many terms carry similar information. The problem of synonymy is not only for single terms that could be solved by using a thesaurus, but also for phrases. A certain phrase concept might be represented in a certain field by a phrase while in another field represented by a totally different phrase, such as "Cluster Analysis" in statistics community and "unsupervised learning" in machine learning. The problem of polysemy has been tackled by many researchers trying to resolve the term-ambiguity problem that can degrade IR and TC performance greatly because of their multiple possible meaning such as "suit" which can have several meaning as a noun or as a verb [5]. The following methods are used for different feature selection to replace or enhance the term-frequency feature.

2.7.2 Improving VSM:

In general term-frequency-based models are unclear if we should use more or less terms in the query for higher effectiveness. Using less terms might make the model unable to retrieve relevant information to the topic while increasing the terms might retrieve information that is irrelevant to the topic. Replacement of or enhancement to the quality of query results in better performance. This can be achieved by other variables (features) in the query instead of terms such as phrases or term clusters. The following sub-sections describe models that deal with this type of enhancement [5].

2.7.2.1 Query Expansion:

Query Expansion provides a solution for the synonymy problem. It fetches all other terms describing concept of query terms and adds them to the query to cover all possible documents talking about the same concept but described in different terms. One approach for that is to use cluster analysis. For each query term, add to query all other terms in the same cluster where the original term exists. Most of cluster analysis

experiments slightly improved performance for some collections. They mainly differ in how they compute term similarity. Early experiments measured similarity by finding co-occurrence of terms in the documents, while later experiments first cluster documents then measure occurrence for terms in single cluster documents. An alternative approach is to use a thesaurus to figure out similar terms for query terms. Other approaches used term-term similarity matrix to obtain and add terms of greater similarity to the entire query concept instead of individual terms. In general Query Expansion is a useful approach that enhances performance especially when relevance information is used. On the other hand, the computational cost used in the proposed methods is considerably high [5].

2.7.2.2 Phrase Modeling:

Term-based models assume independence of terms, which is not the real case and is considered as poor approximation to reality. If term-dependence is considered, performance is expected to increase. Phrase Modeling considers term-dependence seeking better performance. Noun phrases are usually identified by capitalization, but apparently this is a special case.

In general phrases are recognized based on one of two classes of techniques: statistical and syntactic. Statistical models consider sequence of terms that occur together more often than they would in the term-independent models. Syntactic models on the other hand, use linguistic structures of text to identify phrases of related groups of terms.

Experiments by Fagan [17] to compare statistical and syntactical phrases proved improvements by the two classes. A considerably higher improvement is achieved by statistical phrases. Poor statistical properties of syntactic phrases may be the reason for poor performance by syntactical phrases as described by Lewis and Croft [6]. In general, phrase indexing difficulty increases if collection size increases because the increasing number of possible term-combinations forming too many phrases [5].

In this thesis we are interested in phrase indexing and phrase modeling. Therefore, phrases will be explained in more details later on in Chapter 3.

2.7.3 Principal Component Analysis (PCA):

A commonly used data reduction technique is Principal Component Analysis (PCA). It is used for dimension reduction of the term-document matrix. Principal components are linear combinations of original variables constructed to maximize the variability in data. This data can be explained by a limited number of features. In PCA, the first component is the linear combination with maximum variance. Any subsequent component shares the same property and must be orthogonal to all previous components. PCA is a very useful technique that makes us able to explain almost all the variability of the original data using a small set of components replacing a large set of features. The components are calculated using the covariance or correlation matrix of the original data. A component vector is the eigenvector of the corresponding matrix [5].

$$\underset{x}{\text{Max}} \frac{\|x^{\dagger}Sx\|}{\|x^{\dagger}x\|} = \underset{x}{\text{Max}} \frac{\|x^{\dagger}U^{\dagger}Aux\|}{\|x^{\dagger}x\|} = \underset{x}{\text{Max}} \frac{\|y^{\dagger}Ay\|}{\|y^{\dagger}y\|} \qquad \textit{(Eq. 2.12)}$$

Where S is the correlation matrix of data and equivalent to the spectral decomposition resulting into orthogonal and diagonal matrices of eigenvectors.

2.7.4 Latent Semantic Indexing (LSI):

We have previously described Latent Semantic Indexing basic concepts, as a successful model to represent the documents is a way that reduces or eliminates synonymy, polysemy, and term dependence. This section explains LSI from the point of view of dimensionality reduction.

LSI is a technique similar to PCA but it uses Singular Value Decomposition SVD as previously explained. LSI captures the important correlation structure of terms and documents. It behaves as if it performs query expansion without increasing the dimension of the output since it automatically recognizes synonyms. Using LSI factors reduces the term-dependence problem existing in the VSM. LSI is a very good tool for dimension reduction. A smaller set of feature variables is able to efficiently represent a query or a document. This is very useful because it solves the problem of over-fitting that usually occurs with statistical classification methods [5].

Disadvantages of LSI include adding extra storage and processing cost to the search system. The size of the matrix is large and its values are real numbers, which means extra space and computations to the original integer based frequency matrix. By using LSI, we can no longer make use of the sparse nature of the term-document matrix, which has serious implications.

2.8 Classification Methods:

Routing as we described it before is a special case of classification problems where classes available are only the relevant and non-relevant. Classification in general is to have multiple classes and determine to which class or more each document in the collection belongs. Classification methods use a set of training data for which correct classes are known. It uses this data to teach the system and extract the feature variables. It applies its classification rules learned during training on another set of data called the test data to classify them into their proper classes. The choice of proper feature variables is crucial to the classification process. We can not choose too many feature variables because this will result in over-fitting problem for the training data. Document-specific terms can not be a good feature variable because unique terms associated with relevant documents, when used to define classification rules, will make the classifier work perfectly in classifying the training set documents, but very poorly with the test set documents [5].

The more common terms in general have less power in classifying documents among topics. We have thousands of terms to choose from for term-based classifiers. In that case variable selection or LSI is performed to reduce the number of terms used. Variable selection method does not address the basic distribution problem associated with term-based variables while LSI eliminates the problem of term-dependence and captures the relationships between large number of terms in a single variable. LSI is expected to perform much better when combined with statistical classification methods. In IR and TC statistical classification, documents are classified to be either relevant or irrelevant to the query or topic. Irrelevant documents are irrelevant to a query, but they are relevant to other queries or topics covered by the collection. A large training set should be used to find enough relevant documents and to get a reasonable description for the distribution of non-relevant documents [5]. In the following sub-sections we will describe a few basic statistical classification methods.

2.8.1 Discriminant Analysis:

Quadratic Discriminant Analysis assumes that the populations analyzed can be modeled using multivariate normal distribution. It estimates the mean vector and covariance matrix for each group. New observations are assigned to the group to which it most likely belongs. The goal in finding a Discriminant rule is to find the linear combination of variables that maximizes the separation between groups (categories or topics). One reasonable suggestion is to maximize the separation between the means, scaling to reflect the pooled within categories covariance matrix. New observations are usually classified to the group with the closest mean vector [5].

The choice between linear and quadratic Discriminant Analysis depends on some considerations. The first one is sample data quantity. If the number of observations is less or slightly more than the number of variables, linear Discriminant Analysis is far more successful. Alternatively, quadratic Discriminant Analysis is better if the number of observations is much more than the number of variables. In IR in general, quadratic Discriminant analysis is not suitable because there will rarely exist relevant document than variables [5].

2.8.2 Logistic Regression:

Modeling a response variable using one or more predictive variables is usually called Regression in statistics. In IR regression is used to estimate the probability of relevance for each document. We use the LSI factor scores of the documents as the predictive variables. The predicted values must be allowed to vary over the whole real line and transformed in a way that will lead to having the probability estimates fall on the unit interval. Linear regression produces a poor model because of its least squares-optimization criteria. Logistic Regression has some problems like linear Discriminant Analysis. It requires more data points than variables to be able to produce accurate estimates of the parameters. Moreover, it needs much more time because the weighted covariance matrix must be inverted during each iteration of least squares, unlike linear Discriminant Analysis that does the inversion only once [5].

2.8.3 Optimal Separating Hyper-planes:

Finding the optimal separating hyper-plane directly by minimizing some measures of the error associated with the mis-classified observations is an alternative classification strategy to considering the classification boundary between two groups represented by a hyper-plane in the predictor space modeled by linear classification models like Discriminant Analysis and logistic models. In the Optimal Separating Hyper-plane model the location of the separating plane depends too much on the location of the violating relevant documents. The penalty terms for these violating documents are heavily magnified by the averaging process. The technique was not proven to fit as an effective approach for IR problem because it concentrates only on the boundary conditions. It does not attempt to capture the structure of the correctly classified relevant documents [5].

2.8.4 Classification Trees:

Also known as Recursive Partitioning. It approaches classification problem by creating a classification rule that is mainly based on a binary tree. A decision has to be made at each node. This is a boolean decision based on one variable. This approach derives the decision rule at each node by maximizing a purity measure of the two subsets over all possible binary splits that could be applied to the predictor variables. The model continues splitting until all nodes are pure or the node falls below a default size threshold [18][5].

This method works by producing nested sequence of trees. The first tree consists of the root node. Subsequent trees add splits across every eligible leaf node in the preceding tree in the sequence. The final tree almost certainly will over-fit the training data. Therefore, the goal is to determine the number of branches that must be pruned from the whole complete tree to produce the classification tree. This classification tree should be the most successful in categorizing new data arriving to the system. Usually a cost complexity function is used to balance the estimated error rate and tree size to determine the optimal tree [5].

Classification Trees were used in text categorization on Reuters collection by Lewis and Ringuette [7]. They used words as features and achieved good results in their experiment. They were also tried in Routing problems with less success [24].

The primary reason for not achieving such success in Routing appears to be using a single variable in splitting. From that it was deducted that combinations of large number of variables is important for better performance in IR and TC. An obvious disadvantage of the Classification Tree is that all leaf nodes are of similar score, so this method does not provide a complete ranking for documents. A result to that is more complexity in the evaluation process [5].

Chapter 3

Text Categorization Methodology

Chapter 3
Text Categorization Methodology

3.1 Introduction:

In this thesis we will investigate text categorization using phrases as the primary categorization method. Our categorization strategy will focus on using high quality features (terms and phrases) to classify a subset of Reuters - 21578 test collection to a predefined set of topics. This chapter explains basics of phrase indexing and describes our categorization strategy and how we test our hypothesis.

3.2 Phrase Indexing:

In this section we will define what a phrase is and explain the types of phrases and some important definitions as an introduction for this important part of research. We will then explain phrase structures and how phrases are identified or extracted in IR and TC. We will also be talking about phrase weighting and using clustering with phrases.

3.2.1 Introduction and Definitions:

In a broad perspective, text-based systems are either text categorization systems or text comprehension systems. Text categorization includes the very related areas of IR and TC. IR is mainly concerned with matching information with user query and retrieving the relevant information to the user. TC is concerned with determining the relevance of information to a pre-defined set of categories. Text comprehension on the other hand goes beyond that to transform text to another form such as producing summaries or answer questions.

TC classification function uses natural language based on word statistics and machine learning. In machine learning methods a large set of previously categorized documents are used to make the system construct a classification function [8]. Many modern IR models adopted the concept of "bag-of-words" in dealing with document indexing. A document is a set of words and each word is considered as a dimension in the vector representing the document [26].

Phrases in IR and TC are known with other names such as multiword features and nominal compound [16]. A phrase is defined in the fields of IR and TC as a textual unit usually larger than one word and smaller than a sentence. Another definition is used in this context is n-grams. A 1-gram (unigram) is simply a word. An n-gram is an alphabetically ordered sequence of n unigrams [20]. The first process we have to apply to deal with phrases in IR and TC is to identify phrases. Identifying a phrase is selecting pairs of words in a query to be recognized as a phrase and call it a query phrase. The same process should be applied in documents identifying documents' phrases. Phrase structuring is a process that follows phrase identification process. It recognizes query phrases again in documents. Every recognized phrase in the document must be scored contributing to the total score of the document. Phrase weighting techniques include phrase frequency, phrase IDF, and higher scores for lower proximity phrases [16].

Phrases have been found to be useful indexing units by most of the groups participating in NIST and DARPA sponsoring TREC. Two classes of phrases are explained in our research. They are statistical and syntactic phrases. Statistical phrases are any pair (or more) of non-function words that occur contiguously often enough in a corpus or collection used. Syntactic phrases on the other hand are any set of words that satisfy certain syntactic relation or constitute specified syntactic structures. For example, we can say an adjective followed by a noun make up a phrase [19]. In general, syntactic phrases in IR have obtained discouraging results. A possible reason is that indexing languages based on phrases have inferior statistical qualities with respect to indexing languages based on single words. Another possible reason is having many phrases that denote non-interesting concepts. Statistical phrases are better than syntactic phrases because they are easier to recognize by more robust and less computationally expensive algorithms. Moreover, the effect of irrelevant syntactic variants and uninteresting phrases can be factored out [20]. The current work focuses only on using statistical phrases.

Many collections and topics are used in phrase indexing experiments. Phrases are sometimes extracted from TREC topics [16]. The standard benchmark for TC research though is Reuters-21578 collection, which contains 21578 documents. Distribution 1.0 corpus is the most widely used benchmark in this area of research. Caropreso, Matwin, and Sebastiani [20] used Reuters-21578 in their statistical phrases experiments in TC with 12902 news stories. Documents have an average length of

211 words, and labeled by 118 categories. The average number of categories per document is 1.08 ranging from 0 to 16 categories. It contains 17439 unigrams.

3.2.2 Phrase Identification:

Identifying a phrase is the starting point in the process of dealing with phrases. Sometimes also called feature selection considering phrases as features. A phrase identification function or feature evaluation function (FEF) is used to select a subset of the total set of phrases. These selected phrases are considered the most important phrases in describing a document or query. The reduction from the original set to the subset of selected phrases is due to using a reduction factor in the formula or function used. Most of these functions are based on frequency statistics. Examples for FEF are the document frequency (DF), Information Gain (IG), Chi-square (X^2), and Odds Ratio (OR) [20].

The number of all possible phrases to be extracted from a document is probably very large and from a collection is huge. This number depends on the number of phrase terms. Most of these phrases are of no real meaning and considered as noise. A filtering process must be used to filter out all noisy phrases. The FEFs mentioned above are examples of such filters. Without filtering, it is impossible to use phrase indexing in IR or TC. Although these filtering techniques reduce this noise to a great extent, there still might be some noise or undesired phrases.

3.2.3 Phrase Structure:

As we previously mentioned, phrase structuring is the process of recognizing an identified phrase again in documents. There are two important techniques used for this process: anaphora and proximity. Anaphora is a word or phrase that references another word or phrase [16]. An example for that is the phrase "U S" which is an anaphora for "The United States of America". One conjecture is that if an anaphora and its original phrase co-reference the same object, they will also co-occur within the same document. The contextual strength between an anaphora and its original phrase is calculated using all documents in the collection. Usually co-occurrence measures the value of that strength.

Phrase proximity is the distance between phrase terms. So proximity equal one means that the phrase terms are adjacent. Phrases that have proximity equal one are called adjacent phrases. Boolean phrases, on the other hand, means that the words of the phrase exist in the document at any lexical distance. Phrases can sometimes occur somewhere in between. That happens when phrase terms have a distance more than one but less than the document length. Usually documents that contain adjacent phrases are called adjacent documents, while documents that contain boolean phrases are called boolean documents. Many experiments were made and suggest that it is more evident that adjacent phrases result in better performance than boolean phrases [16].

3.2.4 Phrase Weighting:

When phrases are identified and learned by the system. A phrase structuring process detects these identified phrases in documents and a weight is assigned to each detected phrase. These weights are then combined to contribute to the total document weight. Phrase IDF is one of the techniques used to give weights to phrases. In an experiment by Pickens and Croft [16] IDF was used as weighting method for phrases and they found out two main differences between words and phrases in that context. They found out that phrases are rarer than words and are not that useful as words. They suggested that phrases enhance the performance but can not be used alone. They also noticed that adjacent phrases are better when used with IDF than boolean phrases. They suggested a balance between adjacency and boolean phrases since the first is precise but sacrifices recall while the later is the opposite. In conclusion they reported that IDF is a useful weighting method for phrases as it was with terms.

Other functions for phrase weighting are available such as phrase frequency analogous to term frequency, empirical down-weighting method, Fagan's method, and approximation to Robertson's method [27].

3.3 Test Collection:

Reuters 21578 collection is selected because it is the most widely used test collection for text categorization. It contains 21578 varying sizes news documents in text format. These documents are stored in 22 files, the first 21 of which are a 1000

document each and the last file contains 578 documents. Each document might contain a title and it contains relevance judgement that classifies the document to one or more of 672 categories. These categories are clustered into 135 topics, 56 organizations, 39 exchanges, 175 places, and 267 people. These are the actual numbers as explained by David D. Lewis [9] in the descriptive files of the collection. We selected the topics cluster and processed only on documents that are pre-classified to one or more of these predefined topics.

We selected a sample size of 250 documents for our experiments. We used 150 of them for training the system and extracting the index terms and phrases and 100 for testing. Some of these documents had no topics associated with them and therefore they are ignored while parsing the documents. The resulting training set contained 130 document-topic pair while the test set contained 66 document-topic pair. The training set represents 32 category and we only processed test documents that belong to any of these categories.

3.4 Categorization Strategy:

Training documents pass through a pre-processing stage that prepares the document data for learning. After pre-processing each document a data collection stage starts that stores and indexes all gathered information from the document. After all training documents are finished through pre-processing and data collection stages, a learning stage starts to collect frequency statistics. Test documents on the other hand pass through the same pre-processing stage and data collection stage, but they do not go through the learning phase. Instead they go through a categorization stage that uses the collected information of the document and statistical data learned from training documents and apply categorization rules to classify each document to one or more category.

3.4.1 Pre-Processing and Learning:

The present methodology starts by parsing all documents in the training set, filtering out stop words, stemming remaining words, and extracting all possible terms and phrases. A feature reduction technique based on statistical methods is used to filter out low-quality terms and phrases to reduce the inverted index size. A list of

index terms is constructed and another for phrases. These index terms and phrases are mapped to the topics (categories) to which their containing documents belong. A categorization rule is devised using learned statistics to map documents of the test set to the appropriate categories based on the terms and phrases they contain.

Parsing, stop word removal, and stemming are the preprocessing steps used to prepare the data for further processing steps. Another pre-processing step that follows these steps is phrase identification in which we identify all possible phrases in a document. For all documents in the training set, all terms and phrases are extracted and stored in two separate lists. These terms and phrases are then bind to categories to which their containing documents belong.

After finishing the pre-processing stage for all training set, a learning phase starts by filtering out all low-quality terms and phrases and selecting high-quality ones for indexing using a compound feature evaluation function (FEF). Frequency statistics are calculated during this learning phase to prepare needed data for categorization stage.

3.4.2 Feature Reduction and Categorization:

A testing phase processes all test documents by extracting all possible terms and phrases, ignoring all non-index terms and phrases and using the indexing ones to map the document to relevant categories. Relevance judgement records are used to compare them to categorization results to measure the system performance.

In our methodology we are interested in investigating phrases as a major categorization method. We also investigated other suggested techniques that we believe to be useful for enhancing categorization performance of phrase-based categorization. Therefore, we will study documents categorization using each technique separately and evaluating its performance. After evaluating each individual technique, we will study combining these techniques together to see how their collective results could enhance categorization performance.

Two other proposed categorization techniques are investigated that are based on term statistics, beside term categorization technique. The first technique is the category-term-based categorization. If a document contains a term that is the same as a certain category name, it seems logical that this document might be related to that category. Therefore, we will investigate relevance based on category terms as an

independent classifier. The second technique is title terms of the document. The title of the document contains terms that are considered key classifiers to the whole document. A human reader for that document in most cases can understand what the document is talking about by reading its title. Therefore, a document title should be handled differently giving it higher weight when calculating relevance. For that reason, we will investigate categorization based on title terms as an independent classifier. We also investigated categorization based on document terms to be able to compare our proposed techniques to the standard most widely used technique using same environment and conditions.

Term Frequency (TF) and Document Frequency (DF) of terms are very widely statistical weighting techniques for document relevance calculation. Similarly we considered Phrase Frequency (PhF) and Phrase Document Frequency (PhDF). These weighting techniques are used in both IR and TC fields. For our TC research we added another weighting technique which is the Category Frequency (CF) and used it for terms and phrases. We think this technique will result into better performance. The logic behind that technique is that we are not interested in identifying each and every document by itself, but we are interested in identifying all documents that belong to a certain category and the CF technique explains this relation better than the usual Document Frequency (DF) technique. For example, if a term is repeated 10 times in 10 different documents, that does not tell us anything about the quality of this term when used to identify to which category does this document belong. It does not tell us enough information to say whether we should use it as an index term or not. However, if we know that these 10 documents belong to one category, then this is definitely a high quality term that identifies this certain category. If a term is repeated in all categories, it can not be used to classify a document to any category, but if it occurs only in one category, it is definitely a high quality classifier-term.

Combining CF, with TF or PhF, and DF in a single FEF seems to be very useful in filtering out low quality terms and phrases and selecting only high quality ones to be used for indexing. We applied a combination of these techniques selecting only those terms and phrases that have low CF value while having high frequency and DF values. Choosing low CF terms and phrases is essential in selecting index terms and phrases. We need to use only terms and phrases that separate between categories to achieve high precision. At the same time high TF and PhF are important to select only terms and phrases respectively that are of significant value to the category they

represent. If a term is only represented once in one document, it will be selected by the CF function to be an index term while it is a low quality specific term only to that document but not the whole category. If that term's TF is high but occurs in a single document, it becomes a high quality term, but only for that document. If it is repeated in more than one document in the same category, it becomes an important term for the whole category and hence is considered as a high quality term for that class and can be used as an index term.

Phrases are very similar to terms. Phrases must not be identified only on co-occurrence. Phrases are terms combined together by selecting a certain proximity value. If we choose only adjacent phrases, we will have all possible phrases in the document identified by their adjacent co-occurrence. They are just terms that happen to be adjacent. A phrase is a true significant phrase only when it is repeated in other documents. Therefore by using PhF and PhDF we can filter out phrases that are identified only on misleading co-occurrence. An index phrase should have one more criteria. Phrase terms should be as specific as possible to this phrase. If phrase terms occur as independent terms more often than within the phrase itself, the whole phrase becomes a weak phrase. On the other hand if a term does not occur except within the phrase, this gives more power to the phrase and increases the probability that it is a phrase with significant meaning.

The filter values for TCF (Term Category Frequency), TF (Term Frequency), TDF (Term Document Frequency), PhCF (Phrase Category Frequency), PhF (Phrase Frequency), PhDF (Phrase Document Frequency) are set according to experiment. We tried several values and selected those values that reduce the index size and at the same time achieve minimal loss of information and at the same time increase high precision of classifiers. For TF filter, we first calculated the maximum frequency of all terms in each category and obtained the average of these maximum values, then applied filters to filter out those terms that are less than a certain percentage of this average. We applied the same technique for PhF, TDF, and PhDF filters. CF filter was easier to calculate. We set a fraction of the total categories learned by the system using the training set.

We calculated document-category relevance using category-term classifier based on the frequency of occurrence of category-term within title and document terms. All category terms that are found during processing test documents are considered index terms automatically. Title classifier calculates document-category

relevance score based on frequency statistics of index terms found in the title and their probability of relevance to categories based on learned information during the learning stage. For terms relevance we calculated two scores, one based only on the term frequency in the document and its relevance to that category while the other is based on TF in that document, the term relevance to the category, and the term DF and CF values. For phrase relevance, similar to term classifier, we calculated two scores: one for the phrase frequency in the document and phrase relevance to the category. The second is for phrase frequency in the document, phrase relevance to the category, phrase DF, phrase CF, and the frequency of the separate terms of the phrase relative to the phrase overall frequency.

The logic behind the second score for terms is to combine several factors together. Measuring the importance of the term to the document using its frequency in the document is one factor. The more frequent the term occurs in the document, the more important it is for that specific document. This TF is normalized by the document's length. Measuring the term quality as a classifier term between categories is another factor and calculated by measuring the ratio between the term's DF and CF values. Measuring the importance of the term to a specific category is another factor. It is calculated by multiplying two ratios. The first is the ratio between the term frequency in all documents that belong to that category and the total number of frequencies for all index terms that occur in all documents that belong to the same category. The second ratio is the number of documents in that category where the term occurs to the total number of documents pre-categorized to that category.

Phrase second scoring formula is similar to that of terms with one extra ratio. It is the ratio between the frequency of occurrence of the phrase and the summation of all frequencies of the terms of that phrase. This is to give less weight to those phrases that their terms occur as independent terms more frequent and more weight to those phrases that their terms do not occur or occur only a few times as independent terms other than occurring within the phrase. The reason behind investigating two formulas for term and phrase classifiers is to study the effect of the added part in the second formula and see if it is useful or can be ignored to reduce processing overheads.

Having separate scores for different techniques enables us to investigate each technique alone as an independent classifier. Our focus is to study phrases as an independent classifier and combined with other categorization techniques. Depending on these scores, we are able to decide if we need to further tune the index attributes or

not and which techniques perform better than the other. The goal of the filtering process is to obtain a small set of high-quality index attributes for categorization. We then feed the output of the filtering stage to the categorization rules to generate document-category relations based on each categorization technique independently.

3.5 Categorization Software System:

In order to test our hypothesis for the proposed feature reduction and categorization techniques, we implemented an Object-Oriented software system using C++ language. The system is not really a part of the research focus of this thesis, but we see it is useful to briefly mention some information about it. The main reason for describing the software in this thesis is its powerful design that enables users of the system to control almost all parameters of feature reduction and categorization processes. Using a user-friendly interface, the user can easily specify training set size and test set size. He can also specify filter values to be used during data collection phase, learning phase, feature reduction phase, and categorization phase. He is also able to specify filtering values and strategy used for filtering out the final result set.

Moreover the system is well designed using object-oriented programming and well documented. This enables researchers to use the system not only for investigating what is does, but also to modify it to investigate other related research issues. The system is easy to understand and modify in very little time. A user can modify the code of the system to change the stop words list in seconds while support reading a totally new test collection in a few hours.

The second main reason for describing the system is to enable the user to better understand the sequence of processing for both training and test data sets. The process is explained before in the previous section but it can be better understood using a process diagram. The following figure (figure 3) describes the categorization process).

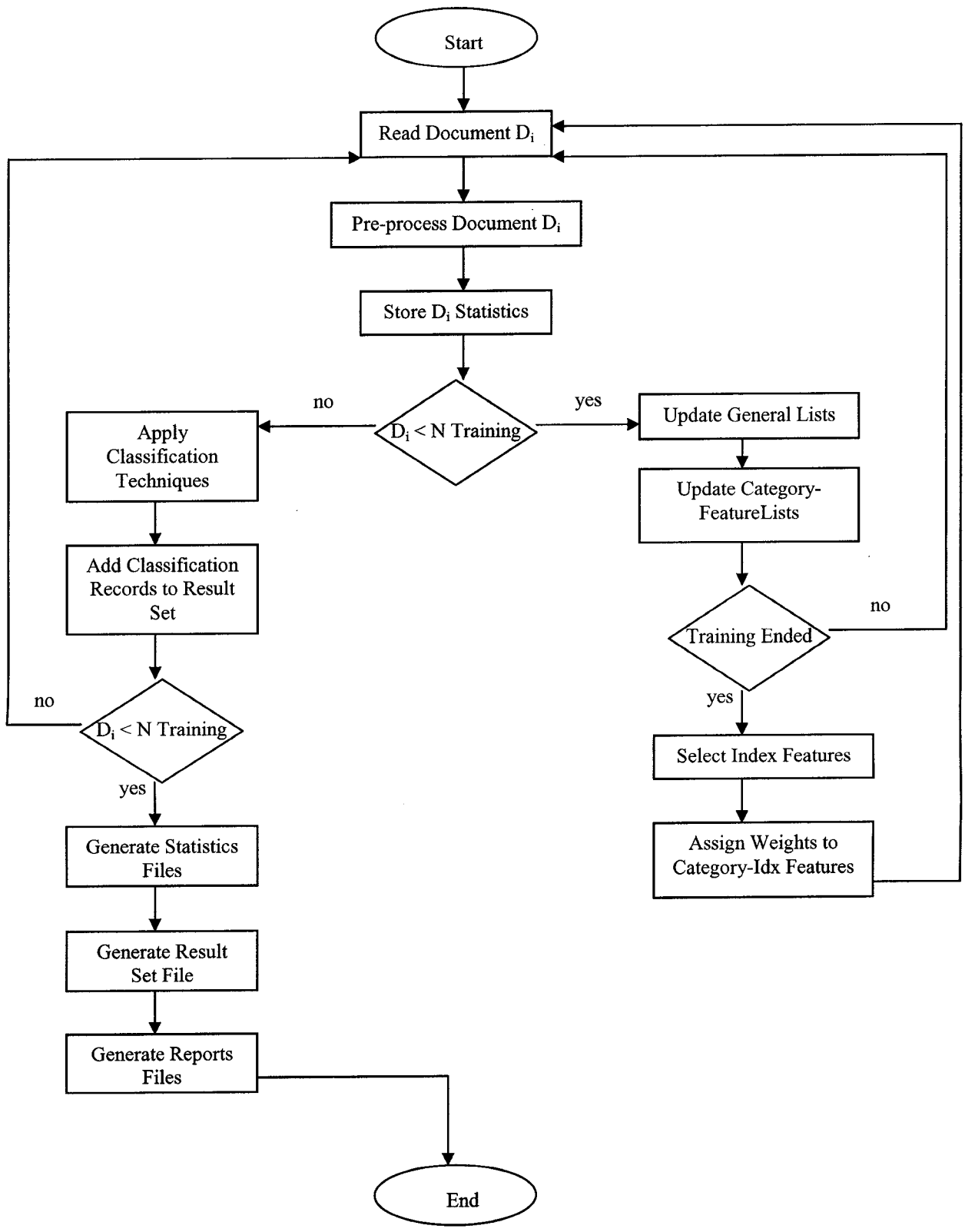

Figure 3: Categorization System Process Diagram

3.6 Summary:

In this chapter we explain phrase indexing, phrase types, identification and structuring. We discussed Reuters – 21578 test collection, which we used for our research. We also highlighted the pre-processing, data collection, and learning stages that training documents pass through. We also explain how test documents pass through the same pre-processing and data collection stages. Statistics gathered during the learning stage are applied on test documents in categorization stage to categorize documents to predefined categories. We explained feature reduction techniques, which we propose based on threshold techniques. Categorization techniques are then explained in abstract. A statistical phrase classifier is proposed as our main focus of the present work. Other term-based categorization techniques are explained in abstract such as category-term classifier and title classifier. We also explained the proposed concepts applied on phrase categorization when applied on term categorization. Finally we described the system used to test our proposed techniques and explained that it can be used as a testing tool for other research efforts in the field of TC.

As mentioned earlier, the primary goal of this thesis is to study Text Categorization based on statistical phrases as an independent categorization method and prove that phrases can be used to achieve high-precision categorization. The secondary goal of the present work is to study other term-based categorization techniques also proposed in this thesis that can be used to enhance phrase-based categorization such as using category-term categorization and title-terms categorization. We think that these new term-based techniques can be at least as precise as term classifiers with much less computational costs.

In the following chapters, we will present in details the different phases of preprocessing and processing stages for the documents and the results achieved by each stage and the logic behind moving from a stage to another. The following chapter will explain preprocessing steps, learning, and feature reduction based on threshold techniques. Chapter 5 will explain categorization methods and categorization results. Chapter 6 will conclude and suggest future work that could build on the top of the present work.

Chapter 4

Preprocessing, Learning, and Feature Reduction

Chapter 4
Preprocessing, Learning, and Feature Reduction

4.1 Introduction:

In the previous chapter we had an overview of the proposed categorization techniques and different pre-processing and processing stages in which documents pass through. In this chapter we will explain the preprocessing steps taken to prepare the data for further processing stages. Each document in both training and test set passes through parsing, stop word removal, and stemming. The document then is processed to identify all possible phrases. We will then explain the learning stage and feature reduction techniques. We will show how index lists are constructed and how the system creates the relation between different terms and phrase and categories.

4.2 Preprocessing:

Pre-processing stage is the first stage that a document has to pass through to prepare the data for further stages. All documents must pass through this stage to recognize the document structure, parse it, remove stop words, stem remaining document words, and identify phrases. We have four steps in this stage and will be explained in their sequence of processing.

4.2.1 Parsing:

Reuters-21578 has a standard format that makes it easy for the system to parse the document and recognize its structure. It contains 21 text files of 1000 documents each and the 22nd and last text file contains 578 documents. Each document has a standard format that makes it easy for the parser to identify the beginning and ending of the document as well as its title, topics to which it is judged to be related, and text. There are other sections for the document structure such as people, dates, places, organizations, exchanges, etc. but they are out of our scope and therefore ignored while parsing. The parser ignores all documents that have no relevance judgment that relates the document to at least one topic. Parsing test set documents adds one more condition that a document must satisfy before it is parsed. A test document must be

relevant to at least one topic of those learned while training the system using training set.

While parsing, all special characters and numbers are ignored. The parser processes line by line trying to build up a sentence. It identifies the end of the sentence by a period, special collection-specific format characters, or end of document. If a complete sentence is found within a line, it processes it; otherwise, it continues reading lines until a complete sentence is formed. Therefore this step outputs a complete sentence of words of alphabetic characters. Only the character "-" is kept because it represent a part of some words such as "object-oriented" and used frequently in this collection. All numbers and special characters are ignored. The parser also capitalizes all characters to have a uniform output.

Each sentence the parser completes is sent for next step and so on until all steps of pre-processing stage process it then the parser starts to read the next sentence from the source file. The resulting output of the parsing phase is full of function words and other words that has low categorization value and should be removed to reduce the number of features which the system will use for indexing purposes. This is achieved by the next step in the pre-processing stage, which is stop words removal.

4.2.2 Stop Words Removal:

The output of the parser is a list of words. They are compared against a list of stop-words that contains all function words, other words based on general language word frequency counts, and we added the word "Reuters" since it is considered as a non-content-bearing word specific for this collection. It is repeated in almost every document in this collection and it has a very low discrimination value for all categories. The stop-words list used is available in appendix A of this thesis and it is the combination of most terms of two lists. The first is the standard Verity stop list and the second is the Defense Technical Information Center / Defense RDT&E Online System (DTIC-DROLS) [10].

The goal of this step is to remove all terms that are known to be of low discrimination value. This stop-words list contains words that are repeated in almost every English document and thus can not be used to distinguish documents from each other and can not be used as index terms for TC purposes. The resulting output of this step is a list of words that do not include function words or non-content bearing words

but some word linguistic variations are considered different words and might co-exist in the output list. This means that there might be different words that have the same linguistic root in the output of this step. Stemming is useful to reduce these words by obtaining words roots and reducing linguistic variations of the same word to one term.

4.2.3 Stemming:

Stemming is a process of removing prefixes and suffixes of words to obtain the raw form or linguistic root of words. The result of this process is a list of terms that represent the roots of words in the document. Porter stemmer as described before is one of the most widely used stemmers and it uses a set of rules that are applied in sequence to a word and the output of the final rule is the stem of the word. Stemming in general sometimes ends up with a stem that might be confused with another word's stem. Porter stemmer has the same defect, but its ability of reducing the list of words and grouping different word linguistic variations into one word stem makes it useful to use and make us ignore its deficiency.

The stemmer measures the variable m for this form $[C](VC)^m[V]$ where C is a sequence of consonants and V is a sequence of vowels including "y". Examples for m are (m=0) in "tree" and (m=1) in "trees" or "trouble" and (m=2) appears in "troubles" or "private". The algorithm applies a set of steps in sequence to transfer the word from its old form to its new form. In all the steps it searches for a certain sequence of characters and replaces it with another sequence. For examples:

Step 1a: "sses" is ➔ "ss" as in (caresses ➔ caress)

Step 1b: if m > 0: "eed" ➔ "ee" as in (agreed ➔ agree)

Step 1c: *v*Y ➔ I as in (happy ➔ happi)

Step 2: for m > 0 do the following replacements:

Step 3: for m > 0 do the following replacements:

Step 4: for all m > 1 remove the following suffixes:

Step 5a: For m > 1 remove the suffix "e" as in (probate ➔ probat)

Step 5b: For m > 1 and *D and *L ➔ single letter as in (controll ➔ control)

A complete description of Porter Stemmer rules is available in appendix B of this thesis. The result of this step is a list of terms. We will use this list of terms to identify phrases and filter out low-quality terms and phrases in order to select only high-quality terms and phrases to be used for indexing.

4.2.4 Phrase Identification:

Since research results, as previously explained in the survey chapter, suggested that adjacent phrases perform almost as good as higher-proximity-phrases with much less processing overheads, we decided to use only adjacent phrases in our research. Another reason is the proven higher precision for adjacent phrases, while highly precise phrase categorization is our main goal of this thesis. This reduces the complexity of the phrase identification process and reduces phrase list size. We also selected the number of terms in the phrase to be 2 terms. A phrase is any two adjacent terms within a single sentence. Phrase identification process results in a very large number of phrases.

All pre-processing steps including phrase identification are applied on both training and test documents. They extract features from training documents to be used for training and learning and from test documents to be used for categorization.

Having a large number of phrases as well as terms initiates the need for feature reduction to filter out low quality terms and phrases and use only high-quality features to index the documents. In order to reduce the number of these features, we need to learn first which features should be kept and which should be filtered out. The following section explains how the system learns information that enables it to reduce the number of features and how this reduction occurs. Part of the learned information is also used in the following categorization stage.

4.3 Learning and Feature Reduction:

Pre-processing stage is applied for both training and test documents. Learning and feature reduction phases are processed only on training set data. Learning occurs in two stages. The first as mentioned in the previous chapter is considered a data collection process and occurs after each training-document is pre-processed. The second learning process occurs after all training documents have been pre-processed. The result of pre-processing a training document includes two output lists. These lists are processed in data collection stage to gather statistical information about terms and phrases in that document. The first list contains the list of terms in that document, while the second contains the phrases in the document. These two lists are used to

update the global system lists that contain all terms and phrases in all training documents read by the system. They are also used to build another two general lists. The first creates relations between each term and the categories to which this document belongs. The second creates relations between each phrase and the categories to which this document belongs.

After pre-processing all training documents and constructing the general lists, the second learning stage starts. We get frequency information within a document from the document-specific lists while the general frequency, DF, CF information from the general lists. We also get statistics of a term or a phrase relation to a certain category from the category-term and category-phrase lists respectively. Learning uses general frequency information to assign scores to all terms and phrases in order for the feature reduction phase to select the high-quality features to be used for indexing.

The output lists of the learning phase contains large numbers of terms and phrases as shown in Table 1. This table lays out the number of distinct terms, phrases, and categories learned from training set documents. These large numbers must be reduced for indexing purpose. Therefore they are fed to the feature reduction phase to select only high-quality terms and phrases. The indexing terms and phrases will be used to categorize test documents to categories learned during the training phase.

Table 1: Data Learned from Training Set

Total Terms	2023
Total Phrases	5838
Total Categories	32

In fact learning and feature reduction are two tasks that are mixed together. Feature reduction works on lists mentioned above to select indexing terms and phrases. Learning starts again after that selection to calculate relevance scores of terms and phrases to categories.

Feature reduction stage uses the general lists of terms and phrases and the general lists of terms and phrases relation to categories as inputs and outputs two indexing lists. The first list contains index terms and the second for index phrases. As explained before in the past chapter we consider a term as an index term if it has low CF value and high TF and DF values. Filtering concept is the same for both terms and

phrases. We will explain this concept only for terms only since phrase filtering is similar to that of terms.

A term is considered an index term only if it passes these three condition:

A) TCF < FilterCF * NC *(Eq. 4.1)*

B) TF > FilterTF * Avg (MaximaTF) *(Eq. 4.2)*

C) TDF > FilterDF * Avg (MaximaTDF) *(Eq. 4.3)*

Where TCF is the number of categories where this term occurs, NC is the total number of categories found in all relevance judgement records of the training set, TF is the general term Frequency, and TDF is the total number of documents where this term occurs. MaximaTF is the maximum of all maximum frequency terms in all categories. MaximaTDF is the maximum of all maximum number of documents per categories. FilterCF, FilterTF, and FilterDF are coefficients for CF, TF, and DF respectively.

We tried several values of FilterCF , FilterTF and FilterTDF. A good filter value combination should reduce the number of index terms and phrases as much as possible and at the same time maximize precision. Table 2 shows a sample of different filter values tried and the resulting numbers of index terms and phrases. By applying the filter values of any row to terms, we get the number of index terms in the same row in column "Index Terms". By applying the filter values of any row to phrases, we get the number of index phrases in the same row in column "Index Phrases". Our goal is to obtain the minimal number of terms and phrases that can index maximum number of documents to their relevant categories with high precision using our categorization techniques.

Decreasing the value for FilterCF means filtering out more features and selecting less number of index features. Increasing FilterF and FilterDF filters out more features and select less index features. By analysis of the results we notice that when we filter out all features as in Run 28, we still get 21 index terms and 0 index phrases. These 21 terms are category names found within document terms and are selected as index terms without passing through any filters. The reason for that will be explained in details in the next chapter. We also notice that FilterCF must be very aggressive to achieve little features reduction, while FilterDF needs to be less aggressive to achieve even better reductions. We also notice that FilterF is the most powerful in reduction. We also notice that combining the three filters achieves a great reduction percentage in both terms and phrases.

Table 2: filter values and resulting index terms and phrases.

Run	FilterCF	FilterF	FilterDF	Index Terms	Index Phrases
1	0	0	0	2023	5838
2	0.9	0	0	2023	5838
3	0.7	0	0	2023	5838
4	0.1	0	0	1805	5600
5	0.05	0	0	1500	5342
6	1.0	0.25	0	756	5838
7	1.0	0.5	0	370	270
8	1.0	0.75	0	223	139
9	1.0	1.0	0	162	57
10	1.0	1.25	0	116	44
11	1.0	1.5	0	89	29
12	1.0	0	0.25	2023	5838
13	1.0	0	0.5	2023	5838
14	1.0	0	0.75	888	5838
15	1.0	0	1.0	550	669
16	1.0	0	1.25	550	669
17	1.0	0	1.5	396	154
18	0.1	0.75	0.75	126	126
19	0.1	1.0	0.75	90	49
20	0.1	1.0	1.0	87	46
21	0.05	1.0	1.0	37	39
22	0.05	1.25	1.0	29	30
23	0.05	1.25	1.5	28	28
24	0.05	1.0	1.5	34	36
25	0.05	0.75	0.75	54	108
26	0.05	1.5	1.5	25	19
27	0.05	2.0	2.0	25	4
28	0	100	100	21	0

I

At this stage we can not figure out how good or bad are the performances of these filters to the final categorization process since we do not yet know their effect on categorization performance. We need to use the resulting index terms and phrases in a categorization technique to measure precision and recall values based on these different filters. However as an independent result and based on boolean existence of index terms and phrases in documents, even the most aggressive filter combinations achieve very high recall values. This means that even the most aggressive filters resulted in minimal loss of information while greatly reduced index sizes. This will be explained in more details in the following chapter when explaining categorization techniques.

After the application of a categorization technique and measuring precision and recall values, we will be able to select the filter values that give us the best performance values according to data analysis. We expect these filter values to depend on the test collection used and its size. Therefore, we will only explain the technique and the process but filter coefficients values selection need further research on different larger test sets and that is not the focus of this thesis. For that reason we will experiment different filter coefficient combinations and measure performance of each categorization technique at each combination. In the next chapter we will explain the categorization techniques and show results achieved by each technique for the sample filter coefficients used. These sample coefficients are selected because they result into a relatively small number of index features and thus will limit the size of the result set contributing to the increase of precision.

It is important to say that even without using filters at all, the system still do not achieve 100% recall. This is because there is one document that has a different format than the whole collection. The parser did not recognize its structure correctly and did not read its text, but it was able to read its relevance judgement only. This document is ignored because its existence produces misleading results.

4.4 Summary:

In this chapter we explained four pre-processing steps applied to documents. Text of each document is parsed, compared against stop list to remove stop words, then stemmed. The output terms are used to identify adjacent phrases of phrase size of two terms. Frequency information is learned and used in a proposed filtering

technique to reduce the number of terms and phrases and select high quality features to be used for indexing. Features' relevance to categories are calculated for future use by categorization techniques. Different filter coefficient values were tried to study the behavior of the proposed filtering technique. Filtering technique performance achieved very good results in reducing the number or features and performance will be experimented by different categorization techniques that we will propose in the following chapter. The following chapter will explain our proposed categorization techniques and results achieved by each one of them at samples of different filter coefficient combinations. It will also compare performances of these techniques together and study the effect of combining them together. A comparison between our main proposed categorization technique based on phrases and other researchers results based also on phrases will be also presented.

Chapter 5

Categorization Techniques and Results

Chapter 5
Categorization Techniques and Results

5.1 Introduction:

In the previous chapter we showed how the system pre-processes documents and reduces terms and phrases lists. We also proposed a filtering technique that we used for feature reduction and experimented its effect. This chapter presents a categorization rule that assigns scores representing the document relevance to categories using our proposed categorization techniques. We have four different scoring schemes for the document-category pair. The main technique that represents the focus of this thesis is based on phrases. The other ones are proposed to suggest other techniques for TC and also to compare their results to phrase categorization technique. The three other techniques are based on terms. One of them is focused only on terms that are similar to category names, the second is focused only on document title terms, and the third is focused on all document terms. The last one is the traditional and most widely used in concept, but with the application of our new categorization rule.

We present separate results for these techniques and compare between them and also study the effect of combining them together. At the end of the chapter we compare our phrase classifier performance results to those of other researchers based also on phrases.

5.2 Categorization Techniques:

In this section we will explain the different categorization techniques proposed and the relevance scoring schemes for each technique. We will start by category-term relevance followed by title term relevance, then term relevance followed by phrase relevance.

5.2.1 Category Term Relevance:

Since the category name seems to be the most relevant term to that category, we investigated document-category relevance based on terms similar to category

names. We calculate document-category relevance based on the frequency of occurrence of category name terms in the document. A document is considered relevant to a category if the category name occurs in the document. Since a document title probably reflects the contents of the whole documents in a few words, we gave category-term higher score if it occurs within the title.

Since all category names found in the test set are considered index terms, relevance on category terms does not depend on the filters previously introduced in the feature reduction process (chapter4). The following table (table 3) shows precision and recall based on category terms relevance.

Table 3: Precision and Recall based on Category-Term Relevance

	No. Of Documents	**Precision**	**Recall**
Category Term in Title only	1	100%	1.5%
Category Term in Document only	16	56.25%	25.6%
Category Term in Title & Document	7	85.71%	10.77%
Total	24	66.66%	36.9%

By analyzing these results, we can notice that this technique achieves good precision values if a category-term occurs in the title or in both title and document text. We also notice that it is less precise if it occurs in document terms only. This technique seems to be a promising classifier technique as a supplementary technique to another major one. This will be clearer when we combine results of different techniques together.

5.2.2 Title Relevance:

A document's title is parsed before the document itself. We believe that the title can be a key factor for the categorization problem. If manual categorization is performed, a human may be able to classify the document based on its title for many documents without having to read the document itself. General term frequency and relevance score to categories are used to calculate title-terms relevance scores to

categories. Index terms found in title are evaluated one by one to calculate their relevance to the category.

The following formula is used to calculate the relevance score based on title terms:

$$R = (CatTermRel * TDF) / TCF \qquad \text{(Eq. 5.1)}$$

Where TDF is the total number of documents where this term occurs and TCF is the total number of categories where this term occurs. CatTermRel represents the relevance of this term to the category and is calculated as follows:

$$CatTermRel = \frac{\left(\left(\frac{CatTermTF * 100}{TCTF}\right) * \left(\frac{CatTermDF * 100}{TCD}\right)\right)}{100} \qquad (Eq.5.2)$$

Where CatTermTF is the term frequency in all documents belong to this category, TCTF is the total frequency for all terms in this category, CatTermDF is the term Document Frequency in this category, and TCD is the total number of documents pre-classified to this category.

This later formula represents the relevance of a term i to category j. The first part of it calculates the probability of having this term relevant to the category based on its frequency within this category as compared to the frequency all terms related to this category. If a term occurs 50 times in this category and this category has total terms frequencies of 100, then there is a 50% probability that a document containing this term belongs to this category. The other part of the formula emphasizes the same concept but based on the document frequencies. If a term occurs in 5 documents in this category and the total number of documents classified to this category is 10 documents, then there is a 50% probability that a document containing this term belongs to this category. We combine the two ratios in the formula to achieve more accuracy and higher precision values.

The following table (table 4) shows precision and recall values using categorization technique based on title terms only.

Table 4: Precision and Recall based on Title Terms Relevance

Run	FilterCF	FilterTF	FilterDF	Retrieved	Precision	Recall
1	0.1	0.75	0.75	159	25.78	63.08
2	0.1	1.0	1.0	149	26.17	60.00
3	0.05	0.75	0.75	93	26.88	38.46
4	0.05	1.0	1.0	91	26.37	36.92
5	0.05	1.5	1.5	86	23.25	30.77

By analyzing this data, we notice that both precision and recall values are low for relevance based only on title terms. That does not mean that this is the final conclusion about this technique, but it only suggests that it is not a good independent technique. We will perform more analysis when combining all results of all studied techniques to get a better understanding of the problem and provide better categorization based on combining techniques.

5.2.3 Terms Relevance:

Terms relevance is very similar to title relevance. It calculates the relevance of each term i to the category j and the summation of all terms' scores represents the overall relevance of this document d to the category j based on terms. Only index terms are considered. If a non-index term is found, it is ignored. The resulting summation of all scores is normalized by the number of index terms found in the document. The following formula represents the Relevance of the document d to category j based on terms:

$$R = \sum \left(\left(\frac{DT_i F}{DTL} \right) * \left(\frac{T_i DF}{T_i CF} \right) * (CatTerm\,\mathrm{Re}\,l_i) \right) \qquad (Eq.5.3)$$

Where DT_iF is the document term frequency for term T_i, DTL is the document length in terms, and T_iDF is the general document frequency of term Ti. T_iCF is the category frequency of term T_i, and $CatTermRel_i$ is the category-term relevance for term T_i as explained in the previous section (Eq.5.2) of title relevance.

This formula has three parts. The first measures the importance of this term to the document itself. The second part represents the importance of the term as a classifier by its ratio between the number of document in which it occurs and the number of categories in which it occurs. If a term occurs in 10 documents and one category, then this term is a good classifier for this category and must be given a higher weight. If it occurs in 10 documents and 10 categories, then this term is not a good classifier for any category. The third part of the formula represents the relevance between the term T_i and the category C_j. This part is calculated the exact same way that the CatTermRel of the title relevance is calculated.

The following table (table 5) shows precision and recall using term classifier based on the previous formula at different filter coefficients.

Table 5: Precision and Recall based on Terms Relevance

Run	FilterCF	FilterTF	FilterDF	Retrieved	Precision	Recall
1	0.1	0.75	0.75	566	11.30	98.46
2	0.1	1.0	1.0	545	11.56	96.92
3	0.05	0.75	0.75	426	13.15	86.15
4	0.05	1.0	1.0	423	13.00	84.62
5	0.05	1.5	1.5	416	12.02	76.92

Analyzing this data shows that Terms Relevance technique achieves high recall values, but low precision. We conclude that our technique on terms can not be used alone because it results in very low precision.

We also tried another simpler formula that is based only on term frequency and category-term relevance without incorporating term general DF and CF values. The following table (table 6) shows precision and recall values for term relevance based on this formula.

$$R = \sum \left(\left(\frac{DT_i F}{DTL} \right) * \left(CatTerm\,\mathrm{Re}l_i \right) \right) \qquad (Eq.5.4)$$

Table 6: Precision and Recall based on Terms Relevance (formula 2)

Run	FilterCF	FilterTF	FilterDF	Retrieved	Precision	Recall
1	0.1	0.75	0.75	554	11.19	95.38
2	0.1	1.0	1.0	536	11.57	95.38
3	0.05	0.75	0.75	422	13.27	86.15
4	0.05	1.0	1.0	421	13.06	84.61
5	0.05	1.5	1.5	415	12.05	76.92

When we compare the results of the two formulas, we find out that they are very similar. The main reason for that is the effect of the filters of TDF and TCF. These filters make the effect of adding the relation between DF and CF negligible since they only select index terms in a very narrow spectrum of DF and CF. We expect this ratio to become more significant on larger test sets since the spectrum will be wider. Therefore we will consider the first results in our future analysis and comparisons.

5.2.4 Phrases Relevance:

Phrases relevance is also similar in concept to that of terms; moreover, it considers an extra piece of information.. A phrase is considered a significant phrase if its terms do not occur or rarely occur as independent terms. For example if a term occurs 100 times of which only 20 times within the phrase, then this phrase might not be a significant phrase and vice versa. Consider the phrase "Los Angeles". There is a very low probability that we find the term "Los" or the term "Angles" as separate terms, but whenever we find one of them we will find the other adjacent to it. This example shows how a significant phrase this is. On the other hand, consider the phrase "Vice President". We may find the term "Vice" in many other contexts and also the term "President"; therefore, this phrase although is a correct phrase is not as powerful as the first example. Another example is the phrase "Boy Football" in the sentence "That boy is a very good football player." These two terms may be found in many other contexts independently and when combined together, they do not build up a significant phrase. They just happen to be adjacent terms after stop words removal. The ratio between the frequency of the terms in the phrase and apart from the phrase gives us an indicator on the quality of this phrase.

We combined the relation between phrase terms with other factors such as the phrase frequency, DF, CF and category-phrase relevance values to calculate the relevance of the document d to category j. All index phrases are considered and their relevance scores are combined to give us one single value for relevance based on phrases.

For phrases we investigated two formulas for relevance similar to those explained for terms. The first is based on the phrase frequency in the document and the category-phrase relevance. The second formula added two extra factors: the relation between the phrase terms and the phrase ratio of DF to CF.

This first formula calculated relevance as follows:

$$R = \sum \left(\left(\frac{DPh_i F}{DPh_i L} \right) * \left(CatPhrase\,\mathrm{Re}l_i \right) \right) \qquad (Eq.5.5)$$

Where DPhF is the phrase frequency in the document, DPhL is the document length in phrases, and $CatPhraseRel_i$ is the relevance of phrase Ph_i to category C_j.

$$CatPhrase\,\mathrm{Re}l = \frac{\left(\frac{CatPhraseF * 100}{TCPhF} \right) * \left(\frac{CatPhraseDF * 100}{TCD} \right)}{100} \qquad (Eq.5.6)$$

Where CatPhraseF is the phrase frequency in all documents belong to this category, TCPhF is the total frequency for all phrases in this category, CatPhraseDF is the phrase Document Frequency in this category, and TCD is the total number of documents classified for this category.

By analogy to the second formula discussed above in term relevance, we shall ignore results of this formula and focus on the second formula, which is based on a more powerful logic.

The second formula we studied had the following form:

$$R = \sum \left(\left(\frac{DPh_i F}{DPh_i L} \right) * CatPhrase\,\mathrm{Re}l_i * \left(\frac{Ph_i DF}{Ph_i CF} \right) * \left(\frac{NPhT * PhF}{\sum PhTF} \right) \right) \qquad (Eq.5.7)$$

Where the first two parts of the formula are exactly like the previous formula. Ph_iDF and Ph_iCF are the document frequency and category frequency of phrase Ph_i

and NPhT is the number of terms in the phrase, PhF is the general phrase frequency, and ? PhTF is the summation of term frequencies for all phrase term.

The following table (table 7) describes the precision and recall vales obtained by only using phrase relevance second formula.

Table 7: Precision and Recall based on Phrases Relevance

Run	FilterCF	FilterPhF	FilterDF	Retrieved	Precision	Recall
1	0.1	0.75	0.75	31	83.87	40.00
2	0.1	1.0	1.0	25	84	32.30
3	0.05	0.75	0.75	29	89.66	40.00
4	0.05	1.0	1.0	22	90.90	30.77
5	0.05	1.5	1.5	20	95	29.22

Analyzing these results shows us that phrase relevance achieves very high precision categorization, but low to medium recall. This result by itself satisfies our first goal of achieving high precision categorization based on phrase indexing. In the next section, we will combine all results of the four techniques and do further analysis to these data trying to group different techniques to obtain satisfying results in both precision and recall.

5.3 Combining Techniques:

In the previous sections we have seen the independent performance of each categorization technique.

In this section will do the following:

1. We will compare our results for the four techniques we proposed.
2. We will investigate combining these techniques and analyze combined performance.
3. We will compare our phrase indexing technique with other researchers' results based on phrase indexing.

5.3.1 Comparing Individual Techniques:

In comparing our results for the four techniques, we will compare both evaluation scores for Precision and Recall. The following two graphs respectively show different precision (figure 4) and recall (figure 5) curves for the four techniques.

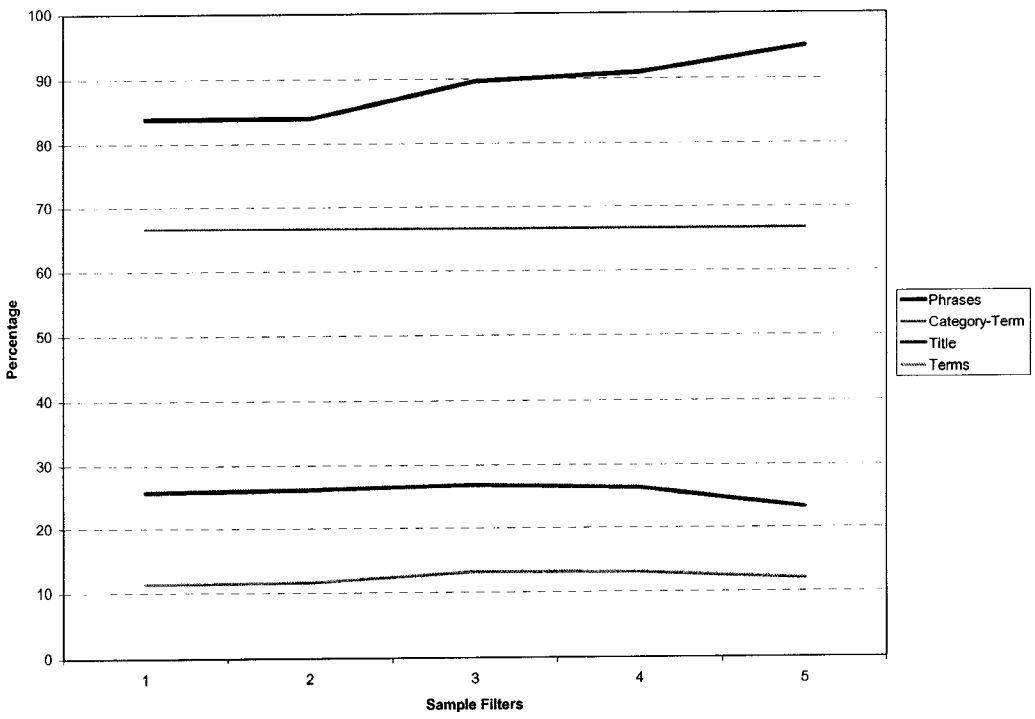

Figure 4: Individual Categorization Techniques Precision

As apparent in the precision graph, our phrase-based classifier achieves much higher precision values than the remaining three techniques at all sample filters used. This proves that our phrase classifier is a highly precise independent classifier. This result achieves the primary goal of this thesis. It also shows that category-term classifier is the second in descending order of precision followed by title-terms classifier, and finally term classifier. These results show that our first three proposed techniques achieve higher precision than the term-based technique. On the other hand, recall graph shows that term-based classifier achieves better recall values than the other three techniques. This suggests that a combination of categorization techniques is needed to achieve high precision while keeping a relatively high recall. Achieving high recall values is not the focus of our thesis, but we're still interested in achieving a reasonable recall.

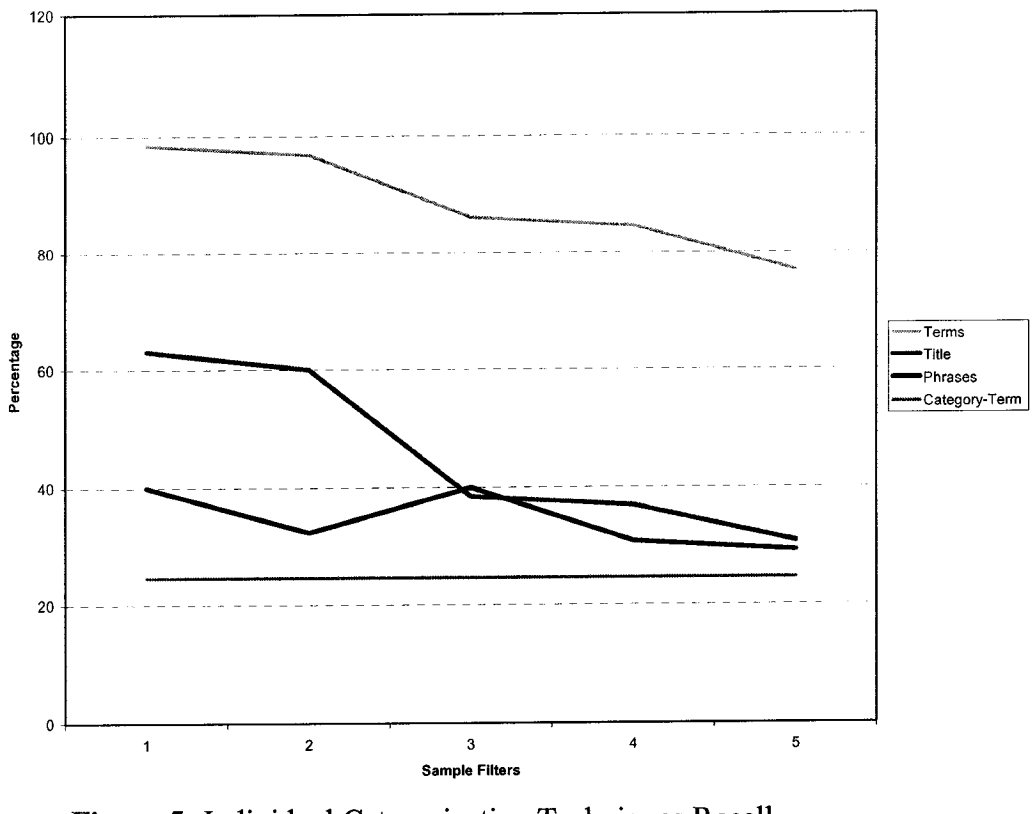

Figure 5: Individual Categorization Techniques Recall

5.3.2 Combining Techniques:

The present phrase-based classifier achieves the highest precision and will be our base upon which we will try to investigate adding different techniques to boost up recall values of the compound categorization technique. An interesting question arises here asking whether the result sets of these classifiers are independent or not. If they are totally independent, then combining techniques will increase recall values and will average precision. If one of them is totally contained in the other, then one of them might not be needed at all and the one with higher performance will be sufficient. It is important as well to study this behavior at different filter coefficient values to see if a distinct result set exists will always be distinct for all filters coefficients or not. The following table (table 8) shows total and newly added relevant documents by each technique. Techniques' results are compared in sequence starting by phrase classifier, then category-term classifier, then title classifier, and finally term classifier. The first column is the run number that describes filter coefficient values combination as explained in previous tables explaining each technique independent performance. The remaining 8 columns are two columns for each technique. The first describes the total

relevant number of documents retrieved by this technique while the second holds the number of new relevant documents retrieved by this specific technique after applying the previous techniques. Techniques are considered in sequence from left to right (Phrases, Category Term, Title, and then Terms Classifier).

Table 8: Comparing relevant document retrieved by categorization techniques

Run	Phrase Total	Phrase New	Cat-T Total	Cat-T New	Title Total	Title New	Terms Total	Terms New
1	26	26	16	15	41	13	64	11
2	21	21	16	16	39	10	63	17
3	26	26	16	15	25	11	56	8
4	20	20	16	16	24	10	55	11
5	19	19	16	16	20	8	50	13

In combining categorization techniques, phrase classifier is used as a base classifier for its high precision. Other classifiers are combined to it to increase its low recall value trading off as less precision as possible.

This table shows that each technique retrieves more relevant document when combined to the previous ones. On the other hand precision decreases dramatically. We will analyze these data trying to understand the different behaviors of these techniques and their behavior when combined. The goal is to figure out a good combination of techniques to keep high precision while increasing recall.

The first observation as mentioned before is that category-term classifier is independent of filters. The second observation is that each technique retrieves more relevant documents than the combination of all the previous techniques. On the other hand, precision is decreased as shown in results of independent techniques in the categorization attributes sub-sections. The third observation is that phrase classifier is almost independent of category-term classifier. The results show that these two classifiers retrieve almost distinct result sets. They slightly overlap when the number of indexing phrases increases, which is logic since phrase classifier retrieves more documents and the probability of overlap naturally increases. At the same time we do not expect this probability to be very high since the two classifiers are dependent on

two different bases. The first is based on phrases while the second is based on a few special terms.

The first observation suggests that category-term classifier is useful to be used in all cases but it might need some tuning to increase its precision. The second observation suggests that considering combining these techniques in the suggested order seems to be more logic since each technique add a certain recall value. At the same time tuning, weighting, and also filtering are needed to keep high precision. The third observation suggests that phrase classifier and category-term classifier might be useful to combine together to achieve better recall values while losing very little precision.

We also observe that precision results depending on different filter coefficient values are proportional to these values and as mentioned before recall values are inversely proportional to these filter coefficient values. We will choose a sample filter coefficient values of:

FilterCF = 0.05, FilterF = 1.0, and FilterDF = 1.0 to draw a set diagram for the relevant and retrieved documents by each technique and the relation between these result sets to better understand the behavior of each technique and expect the result of combining different techniques together. The following diagram (figure 6) shows the intersections of the result sets of the four techniques.

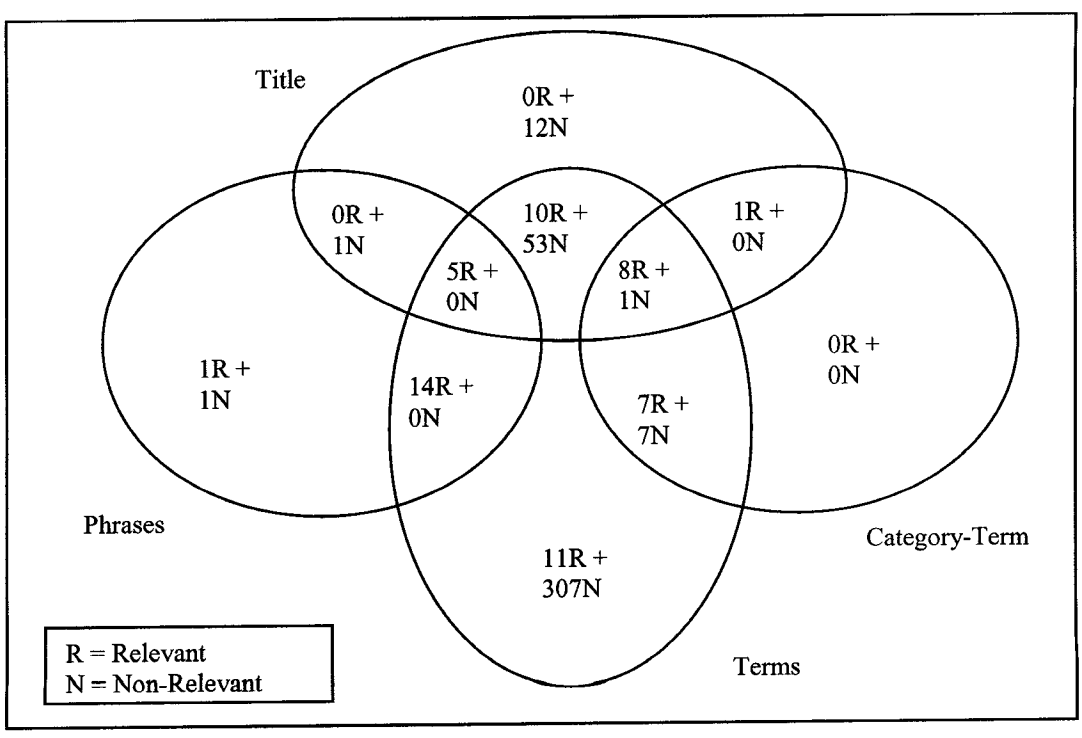

Figure 6: Results Sets of Categorization Techniques

In the above diagram R stands for relevant documents and N stands for non-relevant documents. The diagram makes it clear to see the behavior of the four techniques. We can grasp many pieces of information from this diagram that can help us to combine different techniques to achieve better performance. The following are a list of observations:

1. Phrase classifier result set is distinct from result set of category-term classifier.
2. Documents that are classified by more classifiers are more probable to be relevant.
3. Phrase classifier alone can achieve high precision.
4. Combining category-term and title classifiers results into highly precise result set.
5. Documents that are classified only by title classifier are not relevant.
6. Result set of combining both title and term classifiers have much higher precision than the result set that contains document classified only by terms classifier.

These observations lead to some conclusions:

1. We can use phrase classifier as a boolean categorization method.
2. We can consider those documents co-classified by category-term and title classifiers.
3. Title classifier achieves better precision than term classifier and can replace it to achieve better precision while keeping almost same recall.
4. Term-based techniques can be enhanced for both title and terms classifiers then combined with the compound method of phrase and category-term classifiers.

These conclusions are based on the test set we investigated and considering all used classifiers as boolean classifiers. We do not know if these conclusions will be confirmed when using different or larger test sets. Therefore, we need to confirm our conclusions using different and larger test set. We also need to study relevance scores because we might need to use relevance scores to enhance performance instead of using boolean categorizations.

5.3.3 Relevance Scores and Larger Test Data Size:

In the previous section we discussed combining different techniques based on boolean results. The results showed us that we can use phrase classifier and the intersection of results of category-term and title classifiers as boolean classifiers. In this section we will analyze relevance scores obtained by only phrase categorization since it is our main focus in the present work as an example to the scoring function and to see how scoring can enhance the results for other techniques as well. We will study scores on the same data set analyzed in the previous section. The following graph (figure 7) shows precision curve at different relevance scores based only on phrase categorization.

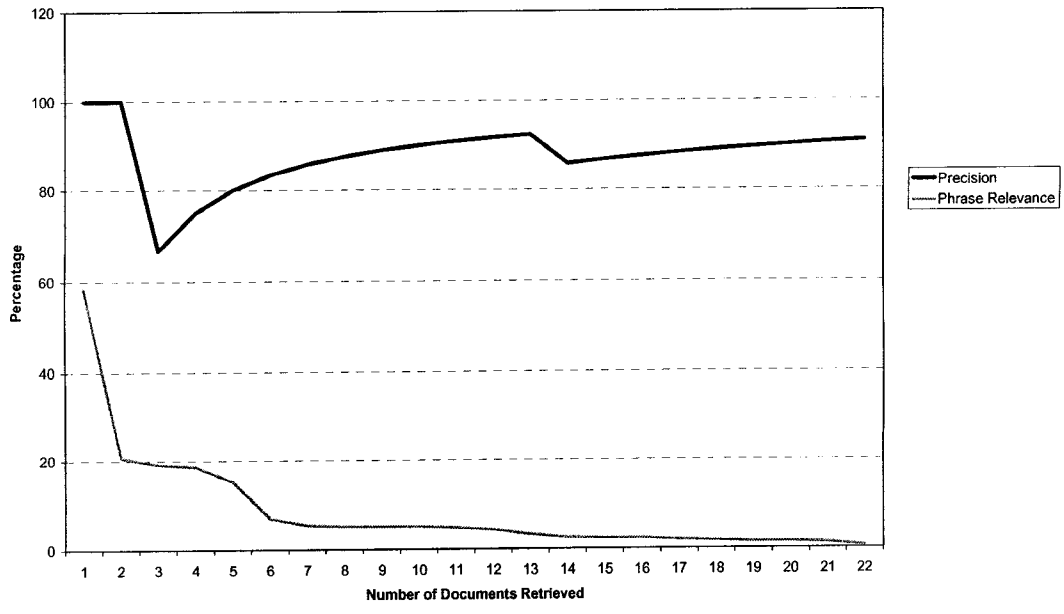

Figure 7: Phrase Classifier Relevance Scores (Test Data 1)

The graph shows precision curve against phrase relevance scores curve. A trend is not very clear in this graph. This is expected to be due to small data set. In order to investigate this observation further, we try phrase categorization over larger training and test sets. We will also run experiments on this new test set to investigate whether our methodology will scale up with larger test data size or not. We used a training set of 600 documents and a test set of 300 documents. The following graph (figure 8) shows precision and scores using the new sets.

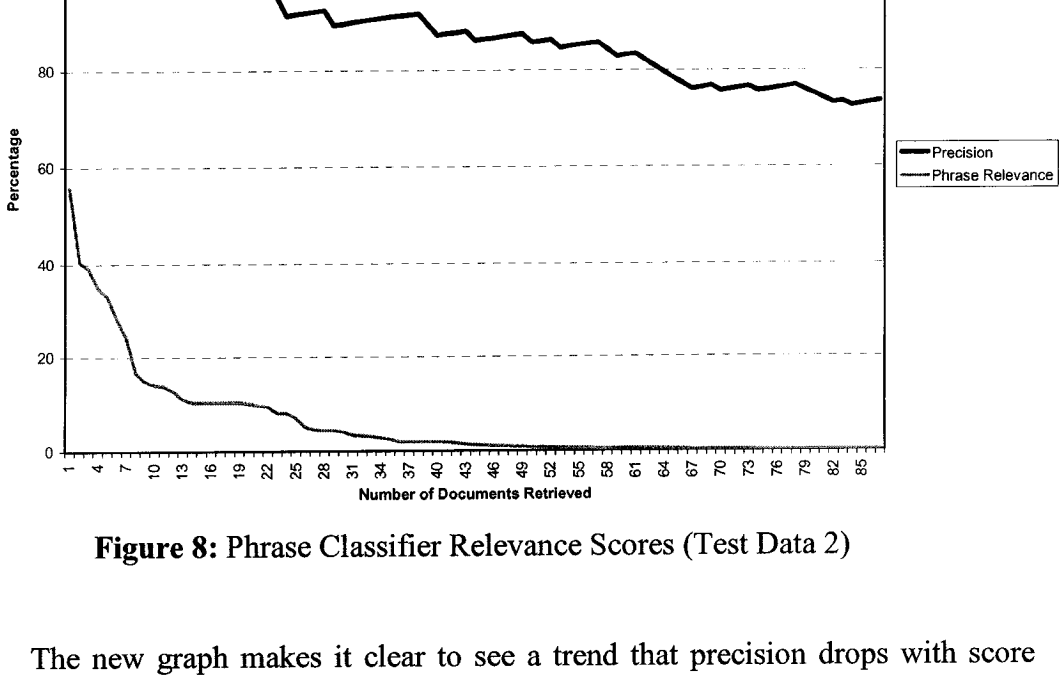

Figure 8: Phrase Classifier Relevance Scores (Test Data 2)

The new graph makes it clear to see a trend that precision drops with score dropping. This observation suggests that precision is higher at higher relevance scores. It also suggests that precision might decrease with larger data sets. This observation if confirmed might initiate a need for a threshold to be used to filter out low-score documents. To be able to set a threshold, larger data sets must be tested to be able to select a valid threshold value.

These results show us that combining techniques together just like we proposed before might be useful to enhance performance of phrase classifier as well. We can simply investigate the idea of combining techniques by using the set diagram just like we did with the original test data.

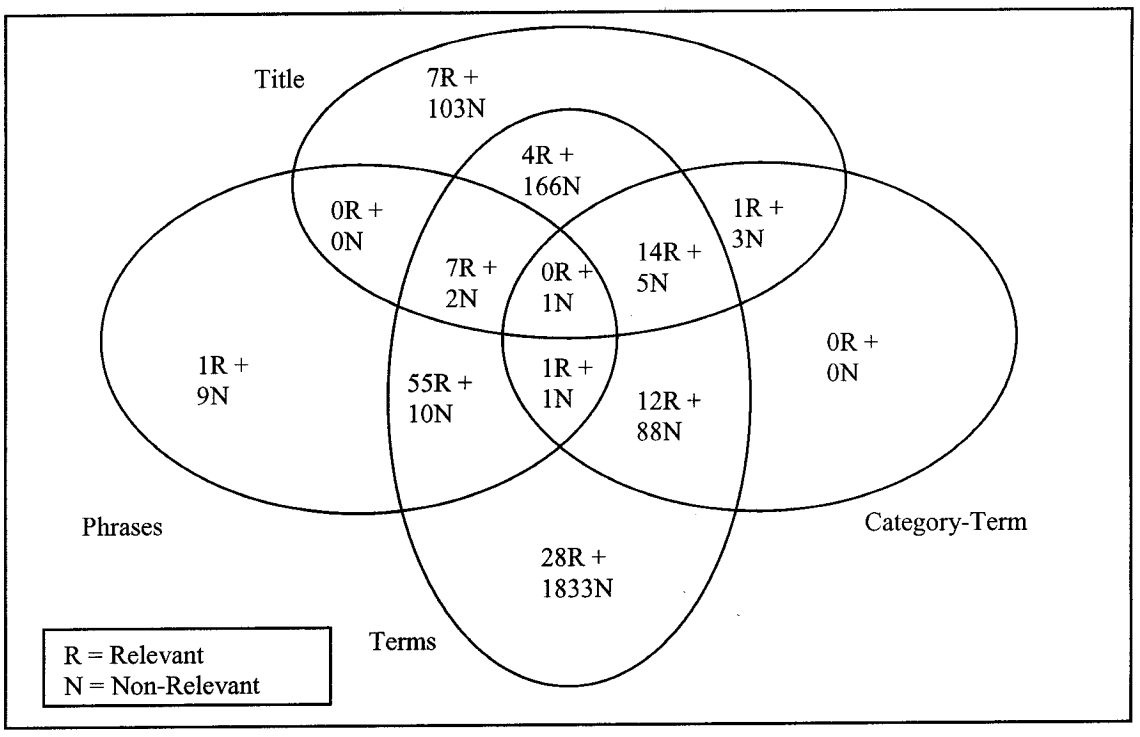

Figure 9: Set Diagram for categorization techniques (Test Data 2)

The set diagram for this larger data sets size (Figure 9) shows clearly the relation between result sets of different techniques and confirms our initial results. It shows that combining techniques result into better performance. All observations confirm our initial results except for one. These results correct our initial results of using phrase classifier as a boolean classifier. It suggests that combining it with other techniques is more useful for larger data sizes and indicates the importance of relevance scores.

5.4 Related Work:

A few researchers investigated the usefulness of phrase indexing in IR and TC. In this section we will compare other work done by other researchers on IR or TC using phrase classifiers alone or combined with other techniques. We will compare their results to ours to understand where do we stand from other research in this area.

Fujita [26] reported performance results for three techniques, corresponding to automatic short query (JSCB1), automatic long query (JSCB2), and manual (JSCB3). Experiments were designed to measure effects of phrasal term indexing in IR regarding different query length and different weighting. Results did not achieve high

precision. For example, Results for automatic short queries (JSCB1) achieved precision values ranging between 8% and 70% for recall values ranging between 100% and 0% respectively.

David D. Lewis [8] investigated the effect of representation quality in text categorization. In his experiments he measured precision and recall values using terms, terms and phrases, and terms and clusters of different sizes. His results for terms and phrases showed precision values between 4.9% and 61.1% at recall values between 100% and 10% respectively.

Carporeso, Matwin, and Sebastiani [20] report their research results on statistical phrases in automatic text categorization. They used a subset of Reuters – 21578 in their research without stemming and using a different stop words list. They investigated several feature evaluation function (FEF) including document frequency, information gain, chi-square, and odds ratio. They proposed a classifier-independent evaluation technique to measure the usefulness of statistical phrases in TC. Their results using bi-grams (2-word-phrases) for different FEF at different filtering values had precision values between 47% and 79% and recall values between 26% and 69% [20]. The following table (table 9) compares precision and recall using a sample FEF (Document Frequency) that they used to our results at relatively similar feature reduction factors. Each cell will contain a precision value @ recall value / feature reduction factor. We think the superiority in our precision values are referred to the accuracy of the phrase classifier formula while the recall values differences resulted from the selection of index features by the feature reduction function. Combining other techniques to the phrase classifier is also expected to enhance the precision even more. Changing coefficients of the feature reduction function is expected to enhance the recall values while keeping same or similar precisions.

Lewis [11] studied the effect of syntactic phrase indexing in TC and reported significantly lower effectiveness than standard "set-of-words" indexing. Dumais et al. [29] reported no benefits at all from the use of syntactic phrases with several text classifiers using Reuters – 21578 collection. Furnkranz et al. [20] showed that syntactic phrases yield precision improvements at low recall levels. Mladenie and Grobelnik extracted n-grams of length up to 5 using a fast algorithm that relies on a statistical filter and found that n-grams of length up to 4 give significant benefits with respect to the single word case. 5-grams do not provide additional benefits. Funkranz [20] extracted n-grams of length up to 5 on Reuters – 21578 and found significant

improvements using bi-grams while longer n-grams reduce categorization performance. He also tried the statistical technique used on another collection and found that 3-grams was useful, while longer n-grams are confirmed to reduce performance.

Mitra, Buckley, Singhal, and Cardie [19] applied different techniques on syntactic and statistical phrases and achieved precision value that ranges between 0.76% and 85% at recall levels ranging between 100 % and 0% respectively using different techniques and phrase factor values.

Table 9: Comparing effectiveness to related work based on phrase indexing

Related Work **Precision @ Recall / Reduction Factor**	**Our Phrase Classifier** **Precision @ Recall / Reduction Factor**
78.8% @ 68.3% / 97.40%	66.67% @ 36.92 / 97.52%
78.8% @ 68.3% / 98.05%	87.10% @ 41.54% / 98.15%
78.5% @ 68.1% / 98.70%	84% @ 32.31% / 98.51%
77.2% @ 66.9% / 99.35%	90.90% @ 30.77% / 99.34%

5.5 Summary:

This chapter presented the proposed four categorization techniques, phrase classifier, category-term classifier, title classifier, and term classifier. It explained how each technique considered a document relevant to a category and how it calculates the relevance score. We compared between each independent technique and found out that phrase classifier achieved high precision value and medium recall values. We also investigated combining different techniques together to achieve better recall while keeping high precision. To figure out a trend for relevance scores produced by phrase classifier we investigated the application of the classifier on a larger test set and using larger training set. The experiment on larger test set confirmed our results from the original experiment that our phrase classifier is a highly precise classifier. For example, our phrase classifier achieved precision value of 91% at recall level of 31% when applied on the original test set using filter coefficient combination of CF = 5% , PhF = 100%, and PhDF = 100%. The same classifier settings achieved precision value of 74% at recall level 46% when applied on a larger test set using the same filter coefficient combination sample. These results initiated the need for using threshold to

filter out low relevance-scores retrieved documents. It also confirmed the benefits achieved by combining techniques together. Results showed that considering only intersection of phrase classifier result set and result sets of the other three other proposed classifiers, we enhanced precision to 82% while keep recall at 45%. Finally we compared our phrase classifier results with other phrase-based classifiers by other researchers and this comparison showed that our classifier achieved better precision.

Precision and recall values for category-term and title classifiers show better precision than term classifier. Categorization based on category-term achieved 67% precision and does not include computational costs such as term classifier that achieved an average of 12% precision. Title classifier also requires less computational costs and produce average precision of 26%.

Chapter 6

Conclusion and Future Work

Chapter 6
Conclusion and Future Work

6.1 Conclusion:

The modern technological advances and huge increase in Internet usage enabled the modern world to share information in much easier ways. The colossal increase in shared information created the need for automatic and powerful tools to retrieve information to satisfy user needs. The Information Retrieval (IR) field of research is interested in providing techniques and models to be used to build such systems. There are many areas of research in this field. IR systems mainly consist of two processes. The first is categorization of documents and the second is retrieving a subset of these classified documents to match a user need represented by a query. All areas of research in IR are investigating parts of the complete process or applications of these parts.

Text Categorization (TC) is one of the areas of research in IR field. TC is concerned with categorizing text documents to a set of pre-defined categories. The TC problem is very similar to the IR problem. If we think of a category as a query we will find TC is trying to retrieve all documents relevant to that specific category just like IR will try to retrieve all documents relevant to a query. Most techniques that are used for IR are also used for TC and vise versa.

We have reviewed in this thesis previous efforts and research done in the IR and TC fields. Standard test collections are used for research purposes in IR and TC. Reuters – 21578 is one of the most important collections used for that purpose. It is considered the standard collection for TC research. We have selected this collection and used a subset of it to conduct our research. Early developments in IR field included full-text scanning, signature files and inversion. Evaluation methods are developed to evaluate IR systems. They evaluate many aspects in systems including effectiveness, efficiency, user interface, and many other aspects, the most important aspect of which is effectiveness. Many effectiveness evaluation techniques are proposed and used. The most widely used performance evaluation technique is to calculate precision and recall of the system. Precision is the ratio between the number of retrieved documents that are relevant and the total number of retrieved documents. Recall is defined as the ratio between the number of documents retrieved that are

relevant and the total number of relevant documents in the collection. Text REtrieval Conference (TREC) was established in 1991 to evaluate large-scale IR systems. It is considered the most well known evaluation setup in the IR field.

Any IR or TC system usually performs some or all of pre-processing steps to documents before starting the categorization process. These steps include parsing, tokenizing, stop-word removal, and stemming. Parsing recognizes the document structure. Tokenizing identifies tokens such as numbers and special words. The output of tokenizing process is a stream of words. Stop-word removal process removes all elements of a stop-word list from the text stream of the document. This stop-word list contains words that are of low discrimination quality. Mostly they include functional words and other low-content bearing words. Stemming is the process of removing or adding a word prefix or suffix to obtain the word root. The final output of these pre-processing steps is a stream of stems. They are candidates to be used for indexing and categorization processes. Feature reduction techniques are developed to reduce the number of terms to be used for indexing. These techniques include Inverse Document Frequency, Information Gain, Chi-Square, and Odds Ratio. Natural Language Processing techniques were also developed for feature reduction and categorization and they include Principal Component Analysis (PCA) and Latent Semantic Indexing (LSI).

Exact and best match models are used to decide on documents' relevance to queries. Exact match uses boolean operations to decide on relevance. Best match uses scores and the result set contains the highest scoring *n* relevant documents to a user query. Vector Space Model (VSM) is one of the most widely used retrieval models. It treats both documents and queries in the same way. They are represented in the form of vectors in a multi-dimensional space. The angel between a document and a query vectors represent the relevance between them. Clustering concepts are used to enhance performance of IR systems. Different techniques are developed to generate term clusters, document clusters, and phrase clusters. Relevance feedback used the feedback of users on the initial retrieved result set to enhance the query and perform another search iteration to enhance the quality of the result set. Techniques such as phrase modeling and query expansion are used to enhance the performance of VSM. Neural Network techniques were developed to solve the IR problem such as applying LSI using a neural back-propagation network. Many categorization methods were

developed and used such as discrimination analysis, logistic regression, Optimal Separating Hyper-planes, and Classification Trees.

Most of the TC methods rely on terms (single words) as basic features for categorization. However, a phrase carries more meaning than a single word and thus it is logic to consider it a better categorization feature than a term. Although previous research did not confirm such hypothesis, various results suggested that phrases can be used to enhance performance, but can not be used alone for categorization. Phrases in general have two types: statistical and syntactic. Statistical phrases are based on co-occurrence of terms while syntactic phrases are based on certain linguistic rules. Statistical phrase classifiers in general achieve better results than syntactic phrase classifiers. Different proximity values are used to construct statistical phrases. Phrase terms of proximity = 0 are called adjacent phrases. Phrase proximity and the number of terms that build up a phrase are two important factors that affect greatly phrase categorization techniques. Previous research results show us that adjacent phrases achieve higher precision than higher proximity phrases. They also show us that two terms phrases (also called bi-grams) achieve relatively good results in general. The number of initial phrases identified by any system is huge and needs reduction to select the highest quality ones to be considered as features and used for indexing. Feature Evaluation Functions (FEF) are used to reduce phrase features such as those used for term features.

The primary goal of this thesis is to prove that phrases could be used to obtain a highly precise text categorization system. We used statistical adjacent phrases of size two. We proposed a feature reduction function for both terms and phrases based on three components. The first is the category frequency (CF) of the feature. We proposed this function on the grounds that occurrence of a feature in different categories is an important factor to identify whether a certain feature is a good discriminating feature between categories or not. The second component is the feature frequency. The third component is the document frequency (DF) of the feature. Results achieved by this feature reduction function were promising. This function was able to dramatically decrease the initial feature lists and achieve good performance when using the resulting features by our classification functions. For example, our feature reduction function achieved over 98% reduction for terms and over 99% reduction for phrases at a given sample of function coefficients.

We proposed a phrase classifier function based on frequency information. It calculates relevance score based on phrases only by combining several components together. The first component calculates the importance of a phrase for the document. The second calculates the significance of a phrase giving less weight to noise phrases that were not eliminated by the feature reduction function. The third component calculates the importance of a phrase to the category to which relevance is being studied. This actually represents the phrase weight representing its relevance to that category. The last component assigns a higher weight for more significant phrases by considering the ratio between phrase frequency and phrase terms' individual frequencies. The results of this methodology achieved a high precision value at medium recall values. Using a sample filter coefficient combination, phrase classifier achieved 91% precision at 34% recall using the original test set. At the same filter coefficient combination, phrase classifier achieved 74% precision at 46% recall and enhanced by combining other techniques to 81% precision at 45% recall. These results prove that phrases can be used as a primary classifier to achieve highly precise categorization. Other experiments on larger test set and using larger training set confirmed our initial results that phrases can achieve high precision and suggested that other techniques can be used to enhance recall.

Our secondary goal of this thesis was to propose other term-based techniques of less computational costs and achieve at least as precise as usual term classifiers. We proposed those to provide other aiding techniques to be used to enhance phrase classifier performance. We proposed a category term classifier, which considers the existence of a category name in a document an indication of relevance of that document to that specific category. Category term classifier achieved good precision and recall values. Category term classifier achieved 67% precision and 37% recall.

Title classifier is our third proposed classifier. Title classifier gives higher weight to index terms found in the title. Title-term relevance to a category is based on the importance of this term to a category and calculated using the term frequency and DF in the category relative to the total terms frequencies in the category and the total number of documents classified to that category. Title classifier achieved an average precision of 26% and average recall of 46% when averaging over a set of filter coefficient combination of 5 samples.

We applied the same formula that is used for phrases on terms and produced results for term classifier. We needed term classifier on the same testing environment

to compare other techniques' results and also to be able to understand the relation between these techniques in a better way. Term classifier achieved an average precision of 12% and an average recall of 89% when averaging over the same filter coefficient combination samples mentioned above.

By comparing and combining these techniques we found out that relevance could be decided using several classifiers together to produce higher performance. We found out that phrase classifier achieves the best precision and combining other classifiers can enhance its recall while keeping high precision.

Finally, it was quite apparent that set diagrams were very useful to analyze result sets of different techniques. This tool helped us great deal in comparing different techniques and studying the effect of merging techniques together. Therefore, we would like to emphasize foreseen benefits by using this analysis tool in IR and TC research.

6.2 Future Work:

Our proposed techniques for feature reduction and categorization introduces several interesting directions for future work. We introduced a new FEF based on Category Frequency, Feature Frequency, and Feature Document Frequency. We also introduced a new categorization rule and applied it on both phrases and terms. We also introduced two new classifiers based on Categories names and Title Terms.

Finding out the proper combination and coefficients of the feature reduction formula is a direction that needs a lot of research. Data analysis can be applied to larger training sets on several test collections using several coefficient values to be able to choose the best coefficient values for this function.

Another direction of research is to study the behavior of the proposed classifiers on larger collections. We expect precision to slightly decrease while increasing recall values.

Similar to introducing techniques based on special parts of a document that seem to be more effective in categorization such as document title, future work could continue investigating other document sections such as references, abstract, and conclusion. However, these new techniques must be tested using test collections documents that contain these sections. Reuters – 21578 can not be used for that purpose because it includes only small-length news articles.

Building upon the previous point of working on other parts or sections of the document, working with more structured formats such as XML will be easier than free text and is expected to speed up research in this direction since simpler parsers will be needed.

Test phrase-based categorization using other stop lists might introduce different phrases that might be candidate for indexing features. This might affect precision and therefore is a good direction of further research. The same direction of changing list of identified phrases can be achieved by investigating the process of identification of phrases before stop-word removal. Keeping the relations between words implied by function words before phrase identification can introduce new techniques for phrase identification, which is expected to achieve better precision with the introduction of higher processing complexity.

Building upon title terms relevance, we can also try title phrase relevance to identify index phrases in titles and assign higher scores for them. By analyzing result sets of title classifier, we notice a great dependency between title terms and terms classifiers. This observation makes us expect to achieve higher precision by title phrase classifier.

The proposed categorization techniques can also be applied to other languages other than English language. We expect the proposed techniques to achieve similar results when used with other languages because they are language-independent.

The suggested directions for future work focus on pursuing research in the same direction to obtain a highly precise TC model. Other directions can be also investigated building on the proposed techniques such as applying these techniques for the IR problem and investigating the proposed phrase-based categorization method using syntactic instead of statistical phrases.

References

1. Chinatsu Horii, Masakazu Imai, and Kunihiro Chihara: **Conceptual Information Retrieval of Technical Papers for Digital Libraries.** IEEE Forum on Research and Technology Advances in Digital Libraries, March 19 - 21, 1999 Baltimore, Maryland P.171
2. Christos Faltoutsos, and Douglas Oard: **A Survey of Information Retrieval Filtering Methods.** Technical Reports CS-TR 3514, Dept of Computer Science, University of Maryland, 1995.
3. C. J. Van Rijsbergen: **"Information Retrieval."** Butterworth, London, England, 1979. 2nd edition.
4. C. T. Yu, K. Lam, and G. Salton. **Term Weighting in Information Retrieval Using the Term Precision Model.** *JACM, Jan 1982.*
5. David A. Hull. **Information Retrieval Using Statistical Classification.** PhD Thesis, Stanford University, November 1994.
6. David D. Lewis and W. Bruce Croft. **Term Clustering of Syntactic Phrases.** *In Proceedings of SIGIR-90, 13th ACM International Conference on Research and Development in Information Retrieval, pages 385*{404, Bruxelles, BE, 1990).
7. David Lewis and Marc Ringuette. **A Comparison for Two Learning Algorithms for Text Categorization.** In Symposium on Document Analysis and Information Retrieval. University of Nevada, Las Vegas, 1994.
8. David D. Lewis. **Representation Quality in Text Classification: An Introduction and Experiment.** *Morgan Kaufmann, San Mateo, CA, pages 288 – 295, 1990.*
9. David D. Lewis. Reuters – 21578 Test Collection http://www.research.att.com/~lewis/reuters21578.html
10. Defense Technical Information Center / Defense RDT&E Online System (DTIC-DROLS) and Standard Verity Stop List. http://dvl.dtic.mil/stop_list.pdf
11. D. D. Lewis. **An evaluation of phrasal and clustered representations on a text categorization task.** In N. J. Belkin, P. Ingwersen, and A. M. Pejtersen, editors, *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval, pages 37*{50, Kobenhavn, DK, 1992. ACM Press, New York, US.
12. G. Salton. **Recent Studies in Automatic Text Analysis and Document Retrieval.** *JACM 20(2): 258-278, April 1973.*
13. G. Salton. **The SMART Retrieval System – Experiments in Automatic Document Processing.** Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1971.
14. Gerard Salton. **The State of Retrieval System Evaluation**. Information Processing and Management. 28(4):441-449, 1992.
15. James Allan: **Information Retrieval.** Course Material - University of Massachusetts, Amherst. Course CMPSCI 646, Fall 2002.
16. Jeremy Pickens and W. Bruce Croft. **An Exploratory Analysis of Phrases in Text Retrieval.** In RIAO' 2000 Content-based Multimedia Information Access. Vol.1, pages 1179 - 1195, College de France, Paris, France, 2000.
17. Joel L. Fagan. **Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods.** PDH Thesis, Department of Computer Science, Cornell University, 1987.

18. L. Breiman, J. Friedman, R. Olshen, and C. Stone. **Classification and Regression Trees.** Wadsworth and Brooks, 1984.
19. Mandar Mitra, Chris Buckley, Amit Singhal , and Claire Cardie. **An Analysis of Statistical and Syntactic Phrases.** *In Proceedings of RIAO-97, 5th International Conference \ Recherche d'Information Assistee par Ordinateur, pages 200*{214, Montreal, CA, 1997).
20. Maria Fernanda Caropreso, Stan Matwin, and Fabrizio Sebastiani. **A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization.** In Amita G. Chin (ed.), Text Databases and Document Management: Theory and Practice, Idea Group Publishing, Hershey, US, 2001, pp. 78-102.
21. Marie-Francine Moens. **Current challenges and prospects of text retrieval.** Interdisciplinary Center for Law & IT K.U.Leuven, Belgium. Sep, 2000.
22. M.F. Porter. **An Algorithm for Suffix Stripping.** Program, 14(3), 130 - 137, Porter 1980.
23. Ricardo Baeza-Yates, Berthier Ribeiro-Neto. **Modern Information Retrieval.** Addison-Wesley-Longman Publishing co. 1999
24. R. Tong, A. Winkler, and P. Gage. **Classification Trees for Document Routing.** *In Proceedings of TREC-1, pages 209 – 228, 1993.*
25. Scott Weiss: "**Glossary for Information Retrieval.**" Jan, 1997. http://web.engr.oregonstate.edu/~jung/research.htm
26. Sumio FUJITA. **Notes on Phrasal Indexing.** JSCB Evaluation Experiments at NTCIR AD HOC, NTCIR Workshop 1, Tokyo 101-108, 1999. JUSTSYSTEM Corporation. Brains park, Tokushima, JAPAN.
27. Sumio FUJITA. **Reflections on "Aboutness ".** TREC-9 Evaluation Experiments at Justsystem, TREC 2000. JUSTSYSTEM Corporation. Brains park, Tokushima, JAPAN.
28. S.E. Robertson and K. Sparck Jones. **Simple, proven approaches to text retrieval.** Department of Information Science, City University, UK & Computer Laboratory, University of Cambridge, UK. May, 1997
29. S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. **Inductive learning algorithms and representations for text categorization**. In G. Gardarin, J. C. French, N. Pissinou, K. Makki, and L. Bouganim, editors, Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management, pages 148{155, Bethesda, US, 1998. ACM Press, New York, US.
30. Text Retrieval Conference (TREC) Web Site. http://trec.nist.gov/
31. Tomek Strzalkowski. **Robust Text Processing in Automated Information Retrieval.** Proceedings of the Fourth ACL Conference on Applied Natural Language Processing (13-15 October 1994, Stuttgart)
32. Thomas Mandl. **Efficient Pre-Processing for Information Retrieval with Neural Network.** In Zimmermann, Hans-Jurgen (1999) (ed.): EUFIT 99. 7th European Congress on Intelligent Techniques and Soft Computing." Aachen, Germany, 13 – 16. September 1999.
33. T. Arampatzis, T. Tsoris, C.H.A. Koster, TH. P. van der Weide. **Phrase-Based Information Retrieval.** Information Processing & Management, 34(6):693-707, December 1998.
34. Wessel Kraaij and Ren´ee Pohlmann. **Viewing Stemming as Recall Enhancement.** Revised version of Proceedings of SIGIR 1996.

35. W. Bruce Croft. **What Do People Want from Information Retrieval? (The Top 10 Research Issues for Companies that Use and Sell IR Systems).** Center for Intelligent Information Retrieval. Department of Computer Science. University of Massachusetts, Amherst. D-Lib Magazine, November 1995
36. W. Bruce Croft, Howard R. Turtle, and David D. Lewis. **The use of Phrases and Structured Queries in Information Retrieval.** *Proc. Of ACM SIGIR, pages 32 – 45, October 1991.*
37. W.B. Croft. **A Model of Cluster Searching Based on Classification.** *Information Systems, 5: 189 – 195, 1980.*
38. W.S. Cooper. **On Deriving Design Equations for Information Retrieval Systems.** *JASIS, pages 385 – 395, November 1970.*
39. Yiming Yong and Jan O. Pederson. **A Comparative Study on Feature Selection in Text Categorization.** In D. H. Fisher, editor, Proceedings of ICML-97, 14th International Conference on Machine Learning, pages 412{420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

Glossary

This glossary of terms is based on glossary written by Scott Weiss [25] and we added some terms to it for terminology used in this thesis..

- **Boolean query**
  A query that is a Boolean combination of terms. Some examples are INFORMATION AND RETRIEVAL, VISION OR SIGHT, and CLINTON AND (NOT GORE).
- **Classification**
  The process of deciding the appropriate category for a given document. Examples are deciding what newsgroup an article belongs in, what folder an email message should be directed to, or what is the general topic of an essay.
- **Cluster**
  A grouping of representations of similar documents. In a vector space model, one can perform retrieval by comparing a query vector with the centroids of clusters. One can continue search in those clusters that are in this way most promising.
- **Collaborative Filtering**
  The process of filtering documents by determining what documents other users with similar interests and/or needs found relevant. Also called "social filtering".
- **Collection**
  A group of documents from which a user wishes to obtain information. See also test collection.
- **Collection Fusion**
  The problem of combining the search results from multiple collections. This could be tricky since some measures such as IDF will differ across collections, and, if one retrieves a fixed number of documents, it is unclear how many to take from each collection.
- **Content-Based Filtering**
  The process of filtering by extracting features from the text of documents to determine the documents' relevance. Also called "cognitive filtering".
- **Cosine Similarity**
  See similarity.
- **Document**
  A piece of information the user may want to retrieve. This could be a text file, a WWW page, a newsgroup posting, a picture, or a sentence from a book.
- **Indexing**
  The process of converting a collection into a form suitable for easy search and retrieval.
- **Information Extraction**
  A related area that attempts to identify semantic structure and other specific types of information from unrestricted text.
- **Information Filtering**
  Given a large amount of data, return the data that the user wants to see. This is the standard problem in IR.

- **Information Need**
  What the user really wants to know. A query is an approximation to the information need.
- **Information Retrieval**
  The study of systems for indexing, searching, and recalling data, particularly text or other unstructured forms.
- **Inverse Document Frequency**
  Abbreviated as IDF, this is a measure of how often a particular term appears across all of the documents in a collection. It is usually defined as log(collection size/number of documents containing the term). So common words will have low IDF and words unique to a document will have high IDF. This is typically used for weighting the parameters of a model.
- **Inverted File**
  A representation for a collection that is essentially an index. For each word or term that appears in the collection, an inverted file lists each document where it appears. This representation is especially useful for performing Boolean queries.
- **Phrase**
  A Phrase is a group of two or more words that occur frequently in a collection or a corpus. In general there are two types of phrases: syntactic and statistical.
- **Phrase Indexing**
  Phrase Indexing is the process of constructing an index based on phrases' occurrences in documents.
- **Precision**
  A standard measure of IR performance, precision is defined as the number of relevant documents retrieved divided by the total number of documents retrieved. For example, suppose there are 80 documents relevant to widgets in the collection. System X returns 60 documents, 40 of which are about widgets. Then X's precision is 40/60 = 67%. In an ideal world, precision is 100%. Since this is easy to achieve (by returning just one document), a system attempts to maximize both precision and recall simultaneously.
- **Pre-coordination of terms**
  The process of using compound terms to describe a document. For example, this page may be indexed under the term "information retrieval glossary".
- **Post-coordination of terms**
  The process of using single terms to describe a document, which are then combined (or coordinated) based on a given query. For example, this page may be indexed under the words INFORMATION, RETRIEVAL, and GLOSSARY. We'd then have to combine these terms based on a query like "INFORMATION and RETRIEVAL".
- **Probabilistic Model**
  Any model that considers the probability that a term or concept appears in a document, or that a document satisfies the information need. A Bayesian inference net is a good framework for this style of model. The INQUERY system is the most successful example.
- **Query**
  A string of words that characterizes the information that the user seeks. Note that this does not have to be an English language question.

- **Query Expansion**
  Any process which builds a new query from an old one. It could be created by adding terms from other documents, as in relevance feedback, or by adding synonyms of terms in the query (as found in a thesaurus).
- **Question Answering**
  The problem of finding the exact answer to a user's natural language question in a large collection.
- **Recall**
  A standard measure of IR performance, recall is defined as the number of relevant documents retrieved divided by the total number of relevant documents in the collection. For example, suppose there are 80 documents relevant to widgets in the collection. System X returns 60 documents, 40 of which are about widgets. Then X's recall is 40/80 = 50%. In an ideal world, recall is 100%. However, since this is trivial to achieve (by retrieving *all* of the documents), a system attempts to maximize both recall and precision simultaneously.
- **Relevance**
  An abstract measure of how well a document satisfies the user's information need. Ideally, your system should retrieve all of the relevant documents for you. Unfortunately, this is a subjective notion and difficult to quantify.
- **Relevance Feedback**
  A process of refining the results of a retrieval, using a given query. The user indicates which documents from those returned are most relevant to his query. The system typically tries to find terms common to that subset, and adds them to the old query. It then returns more documents using the revised query. This can be repeated as often as desired. Also called "find similar documents" or "query by example".
- **Robot**
  See spider.
- **Routing**
  Similar to information filtering, the problem of retrieving wanted data from a continuous stream of incoming information (i.e. long-term filtering).
- **SIGIR**
  The ACM's special interest group on Information Retrieval. They publish SIGIR Forum and have an annual conference.
- **Signature File**
  A representation of a collection where documents are hashed to a bit string. This is essentially a compression technique to permit faster searching.
- **Similarity**
  The measure of how two documents are alike or how a document and a query are alike. In a vector space model, this is usually interpreted as how close their corresponding vector representations are to each other. A popular method is to compute the cosine of the angle between the vectors.
- **Spider**
  Also called a robot, a program that scans the web looking for URLs. It is started at a particular web page, and then accesses all the links from it. In this manner, it traverses the graph formed by the WWW. It can record information about those servers for the creation of an index or search facility. Most search engines are created using spiders. The problem with them is, if not written

properly, they can make a large number of hits on a server in a short space of time, causing the system's performance to decay.

- **Stemming**
  The process of removing or adding prefixes and suffixes from words in a document or query in the formation of terms in the system's internal model. This is done to group words that have the same conceptual meaning, such as WALK, WALKED, WALKER, and WALKING. Hence the user doesn't have to be so specific in a query. The Porter stemmer is a well-known algorithm for this task.
- **Stop-word**
  A word such as a preposition or article that has little semantic content. It also refers to words that have a high frequency across a collection. Since stop-words appear in many documents, and are thus not helpful for retrieval, these terms are usually removed from the internal model of a document or query. Some systems have a predetermined list of stop-words. However, stop-words could depend on context. The word COMPUTER would probably be a stop-word in a collection of computer science journal articles, but not in a collection of articles from Consumer Reports.
- **Term**
  A single word or concept that occurs in a model for a document or query. It can also refer to words in the original text.
- **Term Frequency**
  Abbreviated as TF, the number of times a particular term occurs in a given document or query. This count is used in weighting the parameters of a model.
- **Test collection**
  A collection specifically created for evaluating experimental IR systems. It usually comes with a set of queries, and a labeling (decided by human experts) that decides which documents are relevant to each query. TIPSTER is one of the most prevalent test collections currently. Another useful collection for classification is the Reuters text categorization test collection. Here there are no queries, but the documents are news articles labeled with a variety of topic designations.
- **TIPSTER**
  An ongoing project where various groups and institutions have pooled their resources to solve problems in routing and information extraction. The framework is such that each team can work on a different piece and simply "plug" their application into the general architecture. The project also has a large test collection available.
- **TREC**
  Text REtrieval Conference. This group gives IR researchers a common test collection and a common evaluation system. Hence, systems can be compared and contrasted on the same data. You can visit the conference's home page for information about the conference and on-line versions of the proceedings.
- **Vector Space Model**
  A representation of documents and queries where they are converted into vectors. The features of these vectors are usually words in the document or query, after stemming and removing stop-words. The vectors are weighted to give emphasis to terms that exemplify meaning, and are useful in retrieval. In retrieval, the query vector is compared to each document vector. Those that

are closest to the query are considered to be similar, and are returned. SMART is the most famous example of a system that uses a vector space model.

- **Weighting**
  Usually referring to terms, the process of giving emphasis to the parameters for more important terms. In a vector space model, this is applied to the features of each vector. A popular weighting scheme is TF*IDF. Other possible schemes are Boolean (1 if the term appears, 0 if not), or by term frequency alone. In a vector model, the weights are sometimes normalized to sum to 1, or by dividing by the square root of the sum of their squares.

Appendix A

Stop Word List

Defense Technical Information Center / Defense RDT&E Online Systems (DTIC-DROLS) and standard Varity Stop List [10] and an added collection specific term "Reuter"

"A", "ABOUT", "ABOVE", "ACCORDING", "ACROSS", "AFTER", "AFTERWARDS", "AGAINST", "ALBEIT", "ALL", "ALMOST", "ALONE", "ALONG", "ALREADY", "ALSO", "ALTHOUGH", "ALWAYS", "AMONG", "AMONGST", "AN", "AND", "ANOTHER", "ANY", "ANYBODY", "ANYHOW", "ANYONE", "ANYTHING", "ANYWAY", "ANYWHERE", "APART", "ARE", "AROUND", "AS", "AT", "AUTHOR", "AV", "AVAILABLE", "BE", "BECAME", "BECAUSE", "BECOME", "BECOMES", "BECOMING", "BEEN", "BEFORE", "BEFOREHAND", "BEHIND", "BEING", "BELOW", "BESIDE", "BESIDES", "BETWEEN", "BEYOND", "BOTH", "BUT", "BY", "CAN", "CANNOT", "CANST", "CERTAIN", "CF", "CFRD", "CHOOSE", "CONDUCTED" , "CONSIDERED", "CONTRARIWISE", "COS", "COULD", "CRD", "CU", "DAY", "DESCRIBED", "DESCRIBE", "DESIGNED", "DETERMINE", "DETERMINED", "DIFFERENT", "DISCUSSED", "DO", "DOES", "DOESN'T", "DOING", "DOST", "DOTH", "DOUBLE", "DOWN", "DUAL", "DUE", "DURING", "EACH", "EITHER", "ELSE", "ELSEWHERE", "ENOUGH", "ET", "ETC", "EVEN", "EVER", "EVERY", "EVERYBODY", "EVERYONE", "EVERYTHING", "EVERYWHERE", "EXCEPT", "EXCEPTED", "EXCEPTING", "EXCEPTION", "EXCLUDE", "EXCLUDING", "EXCLUSIVE", "FAR", "FARTHER", "FARTHEST", "FEW", "FF", "FIRST", "FOR", "FORMERLY", "FORTH", "FORWARD", "FOUND", "FROM", "FRONT", "FURTHER", "FURTHERMORE" , "FURTHEST", "GENERAL", "GIVEN", "GET", "GO", "HAD", "HALVES", "HARDLY", "HAS", "HAST", "HATH", "HAVE", "HE", "HENCE", "HENCEFORTH", "HER", "HERE", "HEREABOUTS", "HEREAFTER", "HEREBY", "HEREIN", "HERETO", "HEREUPON", "HERS", "HERSELF", "HIM", "HIMSELF", "HINDMOST", "HIS", "HITHER", "HITHERTO", "HOW", "HOWEVER", "HOWSOEVER", "I", "IE", "IF", "IN", "INASMUCH", "INC", "INCLUDE", "INCLUDED", "INCLUDING", "INDEED", "INDOORS", "INSIDE", "INSOMUCH", "INSTEAD", "INTO", "INVESTIGATED", "INWARD", "INWARDS", "IS", "IT", "ITS", "ITSELF", "JUST", "KIND", "KG", "KM", "LAST", "LATTER" , "LATTERLY", "LESS", "LEST", "LET", "LIKE", "LITTLE", "LTD", "MADE", "MANY", "MAY", "MAYBE", "ME", "MEANTIME", "MEANWHILE", "MIGHT", "MORE", "MOREOVER", "MOST", "MOSTLY", "MORE", "MR", "MRS", "MS", "MUCH", "MUST", "MY", "MYSELF", "NAMELY", "NEED", "NEITHER", "NEVER", "NEVERTHELESS", "NEXT", "NO", "NOBODY" , "NONE" , "NONETHELESS", "NOONE", "NOPE", "NOR", "NOT", "NOTHING", "NOTWITHSTANDING", "NOW", "NOWADAYS", "NOWHERE", "OBTAINED", "OF", "OFF", "OFTEN", "OK", "ON", "ONCE", "ONE", "ONLY", "ONTO", "OR", "OTHER", "OTHERS", "OTHERWISE", "OUGHT", "OUR", "OURS", "OURSELVES", "OUT", "OUTSIDE", "OVER", "OWN", "PER", "PERFORMANCE", "PERFORMED", "PERHAPS", "PLENTY", "POSSIBLE", "PRESENT", "PRESENTED", "PRESENTS", "PROVIDE",

"PROVIDED", "PROVIDES", "QUITE", "RARTHER", "REALLY", "RELATED",
"REPORT", "REQUIRED", "RESULTS", "ROUND", "SAID", "SAKE", "SAME",
"SANG", "SAVE", "SAW", "SEE", "SEEING", "SEEM", "SEEMED", "SEEMING",
"SEEMS", "SEEN", "SELDOM", "SELECTED", "SELVES", "SENT", "SEVERAL",
"SFRD", "SHAFT", "SHE", "SHOULD", "SHOWN", "SIDEWAYS",
"SIGNIFICANT", "SINCE", "SLEPT", "SLEW", "SLUNG", "SLUNK", "SMOTE",
"SO", "SOME", "SOMEBODY", "SOMEHOW", "SOMEONE", "SOMETHING",
"SOMETIME", "SOMETIMES", "SOMEWHAT", "SOMEWHERE", "SPAKE",
"SPAT", "SPOKE", "SPOKEN", "SPRANG", "SPRUNG", "SRD", "STAVE",
"STAVES", "STILL", "STUDIES", "SUCH", "SUPPOSING", "TESTED", "THAN",
"THAT", "THE", "THEE", "THEIR", "THEM", "THEMSELVES", "THEN" ,
"THENCE" , "THENCEFORTH", "THERE", "THEREABOUT",
"THEREABOUTS", "THEREAFTER", "THEREBY", "THEREFORE",
"THEREIN", "THEREOF", "THEREON", "THERETO", "THEREUPON",
"THESE", "THEY", "THIS", "THOSE", "THOU", "THOUGH", "THRICE",
"THROUGH", "THROUGHOUT", "THRU", "THUS", "THY", "THYSELF",
"TILL", "TO", "TOGETHER", "TOO", "TOWARD", "TYPES", "TOWARDS",
"UNABLE", "UNDERNEATH", "UNLESS", "UNLIKE", "UNTIL", "UP", "UPON",
"UPWARD", "UPWARDS", "US", "USE", "USED", "USING", "VARIOUS",
"VERY", "VIA", "VS", "WANT", "WAS", "WE", "WEEK", "WELL", "WERE",
"WHAT", "WHATEVER", "WHATSOEVER", "WHEN", "WHENCE",
"WHENEVER", "WHENSOEVER", "WHERE", "WHEREABOUTS",
"WHEREAFTER", "WHEREAS", "WHEREAT", "WHEREBY", "WHEREFORE",
"WHEREFROM", "WHEREIN", "WHEREINTO", "WHEREOF", "WHEREON",
"WHERESOEVER", "WHERETO", "WHEREUNTO", "WHEREUPON",
"WHEREVER", "WHEREWITH", "WHETHER", "WHEW", "WHICH",
"WHICHEVER", "WHICHSOEVER", "WHILE", "WHILST", "WHITHER",
"WHO", "WHOA", "WHOEVER", "WHOLE", "WHOM", "WHOMEVER",
"WHOMSOEVER", "WHOSE", "WHOSEVER", "WHY", "WILL", "WILT",
"WITH", "WITHIN", "WITHOUT", "WORSE", "WORST", "WOULD", "WOW",
"YE", "YET", "YEAR", "YIPPEE", "YOU", "YOUR", "YOURS", "YOURSELF",
"YOURSELVES", "REUTER"

Appendix B

Porter Stemmer:

Following are the set of rules applied in order to achieve words stemming as described by Porter Stemmer [22].

Step 1a: "sses" is ➔ "ss" as in (caresses ➔ caress)
"ies" ➔ "i" as in (ponies ➔ poni)
"s" ➔ NULL as in (cats ➔ cat)

Step 1b: if m > 0: "eed" ➔ "ee" as in (agreed ➔ agree)
if "*v*ed" ➔ NULL as in (plastered ➔ plaster)
then "at" ➔ "ate" as in (conflated ➔ conflat ➔ conflate),
or "bl" ➔ "ble" as in (troubled ➔ troubl ➔ trouble),
or "iz" ➔ "ize" as in (sized ➔ siz ➔ size).
if "*v*ing" ➔ NULL as in (motoring ➔ motor)

Step 1c: *v*Y ➔ I as in (happy ➔ happi)

Step 2: for m > 0 do the following replacements:
"Ational" ➔ "ate" as in (relational ➔ relate)
"tional" ➔ "tion" as in (conditional ➔ condition)
"enci" ➔ "ence" as in (valency ➔ valenci ➔ valence)
"anci" ➔ "ance" as in (hesitancy ➔ hesitanci ➔ hesitance)
"izer" ➔ "ize" as in (digitizer ➔ digitize)
"abli" ➔ "able" as in (comfortably ➔ comfortabli ➔ comfortable)
"alli" ➔ "al" as in (radically ➔ radicalli ➔ radical)
"entli" ➔ "ent" as in (differently ➔ differentli ➔ different)
"eli" ➔ "e" as in (vilely ➔ vileli➔ vile)
"ousli" ➔ "ous" as in (analogously ➔ analogousli ➔ analogous)
"ization" ➔ "ize" as in (privatization ➔ privatize)
"ator" ➔ "ate" as in (operator ➔ operate)
"alism" ➔ "al" as in (federalism ➔ federal)
"iveness" ➔ "ive" as in (defensiveness ➔ defensive)
"fulness" ➔ "ful" as in (successfulness ➔ successful)
"ousness" ➔ "ous" as in (callousness ➔ callous)

"aliti" ➔ "al" as in (formality ➔ formaliti ➔ formal)

"iviti" ➔ "ive" as in (sensitivity ➔ sensitiviti ➔ sensitive)

"biliti" ➔ "ble" as in (extensibility ➔ extensibiliti ➔ extensible)

Step 3: for m > 0 do the following replacements:

"icate" ➔ "ic" as in (triplicate ➔ triplic)

"ative" ➔ NULL as in (informative ➔ inform)

"alize" ➔ "al" as in (formalize ➔ formal)

"iciti" ➔ "ic" as in (electricity ➔ electriciti ➔ electric)

"ful" ➔ NULL as in (successful ➔ success)

"ness" ➔ NULL as in (goodness ➔ good)

Step 4: for all m > 1 remove the following suffixes:

"al" as in (revival ➔ reviv)

"ance" as in (allowance ➔ allow)

"ence" as in (inference ➔ infer)

"er" as in (airliner ➔ airlin)

"ic" as in (microscopic ➔ microscop)

"able" as in (adjustable ➔ adjust)

"ible" as in (defensible ➔ defens)

"ant" as in (irritant ➔ irrit)

"ement" as in (replacement ➔ replac)

"ment" as in (adjustment ➔ adjust)

"ent" as in (dependent ➔ depend)

*s or *t "ion" as in (adoption ➔ adopt)

"ou" as in (homologou ➔ homolog)

"ism" as in (communism ➔ commun)

"ate" as in (activate ➔ activ)

"iti" as in (angularity ➔ angulariti ➔ angular)

"ous" as in (homologous ➔ homolog)

"ive" as in (effective ➔ effect)

"ize" as in (familiarize ➔ familiar)

Step 5a:

For m > 1 remove the suffix "e" as in (probate ➔ probat)

Step 5b:

For m > 1 and *D and *L ➔ single letter as in (controll ➔ control)

Appendix C

Sample Data

Sample Document 1

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="5544" NEWID="1">
<DATE>26-FEB-1987 15:01:01.79</DATE>
<TOPICS><D>cocoa</D></TOPICS>
<PLACES><D>el-salvador</D><D>usa</D><D>uruguay</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN> &#5;&#5;&#5;C T&#22;&#22;&#1;f0704&#31;reuteu f BC-BAHIA-COCOA-REVIEW 02-26 0105</UNKNOWN>
<TEXT>&#2;<TITLE>BAHIA COCOA REVIEW</TITLE>
<DATELINE> SALVADOR, Feb 26 - </DATELINE><BODY>Showers continued throughout the week in the Bahia cocoa zone, alleviating the drought since early January and improving prospects for the coming temporao, although normal humidity levels have not been restored, Comissaria Smith said in its weekly review.

The dry period means the temporao will be late this year. Arrivals for the week ended February 22 were 155,221 bags of 60 kilos making a cumulative total for the season of 5.93 mln against 5.81 at the same stage last year. Again it seems that cocoa delivered earlier on consignment was included in the arrivals figures.

Comissaria Smith said there is still some doubt as to how much old crop cocoa is still available as harvesting has practically come to an end. With total Bahia crop estimates around 6.4 mln bags and sales standing at almost 6.2 mln there are a few hundred thousand bags still in the hands of farmers, middlemen, exporters and processors.

There are doubts as to how much of this cocoa would be fit for export as shippers are now experiencing dificulties in obtaining +Bahia superior+ certificates.

In view of the lower quality over recent weeks farmers have sold a good part of their cocoa held on consignment.

Comissaria Smith said spot bean prices rose to 340 to 350 cruzados per arroba of 15 kilos.

Bean shippers were reluctant to offer nearby shipment and only limited sales were booked for March shipment at 1,750 to 1,780 dlrs per tonne to ports to be named.

New crop sales were also light and all to open ports with June/July going at 1,850 and 1,880 dlrs and at 35 and 45 dlrs under New York july, Aug/Sept at 1,870, 1,875 and 1,880 dlrs per tonne FOB.

Routine sales of butter were made. March/April sold at 4,340, 4,345 and 4,350 dlrs. April/May butter went at 2.27 times New York May, June/July at 4,400 and 4,415 dlrs, Aug/Sept at 4,351 to 4,450 dlrs and at 2.27 and 2.28 times New York Sept and Oct/Dec at 4,480 dlrs and 2.27 times New York Dec, Comissaria Smith said.

Destinations were the U.S., Convertible currency areas, Uruguay and open ports. Cake sales were registered at 785 to 995 dlrs for March/April, 785 dlrs for May, 753 dlrs for Aug and 0.39 times New York Dec for Oct/Dec.

Buyers were the U.S., Argentina, Uruguay and convertible currency areas.

Liquor sales were limited with March/April selling at 2,325 and 2,380 dlrs, June/July at 2,375 dlrs and at 1.25 times New York July, Aug/Sept at 2,400 dlrs and at 1.25 times New York Sept and Oct/Dec at 1.25 times New York Dec, Comissaria Smith said.

Total Bahia sales are currently estimated at 6.13 mln bags against the 1986/87 crop and 1.06 mln bags against the 1987/88 crop.

Final figures for the period to February 28 are expected to be published by the Brazilian Cocoa Trade Commission after carnival which ends midday on February 27.
Reuter&#3;</BODY></TEXT>
</REUTERS>

Sample Document 2

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET"OLDID="5719" NEWID="176">
<DATE>26-FEB-1987 17:57:05.23</DATE>
<TOPICS><D>nat-gas</D></TOPICS>
<PLACES><D>usa</D><D>algeria</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN> &#5;&#5;&#5;Y&#22;&#22;&#1;f0255&#31;reuter f BC-LNG-IMPORTS-FROM-ALGE 02-26 0108</UNKNOWN>
<TEXT>&#2; <TITLE>LNG IMPORTS FROM ALGERIA UNLIKELY IN 1987</TITLE>
<AUTHOR> BY NAILENE CHOU WIEST, Reuters</AUTHOR>
<DATELINE> NEW YORK, Feb 26 - </DATELINE><BODY>Liquefied natural gas imports from Algeria are unlikely to happen in 1987 even though its economically feasible, U.S. industry analysts sources said.

Youcef Yousfi, director-general of Sonatrach, the Algerian state petroleum agency, indicated in a television interview in Algiers that such imports would be made this year.

"Contract negotiations, filing with the U.S. government and the time required to restart mothballed terminals will delay the import until 1988/1989," Daniel Tulis, a natural gas analyst with Shearson Lehman Bros. said.

Sonatrach is currently negotiating with two of its former customers, Panhandle Eastern <PEL> and Distrigas, a subsidiary of Cabot Corp <CBT> to resume LNG export, company officials told Reuters. A third, El Paso Gas, a subsidiary of Burlington Northern <BNI>, has expressed no interest.

Industry analysts said some imports of Algerian LNG were feasible. "On a marginal cost basis, the companies that have made capital investment to handle LNG import can operate profitably even in the current price environment," Frank Spadine, an energy economist with Bankers Trust, said.

Analysts did not forsee a major impact from Algerian imports on U.S. prices which are currently soft but expected to trend higher by the end of 1987.

A decline in gas drilling and the time lag to bring Gulf of Mexico productions onstream will tighten gas supplies and firm prices, Shearson's Tulis said.

In this context, Algerian LNG import would be a source of supplemental supply to U.S. domestic production, he added.

Company sources currently in talks with Algeria agree, saying that Algerian LNG would only serve to meet peak demand.

Company sources also said that any negotiations with Algeria would emphasize looser arrangements which would relate volumes to market requirements and prices to U.S. spot market values.

Reuter&#3;</BODY></TEXT>
</REUTERS>

Categories:

acq	Cpu	instal-debt	oilseed	ship
alum	Crude	interest	orange	silk
austdlr	Cruzado	inventories	palladium	silver
austral	Dfl	ipi	palm-meal	singdlr
barley	Dkr	iron-steel	palm-oil	skr
bfr	Dlr	jet	palmkernel	sorghum
bop	Dmk	jobs	peseta	soy-meal
can	Drachma	l-cattle	pet-chem	soy-oil
carcass	Earn	lead	platinum	soybean
castor-meal	escudo	lei	plywood	stg
castor-oil	f-cattle	lin-meal	pork-belly	strategic-metal
castorseed	ffr	lin-oil	potato	sugar
citruspulp	fishmeal	linseed	propane	sun-meal
cocoa	flaxseed	lit	rand	sun-oil
coconut	fuel	livestock	rape-meal	sunseed
coconut-oil	gas	lumber	rape-oil	tapioca
coffee	gnp	lupin	rapeseed	tea
copper	gold	meal-feed	red-bean	tin
copra-cake	grain	mexpeso	reserves	trade
corn	groundnut	money-fx	retail	tung
corn-oil	groundnut-meal	money-supply	rice	tung-oil
cornglutenfeed	groundnut-oil	naphtha	ringgit	veg-oil
cotton	heat	nat-gas	rubber	wheat
cotton-meal	hk	nickel	rupiah	wool
cotton-oil	hog	nkr	rye	wpi
cottonseed	housing	nzdlr	saudriyal	yen
cpi	income	oat	sfr	zinc

Sample Document Terms:

For Document ID = 1

Term	Term-ID	Term-Freq	Term	Term-ID	Term-Freq
SHOWER	0	1	CONTINU	1	1
ZONE	4	1	ALLEVI	5	1
DROUGHT	6	1	EARLI	7	1
JANUARI	8	1	IMPROV	9	1
PROSPECT	10	1	NORMAL	13	1
HUMID	14	1	LEVEL	15	1
RESTOR	16	1	WEEKLI	19	1
REVIEW	20	1	DRY	21	1
MEAN	23	1	LATE	24	1
MAKE	30	1	CUMUL	31	1
SEASON	33	1	STAGE	35	1
AGAIN	36	1	DELIV	37	1
EARLIER	38	1	OLD	42	1
HARVEST	44	1	PRACTIC	45	1
STAND	48	1	HUNDR	49	1
THOUSAND	50	1	HAND	51	1
MIDDLEMEN	53	1	PROCESSOR	55	1
FIT	56	1	EXPERIENC	58	1
DIFICULTI	59	1	OBTAIN	60	1
SUPERIOR	61	1	CERTIF	62	1
VIEW	63	1	LOWER	64	1
QUALITI	65	1	RECENT	66	1
WEEK	67	1	GOOD	69	1
PART	70	1	HELD	71	1
SPOT	72	1	PRICE	74	1
ROSE	75	1	CRUZADO	76	1
ARROBA	77	1	RELUCT	78	1
OFFER	79	1	NEARBI	80	1
BOOK	83	1	MARCH	84	1
NAME	88	1	LIGHT	90	1
GO	93	1	UNDER	94	1
FOB	98	1	ROUTIN	99	1
APRILMAI	102	1	WENT	103	1
DESTIN	108	1	COVERT	111	1
CAKE	115	1	REGIST	116	1
AUG	117	1	BUYER	118	1
ARGENTINA	119	1	CONVERT	120	1
LIQUOR	121	1	SELL	122	1
CURRENT	123	1	FINAL	124	1
EXPECT	125	1	PUBLISH	126	1
BRAZILIAN	127	1	TRADE	128	1
COMMISS	129	1	CARNIV	130	1
MIDDAI	131	1	COME	11	2
TEMPORAO	12	2	PERIOD	22	2

ARRIV	25	2
CONSIGN	39	2
DOUBT	41	2
FARMER	52	2
SHIPPER	57	2
BEAN	73	2
LIMIT	82	2
OPEN	91	2
BUTTER	100	2
U	109	2
CURRENC	112	2
URUGUAI	114	2
FEBRUARI	27	3
PORT	87	3
AUGSEPT	97	3
OCTDEC	106	3
BAHIA	2	4
SMITH	18	5
MLN	34	5
COCOA	3	6
TIME	104	7
NEW	89	9

KILO	29	2
FIGUR	40	2
ESTIM	46	2
EXPORT	54	2
SOLD	68	2
SHIPMENT	81	2
TONN	86	2
JULI	96	2
SEPT	105	2
S	110	2
AREA	113	2
END	26	3
TOTAL	32	3
JUNEJULI	92	3
MARCHAPRIL	101	3
DEC	107	3
COMISSARIA	17	5
BAG	28	5
CROP	43	5
SALE	47	7
YORK	95	8
DLR	85	14

Sample Document Phrases:

For Document ID = 1

Phrase	Ph-ID	Ph-Freq
SHOWER CONTINU	0	1
COCOA BAHIA	2	1
ZONE ALLEVI	4	1
EARLI DROUGHT	6	1
JANUARI IMPROV	8	1
PROSPECT COME	10	1
TEMPORAO NORMAL	12	1
LEVEL HUMID	14	1
RESTOR COMISSARIA	16	1
WEEKLI REVIEW	19	1
PERIOD MEAN	21	1
TEMPORAO LATE	23	1
FEBRUARI END	25	1
KILO BAG	27	1
MAKE CUMUL	29	1
TOTAL SEASON	31	1
STAGE MLN	33	1
DELIV COCOA	35	1
EARLIER CONSIGN	37	1
FIGUR ARRIV	39	1
OLD DOUBT	41	1
CROP COCOA	43	1

Phrase	Ph-ID	Ph-Freq
CONTINU BAHIA	1	1
ZONE COCOA	3	1
DROUGHT ALLEVI	5	1
JANUARI EARLI	7	1
PROSPECT IMPROV	9	1
TEMPORAO COME	11	1
NORMAL HUMID	13	1
RESTOR LEVEL	15	1
WEEKLI SMITH	18	1
PERIOD DRY	20	1
TEMPORAO MEAN	22	1
END ARRIV	24	1
FEBRUARI BAG	26	1
MAKE KILO	28	1
TOTAL CUMUL	30	1
SEASON MLN	32	1
COCOA AGAIN	34	1
EARLIER DELIV	36	1
CONSIGN ARRIV	38	1
SMITH DOUBT	40	1
OLD CROP	42	1
HARVEST COCOA	44	1

PRACTIC HARVEST 45 1
END COME 47 1
ESTIM CROP 50 1
STAND SALE 54 1
MLN HUNDR 56 1
THOUSAND BAG 58 1
HAND FARMER 60 1
MIDDLEMEN EXPORT 62 1
DOUBT COCOA 64 1
FIT EXPORT 66 1
SHIPPER EXPERIENC 68 1
OBTAIN DIFICULTI 70 1
SUPERIOR BAHIA 72 1
VIEW LOWER 74 1
RECENT QUALITI 76 1
WEEK FARMER 78 1
SOLD GOOD 80 1
PART COCOA 82 1
HELD CONSIGN 84 1
SPOT BEAN 86 1
ROSE PRICE 88 1
CRUZADO ARROBA 90 1
SHIPPER BEAN 92 1
RELUCT OFFER 94 1
SHIPMENT NEARBI 96 1
SALE BOOK 99 1
SHIPMENT MARCH 101 1
TONN PORT 104 1
NEW CROP 106 1
SALE LIGHT 108 1
PORT JUNEJULI 111 1
GO DLR 113 1
UNDER NEW 115 1
SALE ROUTIN 121 1
SOLD MARCHAPRIL 123 1
BUTTER APRILMAI 125 1
WENT TIME 127 1
OCTDEC DLR 134 1
S COVERT 139 1
URUGUAI AREA 142 1
SALE CAKE 144 1
REGIST DLR 146 1
TIME AUG 149 1
U BUYER 151 1
URUGUAI ARGENTINA 153 1
CURRENC CONVERT 155 1
MARCHAPRIL LIMIT 157 1
SELL DLR 159 1
SALE BAHIA 161 1
ESTIM CURRENT 163 1

PRACTIC COME 46 1
CROP BAHIA 49 1
SALE BAG 53 1
STAND MLN 55 1
THOUSAND HUNDR 57 1
HAND BAG 59 1
MIDDLEMEN FARMER 61 1
PROCESSOR EXPORT 63 1
FIT COCOA 65 1
SHIPPER EXPORT 67 1
EXPERIENC DIFICULTI 69 1
OBTAIN BAHIA 71 1
SUPERIOR CERTIF 73 1
QUALITI LOWER 75 1
WEEK RECENT 77 1
SOLD FARMER 79 1
PART GOOD 81 1
HELD COCOA 83 1
SPOT SMITH 85 1
PRICE BEAN 87 1
ROSE CRUZADO 89 1
KILO ARROBA 91 1
SHIPPER RELUCT 93 1
OFFER NEARBI 95 1
SHIPMENT LIMIT 97 1
MARCH BOOK 100 1
SHIPMENT DLR 102 1
PORT NAME 105 1
SALE CROP 107 1
OPEN LIGHT 109 1
JUNEJULI GO 112 1
UNDER DLR 114 1
TONN FOB 120 1
SALE BUTTER 122 1
SOLD DLR 124 1
WENT BUTTER 126 1
YORK JUNEJULI 129 1
U DESTIN 137 1
CURRENC COVERT 140 1
URUGUAI OPEN 143 1
SALE REGIST 145 1
DLR AUG 148 1
OCTDEC DEC 150 1
S ARGENTINA 152 1
URUGUAI CONVERT 154 1
SALE LIQUOR 156 1
SELL MARCHAPRIL 158 1
TIME OCTDEC 160 1
SALE CURRENT 162 1
MLN CROP 165 1

FINAL FIGUR	166	1
PERIOD FEBRUARI	168	1
PUBLISH EXPECT	170	1
COCOA BRAZILIAN	172	1
TRADE COMMISS	174	1
END CARNIV	176	1
MIDDAI FEBRUARI	178	1
MLN ESTIM	51	2
TONN DLR	103	2
YORK JULI	117	2
YORK SEPT	132	2
DEC COMISSARIA	136	2
CURRENC AREA	141	2
CROP BAG	164	2
JUNEJULI DLR	130	3
DLR AUGSEPT	119	4
SMITH COMISSARIA	17	5
YORK NEW	116	8
PERIOD FIGUR	167	1
FEBRUARI EXPECT	169	1
PUBLISH BRAZILIAN	171	1
TRADE COCOA	173	1
COMMISS CARNIV	175	1
MIDDAI END	177	1
TOTAL BAHIA	48	2
SALE LIMIT	98	2
PORT OPEN	110	2
JULI AUGSEPT	118	2
SEPT OCTDEC	133	2
U S	138	2
MARCHAPRIL DLR	147	2
MLN BAG	52	3
YORK DEC	135	3
TIME DLR	131	4
TIME NEW	128	7

Sample Category – Term Frequencies:

Cat-Name	Term	TF in Category	DF in Category
COCOA	COCOA	12	2
COCOA	DLR	14	1
COCOA	TEA	8	1
COCOA	TRADE	2	2
COCOA	CONVERT	1	1
COCOA	CRUZADO	1	1
COCOA	NEW	10	2
COCOA	YORK	8	1
COCOA	TIME	7	1
COCOA	SALE	7	1
COCOA	CROP	5	1
COCOA	MLN	13	2
COCOA	BAG	10	2
COCOA	SMITH	5	1
COCOA	COMISSARIA	5	1
COCOA	BAHIA	4	1
COCOA	DEC	3	1
COCOA	OCTDEC	3	1
COCOA	MARCHAPRIL	3	1
COCOA	AUGSEPT	3	1
COCOA	JUNEJULI	3	1
COCOA	PORT	3	1
COCOA	TOTAL	3	1
COCOA	FEBRUARI	3	1
COCOA	END	3	1
COCOA	URUGUAI	2	1
COCOA	AREA	2	1
COCOA	CURRENC	2	1
COCOA	S	3	2
COCOA	U	3	2
COCOA	SEPT	2	1
COCOA	BUTTER	2	1
COCOA	JULI	2	1
COCOA	OPEN	2	1
COCOA	TONN	13	2
COCOA	LIMIT	2	1
COCOA	SHIPMENT	2	1
COCOA	BEAN	3	2
COCOA	SOLD	2	1
COCOA	SHIPPER	2	1
COCOA	EXPORT	11	2
COCOA	FARMER	2	1
COCOA	ESTIM	3	2
COCOA	DOUBT	2	1
COCOA	FIGUR	2	1
COCOA	CONSIGN	2	1
COCOA	KILO	2	1

Sample Category – Phrase Frequencies:

Cat-Name	Ph-Term1	Ph-Term2	PhF in Category	DF in Category
COCOA	INTERN	COFFE	1	1
COCOA	YORK	NEW	8	1
COCOA	TIME	NEW	7	1
COCOA	SMITH	COMISSARIA	5	1
COCOA	MLN	BAG	7	2
COCOA	HELD	COCOA	1	1
COCOA	PART	COCOA	1	1
COCOA	RISE	CALENDAR	1	1
COCOA	RISE	CONTINU	1	1
COCOA	CONTINU	COCOA	1	1
COCOA	TEA	COCOA	1	1
COCOA	INDONESIA	EXPORT	1	1
COCOA	QUOTA	INTERN	1	1
COCOA	QUOTA	INTRODUCT	1	1
COCOA	PORT	OPEN	2	1
COCOA	TONN	DLR	2	1
COCOA	SALE	LIMIT	2	1
COCOA	MLN	ESTIM	2	1
COCOA	TOTAL	BAHIA	2	1
COCOA	U	S	3	2

Sample General Term Frequencies:

Term	TF	Term-DF	Term-CF	Index-Term
TRADE	149	60	28	1
INTEREST	108	64	15	1
YEAR	129	87	25	0
NET	175	91	7	0
PRICE	254	104	42	0
BANK	377	105	16	0
MARKET	231	108	26	0
BILLION	353	109	24	0
S	219	114	41	0
SHARE	281	114	17	0
CT	306	114	6	0
TWO	163	115	29	0
MARCH	184	116	29	0
U	220	118	41	0
CORP	155	125	10	0
NEW	274	157	37	0
COMPANI	306	163	15	0
PCT	550	182	44	0
DLR	715	249	32	1
MLN	733	256	48	0

Sample General Phrase Frequencies:

Term-0	*Term-1*	*Ph-Freq*	*Ph-DF*	*Ph-CF*	*Idx-Phrase*
RECORD	APRIL	23	21	1	1
SHARE	MLN	22	21	2	1
PRIOR	**PAI**	**22**	**22**	**1**	**1**
NET	DLR	26	24	2	1
SHARE	CT	42	26	2	1
REV	NET	42	26	1	1
NOTE	MLN	27	27	1	1
SHARE	DLR	50	29	2	1
QTLY	DIV	29	29	1	1
SHARE	COMMON	39	30	2	1
PRIOR	CT	32	30	1	1
YORK	NEW	53	32	4	0
SHR	CT	58	37	1	1
NET	CT	59	39	1	1
REV	MLN	73	40	1	1
RECORD	MARCH	60	45	1	1
DLR	BILLION	183	64	8	0
U	S	207	109	41	0
MLN	DLR	254	120	12	0