EXAMINING ORTHOGONAL CONCEPTS-BASED MICRO-CLASSIFIERS AND THEIR CORRELATIONS WITH NOUN-PHRASE COREFERENCE CHAINS

Michelle Khalifé

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

September 2004

Your file Votre référence
ISBN: 0-612-94744-0
Our file Notre référence
ISBN: 0-612-94744-0

Abstract

Examining orthogonal concepts-based micro-classifiers and their correlations with noun-phrase coreference chains

Michelle Khalifé

The classification of a text into a category that corresponds with its content is an important task in Natural Language Processing. As we want to support shallow syntactic techniques, it must be possible to establish this category without a semantic text analysis. We therefore propose a statistical approach featuring a set of categories that cover orthogonal concepts: micro-classifiers. They allow for a robust multi-dimensional and multi-label text classification, which reveals to be beneficial in the context of automatic document summarization. The performance of these micro-classifiers is evaluated for four different cut-off thresholds using measures of precision and recall. Furthermore, we examine noun-phrase coreference chains within documents and attempt to find correlations with single and multi-label categorization. The presence of patterns could suggest better ways to enhance automatic document summarization.

To mom and dad. With love.

Acknowledgments

This has really been a roller coaster. A life experience from which I learned more than I would have ever imagined possible in a span of two years. One that I definitely did not expect and that I might not have been altogether ready for. But I would not trade this for the world. It has all been worthwhile and I owe it to so many people.

There are no words to express how thankful I am to my supervisor, Dr. Sabine Bergler, for her literally unbound patience and support. During many months I failed to see the light at the end of the tunnel and nearly stopped believing the journey would ever terminate. I was touched to find in her, not only a thesis advisor, but also a friend and a counselor. One who never seemed to give up on me. There was just no better motivation to finish.

Dr. René Witte, my friend and mentor, who always listened to me complain and never tired from advising me. I cannot think of a time I asked for his help and did not get it. I am forever grateful to him and consider myself very lucky to have met him and benefited from his experience.

Dr. Leila Kosseim, with whom I took my first Natural Language Processing course and who encouraged me to experience research. She has always been close, motivating, and available for advice and guidance.

Dr. Bassam Shayya, my professor at the American University of Beirut, who always believed in me and knew that I would love research.

I would also like to thank all my colleagues at CLaC, for their friendship and support. In particular, Zhuoyan Li (Robert), who has always been ready to help me and to whom I owe so much, Alina, Abolfazl, and Yunyu, for their academic advice, and Osama, for all the moral support. Thank you from the bottom of my heart.

Finally, I would like to thank my family and my closest friends for believing in me so strongly and supporting me throughout these stressful and fulfilling years.

Contents

List of Figures ix

List of Tables x

1 Introduction 1

1.1 Automatic Document Summarization 2

1.2 Multi-Label Text Classification 4

1.3 Problem Definition and Scope 5

1.4 Thesis Organization 6

2 Literature Review 8

2.1 Text Classification and Text Categorization 8

2.1.1 Single-Label and Multi-Label 9

2.1.2 Binary Systems and Ranking Systems 9

2.1.3 Category Pivoted and Document Pivoted 10

2.2 Applications of Text Classification 10

2.2.1 Document Organization 11

2.2.2 Web Organization 12

2.2.3 Recommending Systems 12

2.2.4 Natural Language Processing 13

2.3 Approaches to Text Classification 13

2.3.1 Knowledge Engineering Approach 13

2.3.2 Machine Learning Approach 14

2.3.3 Non-Learning Approach 20

2.4 Optimization of Text Classification Systems 20

2.4.1 Thresholding Strategies 20

2.5 Evaluation of Text Classification Systems 21
2.5.1 Single Systems 22
2.5.2 Multiple Systems 24

3 Micro-Classifiers **27**
3.1 Binary Categories 27
3.1.1 Common Categories for News Articles 28
3.1.2 Adopted Categories for Classification 33
3.1.3 Orthogonal Concepts-based Categories 36
3.2 Statistical Text Classification 38
3.2.1 Naive Bayes and Rainbow 39
3.2.2 Training and Testing Corpora 41
3.3 Multi-Dimensional Categorization 43
3.3.1 Gold Standard and Delta Matrix 44
3.3.2 Difference and Accuracy 46
3.3.3 Precision and Recall 50
3.3.4 Data Sensitivity 56
3.3.5 Structure Sensitivity 59
3.3.6 Binary Structure 64

4 Categories and Coreference Chains **68**
4.1 Category Labels 69
4.2 Coreference Chains Statistics 70
4.2.1 Length and Frequency 70
4.2.2 Singleton Chains 74
4.2.3 Non-Singleton Chains 77
4.3 Chains and Multi-label Category Patterns 81

5 Conclusion **85**

Bibliography **88**

A Similarity metrics **97**

B Clustering Algorithms **98**
B.1 Hierarchical Clustering 98

B.1.1 Agglomerative clustering 98
B.1.2 Devisive clustering 99
B.2 Non-Hierarchical Clustering 99
B.2.1 Sequential Clustering Algorithms 100

C Training the micro-classifiers **101**

D Evaluation results **105**
D.1 Training 105
D.2 Orthogonal categories 107

List of Figures

2.1 The CONSTRUE rule for the category Australian-Dollar 14
3.1 Distribution of difference per news source overall documents 50
4.1 Frequency of chains of different lengths 72
4.2 Singleton chains vs. total chains overall documents 74
4.3 Percentage of singleton chains over all documents 75
4.4 Number of non-singleton chains per document 78
4.5 Total number of non-singleton chains per document across categories 81

List of Tables

2.1 A contingency table for category c_i 22
2.2 Five versions of the Reuters benchmark 25
2.3 20 Newsgroups 25
3.1 Linearly structured news source categories 30
3.2 Total number of categories for linearly-structured news sources 31
3.3 Hierarchically structured news source categories 32
3.4 News sources for DUC 2002 and DUC 2003 42
3.5 Number of training documents for different categories 43
3.6 Gold standard entry for document NYT19981023.0251 45
3.7 Gold-standard (★), rainbow (⋒), and Δ-matrix entries for three documents 46
3.8 Frequency of documents and number of categories for which $|\delta| > 0$. 47
3.9 Frequency of documents for average difference ranges 48
3.10 Accuracy and average difference for all categories 48
3.11 Accuracy for category *Disasters* 48
3.12 Document frequency and difference ranges across categories 49
3.13 Precision and recall 52
3.14 Categories with best precision and recall 56
3.15 Accuracy for short and long documents 57
3.16 Precision and recall for long documents 58
3.17 Differences in performance for *Disasters* 60
3.18 Differences in performance for *Events* 62
3.19 Differences in performance for *Politics* 62
3.20 Differences in performance for *Sci|Tech* 64
3.21 F_1 for orthogonal categories 66
3.22 Categories with best f_1 67

4.1 Category assignment and probabilities 69
4.2 Example of chains per document 70
4.3 Set and document frequency for chains of different length across categories 71
4.4 Total number of chains and word-tokens per document across categories 73
4.5 Frequency of the lengthiest chains appearing in the lengthiest documents 73
4.6 Document distribution over different ratio ranges 75
4.7 Average percentage of singleton chains per document, sorted by DUC-cluster 76
4.8 Percentage of singleton chains per category 76
4.9 Source characteristics 77
4.10 Percentage of documents in different chains per document ranges . . . 78
4.11 Number of clusters at different ranges of average number of non-singleton chains per document 79
4.12 Truncated number of non-singleton chains per document for multi-label clusters 80
4.13 Min., max., and avg. number of non-singleton chains per document across categories 80
4.14 Document multi-labels and coreference-chains 82
4.15 Gaps between min. and max. no. of chains for different document multi-labels 83
4.16 Cluster multi-labels and coreference-chains 83
4.17 Cluster multi-labels and topics 84
5.1 Unencountered category associations 87
C.1 Training data for category *Politics* 101
C.2 Training data for category *Business* 102
C.3 Training category *Sci|Tech* 102
C.4 Training category *Health* 102
C.5 Training category *Sports* 103
C.6 Training data for category *Arts* 103
C.7 Training data for category *Disasters* 103
C.8 Training data for category *Events* 104
C.9 Training data for category *People* 104

D.1 Training data for binary category *Disasters* 105
D.2 Training data for modified category *Events* 106
D.3 Training data for modified category *Politics* 107
D.4 Training data for modified category *Sci|Tech* 107
D.5 Precision and recall for modified categories *Events*, *Politics*, and *Sci|Tech* 108
D.6 Precision, recall, and f-measure, for orthogonal categories 108

Chapter 1

Introduction

In the last decade, the number of electronically available documents has been increasing exponentially. As of March 2004, Google searches 4,285,199,774 web-pages; this is 40% more than what [Doandes, 2003] reported for the year before. Moreover, this ten-digit number only accounts for a portion of on-line data.

The abundant documents in digital format cannot be left rampant in an unorganized fashion and the need to access them efficiently calls for automatic indexing, defined by [Ruiz and Srinivasan, 1999] as the process of assigning, to an electronic document, "a set of categories (or index terms) that succintly describe its content." In the context of textual information, this process is also referred to as *text classification* or *categorization.* Note that the former subsumes the latter; we speak of categorization when documents are indexed by their theme or topic (e.g., health, politics, science), as opposed to their genre, author, publisher, or language, among others.

Text classification has become the subject of extensive research in the field of Natural Language Processing (NLP) [Sebastiani, 2002]. In fact, the task of assigning a document one or more predefined categories has many applications and serves numerous purposes, namely document classification, filtering, and information retrieval. In the context of this thesis, we go beyond simply retrieving documents based on their indexes, which is the most common application in the area of information retrieval [Jackson and Moulinier, 2002]. Instead, we look at challenges and requirements for categories when using them for the natural language processing task of automatic document summarization.

The following section presents the context that motivates our research: automatic summarization of news articles. The next describes our plan for using text classification as a means for automatic text summarization and shows how it differs from existing work. In the last sections of this chapter, we summarize our objectives and their scope and outline the thesis's structure.

1.1 Automatic Document Summarization

Notice the bold heading preceeding every article that appears in your daily newspaper: the headline. Like an abstract or an introduction, a headline is meant to attract the attention of the reader. It always holds the topic of the document and sometimes summarizes its most salient ideas. Usually, the headline is short and not always a proper and grammatical sentence. Lengthy headlines, which are less common, also exist. The following are the English translations of two French headlines for articles that appeared in the Lebanese newspaper L'Orient LE JOUR, on Tuesday 27th July 2004.

> *Teheran is our number one enemy, affirms Baghdad.*
>
> *To spend time, Saddam Hussein gardens, reads the Coran, and writes poetry.*

In the context of the Document Understanding Conferences (DUC) [duc, 2004], headlines reflect the general idea of a news article. The first of the DUC 2003 tasks was to automatically create a 10-words short and headline-like summary of a document. There were no restrictions on the format or structure of the summary, i.e., it did not have to be a grammatically correct sentence.

There are many different ways that researchers have adopted and implemented to obtain automatic summaries. ERSS [Bergler *et al.*, 2003] is the CLaC[1] Lab's summarization system that participated in DUC 2003's Task 1. It is based on coreference resolution and fuzzy logic. After detecting the noun-phrases (NPs) in the document, ERSS uses some heuristics to group, according to certain fuzzy thresholds, NPs that refer to the same entity in what is called a coreference chain. Low fuzzy thresholds,

[1] Computational Linguistics at Concordia

such as 0, 0.2, and 0.4, allow more lenient assignments of noun-phrases to coreference chains, while elevated thresholds such as 0.8 or 1 are very strict [Bergler *et al.*, 2003]. Here is an example:

> [...] But while *government officials* and *relief workers* have applauded the imminent *arrival* of the *10 helicopters*, *six* fixed-wing *planes* and *hundreds* of *soldiers*, other *people* in *Mozambique* are complaining that *the outside world* took too long to respond to *the crisis* here.
>
> By *the time* the aircraft and *soldiers* start flying, virtually all of *the people* trapped on the *trees* and *rooftops* will already have been saved. And *the flood waters* of the *Limpopo* and the *other rivers* have started to recede, *officials* said *Saturday*.
>
> *The government*, which had issued an urgent *appeal* for $65 million, has received only about *half* that *amount*, *officials* said *Saturday*. And on *Friday*, *Leonardo Simao*, *the foreign minister*, bluntly told *reporters* that *he* believed the *delays* in *assistance* had resulted in *unnecessary deaths*. *No one* knows exactly how *many people* have died, but *relief agencies* estimate *the toll* in *the thousands* [...]

The detected noun-phrases in this extract are shown in italics and we count four coreference chains for a fuzzy setting $\gamma = 0.4$:

1. government officials, officials, officials.
2. Saturday, Saturday.
3. people, the outside world, the people, many people.
4. relief workers, soldiers, soldiers.

A fuzzy setting $\gamma = 0.6$ would reject "the outside world" from the third coreference chain as well as "relief workers" from the fourth coreference chains.

There are several coreference chains in a document and the primary assumption is that the lengthiest chains, i.e., those that consist of the largest number of noun-phrases, hold the most representative entities in the document. A summary is built by returning the lengthiest noun-phrase in the lengthiest chains, until the required length-limit for the summary is reached.

This is a knowledge-poor approach that sometimes results in incoherent summaries. For instance, the summary for the above document is:

> nearly 1 million people, many government officials, Mozambique, the 10 helicopters.

The hypothesis in this thesis is that the detection of the document's topic could be of great use. In fact, at DUC 2003, we appended the topic of a text to its summary. This actually boosted the usefulness of the latter [Over, 2003].

1.2 Multi-Label Text Classification

What does the following summary tell the reader?

> Both of the islands, inflicting property damage, Antigua's 28 main hotels

There has been some property damage on both islands? In Antigua? And what about its 28 main hotels? Were they damaged too? And where did the damage come from?

These questions are better answered when the article is labeled: WEATHER DISASTER. Now, we know that the document was about some kind of storm, hurricane, or even typhoon. This information is crucial for the reader, who would not have been able to make much sense only from the three selected noun-phrases.

Detecting the topic of an article is a natural language processing task known as text classification. Our goal is to have an accurate classification of news articles to shed some light, if need be, over their summaries. In order to do that, we must first choose a set of relevant categories into which news articles are going to be categorized. These categories must reflect the content of the article. The next step is to decide which classification approach would yield the most accurate classification and we find that a linear structure and a single classifier over all categories is not the best choice for a multi-label classification, i.e., one that results in the assignment of a document to multiple categories. Instead, we adopt a design that allows a multi-dimensional classification of an article over any number of concepts.

Although research in text classification is forty years old [Meretakis *et al.*, 2000], it has reached its peak in the nineties. Not only were numerous tools developed,

but there has also been a lot of work in order to optimize the performance of text classifiers. Testing is carried over common benchmarks and pre-defined categories. Therefore, not much has been reported on category structure or training. We examine these in detail in the context of this thesis. We will first look at the initial set of categories and structures we designed and evaluate the results in terms of precision and recall. Then, by modifying the structure of some categories, we try to find the reasons behind the weak and strong performance of their micro-classifiers and determine what the final and best classification consists of.

1.3 Problem Definition and Scope

This thesis presents text classification as an enhancement for the automatic summarization of news articles; intuitively, the detection of a document's topic should help improve the quality of its summary.

For a long time, researchers in the field have mainly focused their efforts on achieving better accuracy; they investigate the different approaches to text classification and attempt to optimize them by refining or combining old and new methods. Our interests in the results of the classification are slightly different: we are concerned with the categories that reflect the content of a document without a semantic analysis of the latter. Therefore, we design a set of categories that are useful for the particular task of news articles summarization.

First, we adopt the structure that maximises classification gain, i.e., k-independent classifiers (one per category) as opposed to one linear classifier for all categories. Then, we look at the possible categories we can assign news articles to and question their *relevancy*, in terms of the thematic knowledge brought forth by the category, their *feasibility*, in terms of training and testing data, and their *efficiency*, in terms of category performance. The latter is evaluated using both measures of precision and recall, and for different degrees of *generality* or *specificity*, for each category.[2]

With the set of categories and their structure defined, we look into possible correlations between the news articles' categories and their noun-phrase coreference chains. The idea is to determine whether or not Fuzzy-ERS [Witte and Bergler, 2003], a knowledge-poor coreference resolution system [Bergler, 1997], is category sensitive,

[2]Performance for different degrees of generality and specificity will not be measured for all categories.

and consequently set new summary-enhancing parameters for ERSS, the summarization system [Bergler *et al.*, 2003, Witte *et al.*, 2004].

1.4 Thesis Organization

The thesis is divided into five chapters. The current overviews and defines its context and motivation: the automatic summarization of news articles, as part of the Document Understanding Conferences (DUC) [duc, 2004], and its goal: the investigation of text classification as an enhancement for automatic summarization.

In the second chapter, we give a formal definition of text classification and mention some of its applications. We distinguish three main approaches: rule-based, machine learning, and non-learning. We present, in detail, the respective methods that belong to these approaches and that have also been adopted by researchers. Then, we briefly outline some of the optimization strategies that are applied for better performance. The last section in this chapter describes the metrics and benchmarks used for the evaluation of text classifiers.

The third chapter offers a set of orthogonal-concepts based categories for the categorization of news articles and justifies the reasons we trained k-independent classifiers (one per category), instead of a single one for all categories. The structure of the categories and the data used for training and testing are laid out as well. Cases of classifiers' sensitivity to both data and structure are highlighted and more practical models are proposed accordingly. We use both measures of precision and recall to evaluate the performance of the classifiers over the differently structured categories and for four cut-off thresholds. The chapter also introduces Andrew McCallum's *Bow* toolkit and clarifies why Naive-Bayes was the method adopted for the categorization of news articles.

In the fourth chapter, we describe ERSS, the summarization system, and overview its two major components; knowledge-poor coreference resolution [Bergler, 1997] and a fuzzy information system architecture [Witte, 2002]. Together, these two components build fuzzy coreference resolution chains that are needeed to build the summary. We gather the characteristics of noun-phrase coreference chains for every news article in the test corpus, i.e., total number of chains, number of chains for different length, average number of chains, average number of chains per length, and attempt to find

whether or not there exists correlation patterns with the type of categories.

The last chapter summarizes and evaluates our approach and offers an analysis of its results. It also defines future directions of research.

Chapter 2

Literature Review

This chapter includes five sections that are meant to present an overview on text classification. Section 2.1 defines some basic and field-related terminology. Section 2.2 offers examples of classification applications. Section 2.3 lists the different approaches that have been applied to solve classification tasks. Section 2.4 introduces optimization strategies and discusses thresholding methods. Finally, Section 2.5 shows how single or multiple classifier systems are evaluated.

2.1 Text Classification and Text Categorization

Text Classification (TC) is the discipline concerned with assigning a document to one or more predefined categories. Although it dates back to the early 1960's, it is only in the last decade that research, in what we nowadays consider as a natural language processing task, reached for its climax. The growing interest we are witnessing in TC is mainly due to the increasing availability of documents in digital form and the consequent need to have them arranged by categories, such as topic, author, genre, publisher, and language, for better access [Sebastiani, 2002].

The particular case of classifying a text into a category that corresponds with its content is called Text Categorization [Manning and Schütze, 2001]: "The goal in text categorization is to classify the topic or theme of a document." While some researchers remain faithful to the distinction between text classification and text categorization (text classification subsumes text categorization), others use the terms interchangeably, with the assumption that TC is always based on content [Jackson

and Moulinier, 2002]. Unless specified otherwise, we make the same assumption in the present context.

More formally, Sebastiani defines text categorization as the task of assigning a boolean value to each pair $\langle d_j, c_i \rangle \in D \times C$, where $D = \{d_1, d_2, ..., d_{|d|}\}$ is a domain of documents and $C = \{c_1, c_2, ..., c_{|c|}\}$ is a set of predefined categories [Sebastiani, 2002]. The boolean value, T for True and F for False, determines whether or not document d_j is classified under category c_i [Sebastiani, 2002]. Moreover, depending on the application, a document can be labeled with zero, one, or several thematic categories from the set C [Lavelli *et al.*, 2002]. This introduces the concepts of Single-Label and Multi-Label text categorization.

2.1.1 Single-Label and Multi-Label

The single-label case allows for a document to be stored under one category only, i.e., the result of a classification is a single category $c \in C$. On the other hand, in the multi-label case, a document can be assigned to any number of categories ranging from 0 to $|C|$, i.e., the result is a subset $|c_1, c_2, c_3, \ldots, c_n|$, where $c_i \in C$. For example, while in a single-label case, a news article about "Charles Schultz" is only categorized under *People*, in a multi-label case, it can be stored under *People* and *Arts* and we say that the categories overlap. Hence, another terminology for single-label and multi-label text categorization would be non-overlapping and overlapping categories, respectively. Binary TC is a special case of non-overlapping categories in which document d_j must either belong to c_i or to its complement $\overline{c_i}$ [Sebastiani, 2002].

2.1.2 Binary Systems and Ranking Systems

Assigning a boolean value to the pair $\langle d_j, c_i \rangle$ is like saying YES or NO, d_j belongs to c_i. This is called a "hard categorization" [Sebastiani, 2002]. A *text classifier*, a program that carries out the task of automatic text classification, which returns such a result, is known as an Independent Binary Classifier. Categories are assumed to be stochastically independent of each other, i.e., the classifier decides on a category regardless of its decisions on other categories. Examples include CONSTRUE, DTrees, Naive Bayes, DNF, NNet.Parc, CLASSI, Rocchio, and Sleeping Experts [Yang, 1999].

Another type of classifiers are Ranking Systems, also called M-ary Classifiers,

where $m > 2$. Instead of providing a categorical response as to whether or not the document belongs to the category, results are here returned in a ranked fashion. The system produces a list of the candidate categories, ranked by their "confidence score." The latter approximates to what degree the pair $\langle d_j, c_i \rangle$ is True. Examples of ranking systems include kNN, LLSF, and WORD [Yang, 1999]. The above-mentioned examples will be discussed in more detail in Sections 2.3.

M-ary classifiers results can be easily converted to binary by thresholding on the confidence score [Yang, 1999].

2.1.3 Category Pivoted and Document Pivoted

Whether a document is assigned to a category or vice versa, means that d_j belongs to c_i or d_j has been labeled with c_i. In both cases, the decision is T for the pair $\langle d_j, c_i \rangle$. Yet, there is a slight difference at the level of solving the problem. In *assigning a document to a category* or Category Pivoted Categorization (CPC), the classifier pivots around a given category c_i and looks for all documents in the set D that can be filed under it. Alternatively, in *assigning a category to a document* or Document Pivoted Categorization (DPC), the fixed instance is the document d_j and the classifier looks for all categories in the set C that can be assigned to it.

Some classifiers are built with a bias towards either category pivoted or document pivoted. However, DPC is suitable when documents become available at different moments in time and is used more often than CPC. The latter is a better candidate when a new category $c_{|c|+1}$ is added to the set C and all documents have to be reconsidered for classification under it [Sebastiani, 2002].

Furthermore, in DPC, the system ranks the categories in C according to their estimated appropriateness to document d_j. Similarly, in CPC, the classifier returns the list of documents in D ranked according to their estimated belonging to category c_i [Sebastiani, 2002].

2.2 Applications of Text Classification

The following section gives examples of several tasks that rely on text classification. In the context of this thesis, our domain is restricted to textual documents only. However, we would like to mention that text categorization techniques can also be

applied to speech recognition and multimedia documents [Sebastiani, 2002].

2.2.1 Document Organization

According to [Jackson and Moulinier, 2002], information retrieval does not start with a query, rather with the indexing of documents, i.e., documents are assigned key words or phrases that describe their content. The terms belong to a finite set called "controlled dictionary" that can also be viewed as the set C of categories and consequently, document indexing becomes a multi-label document pivoted categorization [Sebastiani, 2002].

Word Sense Disambiguation, the task of finding the sense of a polysemous word in a given context, is another alternative to indexing documents, characterizing them by word senses instead of terms. Again, consider the word occurrence context as the document and the word senses as the categories and we obtain single-label category pivoted categorization [Sebastiani, 2002].

Text Filtering is an instance of single-label TC and is defined by [Sebastiani, 2002] as the task of classifying a stream of incoming documents into two disjoint categories, the relevant and the irrelevant. Text filtering can also be found in the field of Topic Detection and Tracking (TDT). The idea in TDT is to identify and distinguish between events and then determine whether a given document is related to any previously identified event or not [Yang *et al.*, 2000].

The detection of spam emails is yet another case of text filtering [Nottelmann and Fuhr, 2001]. However, upon allowing the delivery of an email to one's inbox, we can further categorize it into appropriate folders such as *Administration*, *Research*, *Family*, *Friends*, *Jokes*,... In this case, text filtering is no longer binary as documents are filtered into one of several disjoint categories.

To illustrate this idea, [Hoch, 1994] presents INFOCLAS, a system that archives German business letters by classifying them into corresponding message types such as *Order*, *Offer*, *Inquiry*, *Enclosure*, and *Advertisement.*

Other examples include Case Law Categorization in fourty broad and high-level categories [Thompson, 2001] and Text Genre Detection such as *Research article*, *Novel*, *Poem*, *News article*, *Editorial*, *Homepage*, ... [Lee and Myaeng, 2002].

2.2.2 Web Organization

With the growing size of the World Wide Web, which presently contains over four billion pages, the automatic classification of hypertext becomes more and more important; search engines such as Google [google, 2004], Alta Vista [altavista, 2004], and MSN [msn, 2004], return web pages in sequentially ranked listings.

A more sophisticated hypertext classification presents web pages within a category structure. It targets web search results and uses the Support Vector Machine (SVM) algorithm to automatically organize them into hierarchical categories (e.g., Computers & Internet, Automotive, Entertainment & Media, Travel & Vacation ...) that are presented to the user. The latter can then focus on items in categories of interest and finds relevant results in half the time [Chen and Dumais, 2000]. Vivisimo [vivisimo, 2004] is another example of clustering engines that automatically organize search or database query results into meaningful hierarchical folders instead of long tedious lists.

Hypertext classification is a complex task as web pages are by far richer in information than simple text documents are. They not only carry HTML tags, but are also connected to other pages via hyperlinks. This allows for the extraction of meta-data information from those related web sites and researchers have just recently begun to explore ways of exploiting the semantic clues that are lost with a purely term based classifier without degrading accuracy [Chakrabarti *et al.*, 1998]. This question was also addressed in a study on hypertext categorization, conducted by [Liu *et al.*, 2002].

2.2.3 Recommending Systems

A recommender is a program that makes personalized product suggestions to a customer either based on the user's previous purchase or on the system's preferences. Learning Intelligent Book Recommending Agent (LIBRA) is a recommending system that searches its database of book information for titles that should appeal to the customer based on his profile [Mooney and Roy, 2000]. The latter is built before the system can make a recommendation. One might ask where, in the LIBRA's book recommending context, does text categorization fit. Essentially, we compare the system's task, which is to recommend a book if it fits the user's profile, to that of a binary classifier.

2.2.4 Natural Language Processing

In the context of natural language processing, the knowledge provided by the classification of a news article into one or more categories that reflect its content has been used for automatic summarization [Harabagiu and Lacatusu, 2002], Topic Detection and Tracking [Yang *et al.*, 2000, Macskassy *et al.*, 2001], Information Retrieval [Lee and Myaeng, 2002], and Information Extraction [Surdeanu and Harabagiu, 2002].

2.3 Approaches to Text Classification

There are three approaches to text classification: (1) the knowledge-engineering approach, which relies on rule-based methods and expert systems, (2) the machine learning approach, which involves supervised, rote, and unsupervised learning, and (3) the non-learning approach. In the following sections, we present these approaches and discuss the performance of their respective methods.

2.3.1 Knowledge Engineering Approach

[Sebastiani, 2002] introduces knowledge engineering as the most popular approach to TC until the late 1980's. It involves (1) an *expert system*, i.e., a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice [Jackson, 1999] and (2) a manually defined set of rules of type

if ⟨*DNFformula*⟩ **then** ⟨*category*⟩.

A document is classified under ⟨*category*⟩ if and only if it satisfies at least one of the clauses of the Disjunctive Normal Form (DNF) formula. Although the approach is very effective and can outperform modern machine learning classifiers, it is costly, labor-intensive, and suffers from the *knowledge acquisition bottleneck* [Sebastiani, 2002], i.e., the transfer and transformation of potential problem-solving expertise from some knowledge source to a program [Jackson, 1999].

CONSTRUE is the most famous example of rule-based classification systems. It was built by Carnegie Group Inc. (CGI) in 1989 for the Reuters News Agency and tested on 3% of the Reuters Corpus[1]. It applied 674 categories to the newswire feed

[1]Details are provided in Section 2.5.2

if	((australian-dollar-concept) & (dollar-concept)	**or**
	(australian-dollar-concept) & (australia-concept)	**or**
	(australian-dollar-concept) & ¬ (us-dollar-concept)	**or**
	(australian-dollar-concept) & ¬ (singapore-dollar-concept))	
then	(**assign** australian-dollar-category)	

Figure 2.1: The CONSTRUE rule for the category Australian-Dollar

(one rule per category) [Jackson and Moulinier, 2002], and achieved, on average, 90% in both precision and recall [Yang, 1999]. A sample rule type, adapted from [Jackson and Moulinier, 2002] and used in CONSTRUE, is illustrated in Table 2.1.

2.3.2 Machine Learning Approach

The machine learning approach consists of automatically inducing a *model* for category c_i through the observation of the characteristics of a given set of documents manually classified under c_i or $\overline{c_i}$ by a domain expert. The model learns the characteristics an unseen document should have to be classified under category c_i [Sebastiani, 2002].

A set of manually classified documents, also referred to as the set of *labeled data*, is split according to different percentages (e.g., 60–40, 70–30, 80–20), into a *training set* and a *test set*. The latter is not used for training. Instead, a small portion of the training set, the *validation test set*, is used to test and tune the parameters of the system and detect overfitting, before the final run. On a different note, the test set is not necessarily always labeled. It is commonly known as the set of unseen documents or (initially) *unlabeled data*.

There are three machine learning approaches to TC: Inductive (Supervised) Learning, Memory-based (Rote) Learning, and Unsupervised Learning.

Inductive Learning

An *inductive learning* program is one that learns classification rules based on a set of examples, which are used by the learning algorithm to create the model. In text classification, inductive learning is also called *supervised learning*. Several classifiers rely on inductive learning techniques. They fall under six categories: Probabilistic Classifiers, Symbolic Classifiers, Linear Classifiers, Regression Methods, Support Vector Machines, and Neural Networks.

1. Probabilistic Classifiers: Naive Bayes (NB) classifiers are widely used in text categorization.

> The basic idea is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document [Yang, 1999]. The naive part of NB methods is the assumption of word independence, i.e., the conditional probability of a word given a category is asumed to be independent from the conditional probabilities of other words given that category [Yang and Liu, 1999].

Classifiers which make the NB assumption use either a Multinomial model or a Multi-variate Bernoulli model. The first is a "uni-gram language model with integer word counts" [McCallum and Nigam, 1998], i.e., it represents documents by their word occurrences, while the second records the presence or absence of vocabulary terms, in the document, using a vector of binary components [Jackson and Moulinier, 2002]. After describing these two models in detail and comparing them, [McCallum and Nigam, 1998] find the multinomial model "to be almost uniformly better than the multi-variate Bernoulli model."

2. Symbolic Classifiers: We mention two types of symbolic classifiers, Decision Trees and Decision Lists.

> Decision tree (DTree) algorithms are used to select informative words based on an information gain criterion, and predict categories of each documents according to the occurrence of word combinations in the document [Yang, 1999].

A DTree text classifier is a tree that is grown from the root downward. A classification decision is made when a leaf-node is reached. The latter represents a category; several leaf-nodes may represent the same category [Jackson and Moulinier, 2002].

Unlike NB classifiers, DTrees use word combinations as predictors and their computation becomes exponentially complex. In order to reduce it, they are induced using "a selected set of features instead of the full vocabulary found in the training documents" [Liu *et al.*, 2002]. DTrees work best with large data sets; smaller ones lead to overfitting [Jackson and Moulinier, 2002]. The most popular DTree program is C4.5 [Quinlan, 1993].

Decision Lists (DL) also fall under the symbolic classifiers header and test for the combinations of terms. They are defined by [Jackson and Moulinier, 2002] as strictly ordered decision rules that contain only boolean conditions and test for the presence or absence of word features. Unlike DTrees, DNF rules are built in a "bottom-up fashion" and tend to generate more compact classifiers as rules are simplified through a series of modifications [Sebastiani, 2002]. However, like DTrees, DL suffer from overfitting. According to [Nottelmann and Fuhr, 2001], learning rules are inappropriate for high-dimensional problems like text classification. Examples of text classifiers based on decision lists are CHARADE [Yang, 1999], DL-ESC [Li and Yamanishi, 2000], RIPPER and Sleeping Experts [Cohen and Singer, 1999], Scar [Sebastiani, 2002], and Swap-1 [Apte *et al.*, 1994].

3. Linear Classifiers: Linear classifiers assign a document to either one of two assumed mutually exclusive sets [Jackson and Moulinier, 2002]. In other words, document d_j either belongs to category c_i or does not. The classifier separates between the positive examples and the negative ones. It is represented by a vector of weights learned using training data. Depending on which side of the hyperplane d_j falls on, it is either categorized under c_i or its inverse $\overline{c_i}$. When the document ends up on the wrong side of the line we have a classification error.

> Rocchio's algorithm is often used as a baseline in categorization experiments. One of its drawbacks is that it is not robust when the number of negative instances grows large [...] It has shown good performance when only a few positive examples are available. Furthermore, its performance can be improved by reducing the number of negative examples, and other enhancements [Jackson and Moulinier, 2002].

There are two types of linear algorithms: batch-linear and on-line linear. Rocchio is a batch-linear algorithm, i.e., weights can be computed directly from the available set of labeled documents. On-line linear algorithms "encounter examples singly and adapt weights incrementally" [Jackson and Moulinier, 2002]. Examples include Perceptrons and three versions of Winnow [Sebastiani, 2002] (positive, balanced, and sleeping experts [Cohen and Singer, 1999]), Gaussian [Chai *et al.*, 2002], and Windrow-Hoff [Jackson and Moulinier, 2002].

4. Regression Methods: We do not go in the details of regression methods. Examples of regression-based methods include Linear Least Square Fit (LLSF). We refer the interested readers to [Yang and Liu, 1999], [Yang, 1999], [Yang and Chute, 1994], and [Zhang and Oles, 2001].

5. Support Vector Machines: The Support Vector Machines (SVM) approach originated in 1995 when Vapnik introduced it for solving two-class pattern recognition problems [Yang and Liu, 1999]. It was then applied to solve binary classification tasks and has been shown to achieve "the highest classification scores in standard text collections" [Meretakis *et al.*, 2000].

> In their simplest form linear SVMs treat examples as points in a high-dimensional space. During training, their goal is to find hyperplanes that maximize the margin between positive and negative examples [...] the resulting hyperplanes are used to classify new examples [Meretakis *et al.*, 2000].

For more thorough discussions and experiments pertaining to SVMs, we refer the readers to the following literature: [Yang and Liu, 1999], [Chai *et al.*, 2002], [Joachims, 2001], [Zhang and Oles, 2001], and [Leopold and Kindermann, 2002].

6. Neural Networks: [Sebastiani, 2002] describes a Neural Network (NN) text classifier as a network which consists of *input units* representing terms, *output units* representing categories and *edges* connecting them. To classify a document, the weights of its terms are loaded in the input units and propagated forward throughout the network until they reach the output units. The values at those ends determine the categorization decisions [Sebastiani, 2002].

Neural networks train by backpropagation, i.e., when a document is misclassified, the error is *backpropagated* through the network. The latter's parameters are changed in order to avoid or reduce other occurences of that same error [Sebastiani, 2002]. This feature allowed neural networks to overcome the limitations of perceptrons [Manning and Schütze, 2001].

There are two types of neural networks: linear (e.g. CLASSI) and non-linear (e.g. NNet.PARC). Both classifiers presented below use a seperate neural network per category and have been tested on the Reuters-21450 corpus [Yang and Liu, 1999].

CLASSI stands for Classification System for Information. Like other classifiers, it assigns a given document to zero, one, or multiple categories. The system developed by [Ng *et al.*, 1997] uses the Perceptron Learning Algorithm (PLA) which starts with a random set of term-weights that are iteratively refined in order to minimize the number of misclassified examples. The system is practical and with its categorization speed, CLASSI is considered to be computationally efficient [Ng *et al.*, 1997].

NNet.Parc is a non-linear network developped at Xerox.PARC that learns not only the non-linear mapping from input words to categories [Yang and Liu, 1999], but also higher-order interactions between terms represented by one or more additional layers of units [Sebastiani, 2002]. However, NNet learning is time consuming [Yang and Liu, 1999] and the approach does not yield any substantial improvements over linear NN classifiers [Sebastiani, 2002].

Memory-based Learning

Unlike inductive learning, in *memory-based learning*, the rules are given to the program, rather than learnt from a set of examples [Jackson and Moulinier, 2002]. An illustration of this approach, also known as *rote learning*, is the k-Nearest Neighbour (k-NN) classifier.

1. k-Nearest Neighbour: The idea behind *nearest neighbour* classification is to assign a document to the category of its most similar neighbour, in the training set [Manning and Schütze, 2001]. In the k-NN case, the system attempts to classify a document by looking at the categories of its k closest neighbours, where $k > 1$.

During the training phase, the classifier memorizes all documents in the training set as well as their related characteristics. A distance metric should be defined to allow the system to measure how close an input document is to those in the training set.[2]

The classifier then returns the k top-ranking neighbours and is ready to decide which category the input document should be assigned to. There are multiple approaches applicable. The simplest one is to choose the majority class or classes among the k-nearest neighbours, for multiple or single assignment [Jackson and Moulinier,

[2]There are several similarity metrics that measure how far apart two documents are in vector space, for example, the Euclidean distance and the Cosine measure. Their formulas are given in Appendix A.

2002]. A more sophisticated approach involves weighting the distances, such that the further a neighbour is from the document, the less chances it has in having its category assigned [Jackson and Moulinier, 2002]. A final approach is described by [Yang, 1999] as follows: "The similarity score of each neighbour document to the new document being classified is used as the weight of each of its categories, and the sum of category weights over the k nearest neighbours are used for category ranking." The document is assigned to the categories with the score greater than a certain threshold value [Lam and Ho, 1998].

k-NN algorithms are sensitive to noisy examples [Lam and Ho, 1998] and require a fair amount of computation time during which documents are matched against each other. On the other hand, training is fast since all one needs to do is store the documents represented as vectors of features [Jackson and Moulinier, 2002]. Furthermore, their document-centricity allows them to efficiently handle classification tasks with a large number of categories [Jackson and Moulinier, 2002].

Unsupervised Learning

The final machine learning approach to text classification that we present is that of *unsupervised learning*. Here, the classification of the data in the training set is not known [Manning and Schütze, 2001]. The classifier learns on its own and has no human feedback, i.e., its decisions are neither supervised nor revised. Clustering techniques are suggested for unsupervised learning; documents with similar conditional word distributions are assigned to the same cluster [Slonim *et al.*, 2002]. However, these algorithms are not very useful in the context of automatic document summarization and especially when we are concerned with the classification of one text at a time. Furthermore, the results of the most outperforming clustering methods are comparable to those obtained by a supervised Naive Bayes classifier [Slonim *et al.*, 2002].

We briefly overview the various methods used for clustering in Appendix B and refer the interested readers to [Manning and Schütze, 2001] for more detailed information.

2.3.3 Non-Learning Approach

There is no model to be learnt in the non-learning approach. The system is expected to perform the classification task without prior training. *WORD* is a simple DPC ranking algorithm based on word matching between the document and the category name, both represented by the vector space model. The purpose of the approach is to measure the improvement statistical learning brings to the field of TC, as opposed to non-learning approaches [Yang, 1999].

2.4 Optimization of Text Classification Systems

Various methods are adopted to improve the performance of text classification systems. Examples include thresholding strategies, hierarchical models, feature selection, and boosting. However, optimizing a classifier does not always require external intervention. The algorithm itself can be enhanced or combined with that of another classifier for better results. We will briefly describe thresholding strategies only, since we use them in this thesis.

2.4.1 Thresholding Strategies

A number of classifiers are primarily ranking systems. They assign a relevance score to every document-category pair and threshold on these scores in order to obtain the equivalent binary assignment. In the absence of ideal classifiers and accurate probabilities, both scoring and thresholding methods influence the performance of the classifier. However, based on the false assumption that optimal thresholding does not make a significant difference, the subject has been relatively underexplored in text classification research [Yang, 2001].

Here, we present three basic and commonly used thresholding methods in TC. For more details on determining thresholds, refer to [Sebastiani, 2002].

Rank-based In rank-based (R-Cut) thresholding, the system produces a ranked list of categories for each document and assigns YES to each of the t top-ranking candidates, where t is an integer between 1 and m, and m is the number of categories. In other words, YES/NO decisions are obtained by thresholding on the listed ranks [Yang, 1999]. The value of the integer t can either be specified

by the user or fixed according to the value that optimizes the global performance of the classifier on the validation set [Yang, 2001].

R-Cut, also known as fixed thresholding, is commonly used with document pivoted TC. In this thesis, we use Bayes' rule and classify document d into category c if the difference between the posterior probabilities $P(c|d)$ and $P(\bar{c}|d)$ is larger than t, where $t \leq 0.4$. We test for better performance given the different values of t.

Proportional-based In proportional-based (P-Cut) thresholding, the system produces, for every category, a ranked list of documents, sorted by confidence score [Yang, 1999]. Here, the classifier assigns a YES to the k_j top-ranking documents, where $k_j = P(c_j) \times x \times m$ is the number of documents assigned to category c_j, $P(c_j)$ is the training-set probability of category c_j, and x is an empirically chosen parameter whose value is tuned on a validation set until the optimal trade-off between precision and recall is reached. [Yang, 2001]

Score-based In score-based (S-Cut) thresholding, the performance of the classifier is optimized for individual categories, rather than globally. The validation test set is used for learning the optimal threshold, i.e., the confidence score that generates the best f_1 value [Yang, 1999]. The latter is a combination of the measures of *precision* and *recall*. Refer to Section 2.5 for a detailed definition.

We will also apply S-Cut thresholding later in this thesis.

"Which is the optimal thresholding strategy?" and "How to jointly use the strengths of different strategies?" are two of the many questions addressed by [Yang, 2001] in a study on thresholding strategies and their effects on text classification. The author not only showed that optimal thresholding is not trivial, but that it also makes a significant difference in the overall performance of a classification system.

2.5 Evaluation of Text Classification Systems

Evaluating the performance of text classification systems is of great importance. In the following sections, we discuss how one measures the performance of a single system or compares that of multiple systems.

	YES is correct	NO is correct
Assigned YES	a	b
Assigned NO	c	d

Table 2.1: A contingency table for category c_i

2.5.1 Single Systems

The *effectiveness* of a classifier system is its ability to take the right classification decision indicating the performance of the system [Sebastiani, 2002]. There are several evaluation metrics that can be used to calculate the effectiveness, however, [Jackson and Moulinier, 2002] defend that "the methodology for evaluating a text classifier depends upon the task that the program is trying to perform."

The following list of metrics, borrowed from the area of information retrieval, is used for measuring the performance of binary classifiers:

Precision and Recall Precision (p) and Recall (r) complement each other. Precision, the degree of soundness, is the probability that the classification of document d_j under category c_i is correct [Sebastiani, 2002]. In other words, it is the proportion of documents for which the classifier correctly assigned category c_i [Jackson and Moulinier, 2002]. Precision for category i is the fraction of categories found and correct over the total categories found. It is computed, from the contingency Table 2.1, as follows: $p^i = \frac{a}{a+b}$ [Yang, 1999].

Recall, the degree of completeness, is the probability that document d_j will be classified under the right category c_i [Sebastiani, 2002]. The authors in [Jackson and Moulinier, 2002] define it as the proportion of target documents correctly classified. Recall for category i is the fraction of categories found and correct over the total categories correct. It is computed, from the contingency Table 2.1, as follows: $r^i = \frac{a}{a+c}$ [Yang, 1999].

Break-Even Point There is a trade-off between precision and recall. Tuning a classifier's parameters in order to obtain high recall comes at the price of low precision. Similarly, obtaining high precision means sacrificing recall. However, we reach the *break-even point* of the system when both measures are adjusted to equal values. Because the values of precision and recall cannot be made exactly equal, the break-even point is often interpolated from the nearest precision and

recall values. Yet, if the gap between the nearest precision and recall values is not modest, the break-even point would not faithfully represent the performance of the system [Yang, 1999].

In [Jackson and Moulinier, 2002], it is remarked that "the break-even point metric reflects more the properties of the recall-precision curve, rather than the performance of a given classifier."

F_β Measure The F_β measure combines the values of both precision (p) and recall (r). It is computed as follows [Yang, 1999]:

$$F_\beta = \frac{(\beta^2 + 1) \cdot p \cdot r}{\beta^2 \cdot p + r}$$

Although the value of β varies between 0 and ∞, allowing differential weighting of precision and recall, the metric is typically known as the f_1 measure, i.e., $\beta = 1$. In this particular case, precision and recall are balanced [Jackson and Moulinier, 2002] and the metric's score is maximized [Yang, 1999].

11-Point Average Precision The value of this metric is the average of precision points taken at 11 fixed recall values: $0.0, 0.1, 0.2, ..., 0.8, 0.9, 1.0$. It relies on ranking systems and is commonly used for routing tasks [Jackson and Moulinier, 2002].

Accuracy and Error Accuracy (A) and error rate ($E = 1 - A$) are alternate measures of effectiveness. They can be computed, from a local or a global contingency table, as follows:

$$A = \frac{a + d}{a + b + c + d}$$

$$E = \frac{b + c}{a + b + c + d}$$

However, they are not widely used in text classification as they are more insensitive to variations than other metrics [Sebastiani, 2002]. A deeper analysis of accuracy and error is offered by [Yang, 1999].

The above-mentioned metrics were applied for one category. To report the overall performance of the system, one must measure its effectiveness over the set $|C|$. Two averaging methods are adopted: *Macro-Averaging* and *Micro-Averaging.*

Macro-Averaging Macro-averaging sums the values obtained for precision and recall over all categories, and then divides by the number of categories $|C|$ to compute the average [Jackson and Moulinier, 2002].

Micro-Averaging Micro-averaging sums all individual decisions into it a global contingency table on which it computes precision and recall [Jackson and Moulinier, 2002].

2.5.2 Multiple Systems

Learning Speed, i.e., the time it takes a system to learn a model, is a measure that can be taken into consideration when evaluating the relative performance of a classification system with respect to a peer group of systems. However, for a reliable comparison of two or more classifiers, certain conditions hold. Not only should the comparisons be based on the same evaluation metric(s), but also the systems should train and test on the same data collection with the same split between the training set and the testing set [Sebastiani, 2002].

The following is a list of the most typically used benchmark corpora in text classification:

Reuters This corpus consists of over 20,000 Reuters newswire stories ranging from 1987 to 1991 [Yang, 2001]. The documents were assembled and indexed with categories by personnel from Reuters Ltd. [Lam and Ho, 1998]. Table 2.2 shows the five major versions of this benchmark. Although Reuters-21578 "ModApte" is the most popular test collection for text categorization, a larger version, Reuters Corpus Volume 1 (RCV1), has recently been made available and is expected to become a new standard benchmark in TC research and replace Reuters-21578.

Ohsumed This subset of the Medline document base, in which "documents are titles or title-plus-abstract from medical journals and categories are the *postable terms* of the MESH thesaurus" [Sebastiani, 2002], was set up by [Hersh *et al.*, 1994], at Oregon Health Sciences University. "The corpus contains 233,445 documents indexed using 14,321 unique categories" [Yang, 1999]. The collection is hard to learn for a good classifier due to the noisiness of the documents [Lam and Ho, 1998].

Version	Split	Number of			
		Docs	Training Docs	Test Docs	Categories
Reuters-22173	ModLewis	21,450	14,704	6,746	135
Reuters-22173	ModApte	14,347	10,667	3,680	93
Reuters-22173	ModWiener	13,272	9,610	3,662	92
Reuters-21578	ModApte	12,902	9,603	3,229	90
Reuters-21578	ModApte	12,902	9,603	3,299	10

Table 2.2: Five versions of the Reuters benchmark

Newsgroup	Total messages
alt.atheism	1001
comp.graphics	1000
comp.os.ms-windows.misc	1000
comp.sys.ibm.pc.hardware	1000
comp.sys.mac.hardware	1000
comp.windows.x	1001
misc.forsale	1000
rec.autos	1000
rec.motorcycles	1000
rec.sports.baseball	1000
rec.sports.hockey	1000
sci.crypt	1000
sci.electronics	1000
sci.med	1000
sci.space	1000
soc.religion.christian	997
talk.politics.guns	1000
talk.politics.mideast	1000
talk.politics.misc	990
talk.religion.misc	1000

Table 2.3: 20 Newsgroups

20 Newsgroups This corpus has originally been developed by Ken Lang in 1995. It holds approximately 20,000 messages from 20 different *Usenet* newsgroups (approximately one thousand messages each) 2.5.2. Table 2.3 shows the categories of these newsgroups.

The Associated Press (AP) Newswire Collection and its variants [Cohen and Singer, 1996] are another benchmark for TC, as are the TREC [trec, 2004] and MUC [muc, 2004] collections. In the context of this thesis, we made use of the

DUC [duc, 2004] collections. The corpus on which classifiers are trained or tested does not necessarily need to be one of the above-mentioned. Experiments in the Netherlands tested on a corpus of articles from the Dutch newspaper NRC [Ragas and Koster, 1998]. The data collection is not only chosen from available corpora, but also varies with the task. For example, when classifying web pages, training on a web page collection such as HV-28 and HV-225 is required [Yang, 2001].

Chapter 3

Micro-Classifiers

This chapter presents the approach we have chosen to classify news articles and offers a detailed evaluation of its results.

Section 3.1 reviews the common categories adopted by news sources for the classification of their articles. It also lists the nine categories we wish to consider for the task and the orthogonal concepts-based approach we adopt to classify the articles. In Section 3.2, we present the statistical method used to build the micro-classifiers, as well as our training and testing corpora. Finally, we evaluate our micro-classifiers in Section 3.3.

3.1 Binary Categories

In the context of automatic text summarization, it is useful to know which category a news article belongs to; a category under which a document falls reflects the content, the topic, or the general idea of the latter. Even a rough detection of a document's topic sheds more light over its simple noun-phrases based summary[1]. The following two summaries illustrate this claim:

> FLOOD DISASTER: nearly 1 million people, many government officials, Mozambique, the 10 helicopters

> WEATHER DISASTER: Both of the islands, inflicting property damage, Antigua's 28 main hotels

[1]ERSS's strategy for Task 1 at DUC 2003 [duc, 2004]

In the absence of the CATEGORIES, the readers would have been left wondering what the documents were really about. Little sense would have been made from the noun-phrases only.

In order to choose a set of categories that would make a significant difference when appended to a document's summary, we must first look at the available types of news articles and their different classifications across news sources. Then, we decide that an orthogonal-concepts based multi-label classification is better than a linear and a possibly single-labeled one.

3.1.1 Common Categories for News Articles

Categories vary across news sources in terms of (1) linear or hierarchical structure, (2) number, and (3) constituents, i.e., two or more news sources might assign the same document to different categories. However, we recognize seven linear headings under which articles are typically stored:

Local News articles group diverse documents pertaining to the city or country from which the news source is issued. Usually, these articles are about local political events but it is also common to report other local events such as concerts, fire incidents, trials, or demonstrations.

World News articles are generally about the international political events that are taking place all over the world, such as wars, treaties, kidnappings, terrorist attacks, and agreements of all kinds. *World News* also reports non-political world happenings like natural disasters.

Business articles are of diverse business, corporate, economic, and financial nature, both at the local as well as the world-levels.

Technology articles are scientific articles pertaining to various topics such as computers, Internet, engineering, high-tech instruments, medical equipment, and space missions.

Health articles vary in nature. Some are serious and are about illnesses, preventive treatments, and research results. Others are lighter and more read for the average person's daily life, like diet profiles and advice on staying healthy, fit,

and young. There are also articles about the inaugurations of medical centers and reports on state of the art equipment.

Sports articles normally carry results of, or news related to, local and reknown international games and events.

Entertainment articles discuss various artistic shows, movies and stars, concerts, fashion shows, Jazz fests, ...

Although news articles about the inauguration of a medical center are classified under category *Health*, they can also appear (depending where in the world the news source and the medical center are located) under *Local News*, *World News*, or *Business*. Similarly, a document describing medical equipment could also be attributed to category *Technology*. In other words, the classification of a news article into a category varies across news sources. Another example is a concert report. Would it fit under *Local* or *Entertainment*? In order to choose and construct our own set of categories and to define its structure, we survey the architectures different news sources have adopted to categorize their articles.

Table 3.1 displays the categories of different linearly structured online news sources as they compare to the seven common linear headings presented above. In the following list, we outline and illustrate the differences between news source categories:

1. **Terminology:** A heading, such as *Local*, may or may not exist among a news source's categories. For instance, category *Business* does not exist in the CALGARY SUN. In case it does, we mark an "X" in the relevant slot. The heading can have a different terminology such as *U.S* or *National*, in CNN INTL. ED. and GLOBE & MAIL, respectively. Both refer to relative local news. Similarly, GOOGLE's *Sci-Tech* holds articles that compare to those found under *Tech-News* in the CALGARY SUN, but ABC's *Money Scope* differs from BBC's *Business* as well as CNN's *World Business*.

2. **Subsumptions:** Some news sources assign a general label (categories subsuming others) to a document while others allow more specificity. The N.Y. TIMES has two headings for *Local*: *N.Y.* and *Nation*. It also distinguishes between *Science* and *Technology*. Furthermore, in the WASHINGTON TIMES, we count two headings for *Business*: *Business* and *Personal Finance*. The latter contains

Source	Local	World	BUS	Tech	Health	Sports	Ent	Extra
ABC News	X| U.S.	Intl| Politics	Money Scope	Sci-Tech	X	ESPN Sports	X| Travel	3
BBC	X|Local Politics	X	X	Sci-Tech	X	–	X	Education
CNN Intl. Ed.	U.S.	X	World Bus	X| Science & Space	–	World Sports	X| Travel	3| Special Reports
Reuters	U.S. News	X| Politics	Top Bus	X| Internet & Science	X	X	X	1-2| Reuters Edge
LA Times	U.S.	X|Politics	X	X	X	X	X	1-2-3
NY Times	NY|Nation	Intl	X	X|Science	X	X	–	3|Education
Washington Times	U.S.	X| Politics	X|Personal Finance	X	X	X	X	1-2-3
British Times	Britain	X	X| Your Money	–	X	X	X| Travel| Shopp.	1-3| Law
Times Online Ed.	Nation	X	Bus & Tech	Bus & Tech	Sci & Health	–	X	–
Financial Times	–	X	X| Markets| Markets Data and Tools| Industries| Management| Your Money|	X	–	X	Arts & WE	Lex| Comment and Analysis
Wall Street Journal	–	X| Politics and Policy	U.S.| Europe| Asia| Americas| Economy| Markets| Earnings| Media and Marketing| News Industry	X	Health Ed.	–	Leisure Weekend	Media and Marketing| Personal Journal
Google	Canada	X	X	Sci-Tech	X	X	X	1
NewsBlaster	U.S.	X	Finance	Sci-Tech	–	X	X	–
Canada.com	National	X	X	X	X	X	X	2| Agriculture
Calgary Sun	Canada	X	–	Tech News	–	–	–	2|Politics Law| War on Terror|
Herald Sun	State | National	X	X	X	Health-Science	X | Football	X	1| The Eye| Multimedia| Learn
Morning Sun	U.S.	X| Politics	X	X	X	X	X	1-2-3
Globe & Mail	National	Intl	Bus News	X	–	X	X	–

Table 3.1: Linearly structured news source categories

no. cat.	5	6	8	9	10	11	12	13	14	15
Source	Times Online	News-Blaster| Calgary Sun | Globe & Mail	BBC| Google	Canada.com	CNN| NYT	ABC| LA Times| Morning Sun	Washington Sun| British Times| Financial Times	Herald Sun	Reuters	WSJ

Table 3.2: Total number of categories for linearly-structured news sources

documents specifically about personal finance. In NEWSBLASTER [newsblaster, 2004], on the other hand, such articles would have been classified under the more general *Finance* heading. We count six specific *Business* categories for the FINANCIAL TIMES and nine for the WALL STREET JOURNAL. Also, the HERALD SUN holds a distinct *Football* category parallel to *Sports*, just as *Travel* and *Shopping* are separate from *Entertainment*, *Arts and Week-End*, or *Leisure week-end*, in ABC, BBC, the BRITISH TIMES, the FINANCIAL TIMES, and the WALL STREET JOURNAL.

3. **Combinations:** A perfect example of category combination is that of the TIMES ONLINE which combines *Business* and *Science* into *Bus & Tech* and here we count a total of five categories intead of six. Table 3.2 shows the different number of categories per news source. There are other combinations such as *Science-Health*, in the HERALD SUN, or *Science & Space*, in CNN INTL. ED.

4. **Additions**: Several categories are offered by some news sources and not others; *Education* in BBC, N.Y. TIMES, and HERALD SUN, *Law* in the BRITISH TIMES, *Agriculture* in CANADA.COM, *War on Terror* in the CALGARY SUN, *Media and Marketing* in the WALL STREET JOURNAL... The articles that fall under one of these particular categories would normally be classified under more general ones in other news sources. For example, a *War on Terror* article could either be assigned to *World* or *Local*. More common categories are [1] *Top Stories*, [2] *Offbeat* or *Oddities*, and [3] *Weather*. When one of these topics occur in the surveyed news sources, we use the mumber in square brackets to represent it in the last column of Table 3.1.

What adds to these particularities in hierarchically structured news sources, is how a number of different categories are clustered together or appear under the heading

Source	[NEWS]	Local	World	BUS	Tech	Health	Sports	Ent	Extra	Total*
Washington Post	–	Nation	X| Politics| Metro	X	X	X	X	X	Education| [Science and Health| Weird News| Special Reports]	11/1/3
Newsweek	Politics| Intl| Terrorism & Security	Local News	–	X	Tech-Science	X	X	X| Travel	Weather	9/1/3
San Jose Mercury News	Local| Nation-World| Education| Science & Health| Weird News| Special Reports	–	–	X	–	–	X	X	–	4/1/6

Table 3.3: Hierarchically structured news source categories

News. For example, NEWSWEEK's *News* section subsumes *politics*, *International*, and *Terrorism & Security*. Similarly, in the SAN JOSE MERCURY NEWS, categories *Local*, *Nation-World*, *Science & Health*, as well as others, fall under *News*. On the other hand, in the WASHINGTON POST, there is no *News* heading but *Science and Health*, *Weird News*, and *Special Reports*, clustered together. Table 3.3 shows the categorization scheme of these three news sources. Its last column is of the form $a/b/c$ where a is the total number of root categories, b is the number of root categories that are sub-categorized, and c is the number of the respective sub-categories. To illustrate this, we look at NEWSWEEK. There are nine root-categories: *News*, *Local*, *Bus*, *Tech-Science*, *Health*, *Sports*, *Entertainment*, *Travel*, and *Weather*. One of these, *News*, is sub-categorised into three sub-categories.

In brief, the assignment of a document to a category varies across news sources. We will have to choose our own set of categories and decide how articles are to be classified and under which different labels.

3.1.2 Adopted Categories for Classification

The news sources inspire us to consider six main categories: Politics, Business, Sci|Tech, Health, Sports, and Arts. Furthermore, given that our application is somewhat goal-driven, i.e., it is initially being developed for the classification of news articles in the DUC 2003 corpus, we add the three categories that divided the DUC 2002 corpus: Natural Disasters, Events (of single and multiple types), and People. In what follows, we clarify our choices and the criteria a document must have to fall under the different categories.

1. Politics

This category is the equivalent of news sources sections *Local News*, *World News*, *International*, and *Politics*. We are not concerned with the location of the event taking place, rather its nature. We filter from the mentioned sections the articles that pertain to political subjects and that relate either local or global matters, on an intra or an international governmental front. Political stories, such as "Kenyan Politics" or "Al-Qaeda News," namely include, court hearings and manifestations, nationwide decisions, international treaties and organizations, political figures, situations and events, terrorism acts, invasions, as well as others.

2. Business

This category is the equivalent of news sources sections *Business*, *Economics*, *Finance*, and *Your Money*. Also, news articles pulled from an economics or a financial journal, such as the FINANCIAL TIMES or the WALL STREET JOURNAL, are assigned to this category. There are numerous business subjects like corporations news, firm inaugurations, management, profits, losses, layovers, consumptions of goods, services, financial risks and gains, and stock market reports.

3. Sci|Tech

Scientific articles are not always technology oriented. Some news sources like the WASHINGTON POST, the NEW YORK TIMES, REUTERS, and the HERALD SUN, distinguish between categories *Science* and *Technology*. Others, like the WASHINGTON

TIMES and the FINANCIAL TIMES do not. In their design, both scientific and technology based articles fall under the same category. Here, our *Sci|Tech* category is the equivalent of news sources sections *Technology*, *Sci–Tech*, *Science & Space*, *Internet & Science*, and *Science*. It includes numerous topics such as architecture, astronomy, computers (hardware and software), electronics, engineering, IT (information technology), medical research, physics, science, and technology.

In addition to the general *Sci|Tech* category, we would like to pay particular attention to two important others, namely *Computers* and *Space*. A medical article, for instance, that does not belong to either of these sub-sections, will fall under the more general category *Sci|Tech*.

4. Health

News sources such as the HERALD SUN and the TIMES ONLINE EDITION combine categories *Science* and *Health* under one heading. In their design, a health document might[2] always be scientific. But this is not always the case. Therefore, scientific health documents are detected in the previous category *Sci|Tech*, and we represent both scientific and non-scientific health related documents in this seperate *Health* category. General topics include clinics and hospitals, doctors and nurses, diseases and infections, medical research and treatments, schools of medicine, as well as health advice and discussions.

5. Sports

This category gathers articles related to athletic events and activities, like marathons, the Olympics, the Formula 1, soccer tournaments, and the SuperBowl. It also accounts for the people who engage in such sportive activities. For instance, an article about the Salt Lake City bribery scandal is an example of what belongs to this category.

6. Arts|Entertainment

The noun "entertainment" is defined in WordNet as a diversion that holds the attention. And there are numerous entertaining and artistic events, such as arts and

[2]The categories could be combined for different reasons.

artists, art works and productions, exhibitions and collected works of art, authors and books, music and musicians, photographers and photography, actors and movies, plays and shows, ...

7. Natural Disasters

Natural disasters include droughts, earthquakes, floods, hurricanes, storms, thunderstorms, tornadoes, cyclones, and volcanic activities. Commonly, an article about a natural disaster will most likely make it in the news sources category *Top Stories*, for which we do not account. And without a *Natural Disasters* category, an article about an earthquake will be classified under *Business*, if it is about economic losses, *Science*, if it is full of technical terms, and *Politics*, if it is about political reprecussions. It is also highly possible that the article does not get assigned to any of these categories disclosing even less information about its content.

Including this category is more an illustration of our meta goal, rather than our resource-driven argument. After all, our approach suggests the future overcoming of unseen and worthy topics. We just happened to encounter that of natural disasters. Although we started off with general encompassing categories, we could expand our set however we choose; it is admissible to have the categories slightly resource-based.

8. Events

This second DUC-based category is the "rough" equivalent of news sources category *Top Stories*. A top story could be anything from a mere house dispute or torrential rains, to a major concert or political decision. We distinguish between two types of events [Over, 2003]:

Single Events Documents about a single event in any domain and created within at most a seven day window.

Multiple Events Documents about multiple distinct events of a single type (no limit on the time window).

Note that natural disasters are also considered single events. The purpose of this category is to specify that a document is about an event that has occured and taken place.

9. People

This category was also derived from the DUC 2002 categorization. Here, documents present biographical information, mainly about a single individual. However, not all articles are biographical per se; some are centered around interactions between more than one person.

3.1.3 Orthogonal Concepts-based Categories

The goal behind choosing an appropriate set of categories is to be able to detect the topic of incoming news articles. Although news sources provide accurate categories for the classification of their articles, their structure is not as useful as that of the 20 Newsgroups corpus. The categories of the latter encode more precise knowledge. For example, a medical and scientific article is stored under category *sci.med*. Another on operating systems is stored under *comp.os.ms-windows.misc*. Political articles are also differentiated: *talk.politics.guns*, *talk.politics.mideast*, and *talk.politics.misc*. This gives us the idea of creating a complete taxonomy of topics. Each node in the taxonomy will represent a topic and articles will be classified under some of these nodes depending on their content.

The idea is to start off with the above-mentioned categories and create relevant and appropriate sub-categories. For instance, in politics, we have issues pertaining to Norh America, Europe, the Middle East, terrorism, intelligence, nuclear weapons, the United Nations, and political figures, among others. Under category *People*, we could include artists, doctors, normal people, scientists, as well as political figures. Also if we were to sub-categorize *Science*, we would think of numerous topics such as architecture, computers, electronics, medicine, and space. And under sub-category medicine, we have doctors, nurses, hospitals, labs, and research. But this creates redundancy in our tree-structure; there are now several nodes that can hold an article about a doctor (*science.medicine.doctors* and *people.doctors*) and several about a political figure (*politics.people* and *people.politics*). These nodes follow different paths and are not on the same level. The question that follows is the required number of levels and the granularity of the structure.

In reality, the granularity of the structure only depends on the requirements of the application. However, deciding where to create these sub-categories is the problem. For instance, in order to have a feel of topic hierarchies, we look at the concept

of natural disasters, earthquakes in particular, in Google. They appear under two different branches of science:

1. Science>Earth Sciences>...>Earthquake>Seismicity Reports
2. Science>Technology>...>Earthquake Engineering

In Google groups, we have at least two more branches:

1. sci.geo.earthquake
2. alt.disasters.earthquake

However, in our design, we meant to associate earthquakes with *Natural Disasters*, rather than *Science*. This is only one illustration of the problem of allowing different paths. Furthermore, which path is to be returned by the classifier? All those that apply? And will category *Politics.people* be slightly different than that of *People.politics*? We do not wish to allow multiple inheritance, i.e., we are not only interested in the leaf category, rather the whole path the doucment falls under as crucial knowledge is encoded in the in its nodes. There are several similar questions one has to answer before attempting to draw the rough lines of the taxonomy.

Implementing this design requires us to makes use of statistical methods. However, our resources, like those of other researchers, do not allow us to develop a complicated and deeply nested version of this structure. We will therefore bind ourselves to a somewhat simpler hierarchy. One that does not allow the same instances across multiple paths. Thus, an article on natural disasters would only appear under the *Natural Disasters* category. Another on economy would appear under *Business*. However, what happens if an article is about the economy of the Bahamas after hurricane Frances? Would it be classified under *Natural Disasters* or *Business*? If we allow a multi-label classification, the article would be classified under both categories, yielding accurate information. For such results, we could use a k-NN classifier, which returns a ranking of categories to which the document can be assigned. A more useful approach is one that would also return a list of categories to which the document does not belong.

Instead of classifying articles over one comprehensive taxonomy, we go for multiple small and concise ones: micro-taxonomies. A document can be classified over each and we can have more sub-categories in these disjoint and independent trees than

we could have allowed in a single extensive one. Furthermore, we can allow as many little trees as we want. For every concept we introduce a standalone taxonomy. Note that our earlier design was meant to guarantee a proper and precise categorization of news articles. However, it is possible that we come across documents with previously unencountered topics. In this case, the structure should attribute the article to an *Empty* class, i.e., the article is not about any of these topics. What happens at the micro-taxonomy level? Assume an incoming text is about "Bill Gates" and "Windows XP." This would belong to potential categories *science.computers*, *science.people*, and *People*. However, this is not on natural disasters. So, it should be attributed to an *Empty* class within the micro-taxonomy for *Natural Disasters*, which goes to say that the article is *not* about this particular concept. We will refer to this *Empty* class at the micro-taxonomy level as the complementary category.

We create nine micro-taxonomies. Each is based on one of the adopted categories for news articles: arts, business, disasters, events, health, people, politics, science, and sports. Each micro-taxonomy will consist of two sets, the *positive* and the *negative*. The positive is that on which the micro-taxonomy is based. It can contain n sub-categories relevant to the topic. The negative set actually represents the complement concept under which articles will fall if they are not about the concept. Thus, we can have n positive categories, where $n \geq 1$, and only one complement category, i.e., we allow for each concept $n + 1$ sub-categories.

Thus, the above-mentioned article on "Bill Gates" and "Windows XP" will be classified over these nine micro-taxonomies, for each of which a distinct classifier will be built. We will call these nine classifiers: *Micro-Classifiers*. The article will belong to the positive set of the *People*, *Business*, and *Science* micro-taxonomies, and to the negative set of all the rest, ruling out their topics from the pool of possibilities.

In short, the classification of an article over these nine micro-classifiers yields a multi-labeled and a multi-dimensional categorization.

3.2 Statistical Text Classification

We have previously seen that there are three approaches to text classification: rule-based, machine learning, and non-learning. We will not consider the non-learning approach, however, we have yet to determine (1) which of the remaining two we

would like to adopt and (2) which method is best suited for our orthogonal concepts-based design.

3.2.1 Naive Bayes and Rainbow

Both the rule-based and the unsupervised learning approach are not well suited for our design.

The knowledge engineering approach holds the promise of accurate classification. However, its labor intensiveness causes a knowledge acquisition bottleneck. Furthermore, introducing new concepts demands a large amount of re-engineering effort, which is just not acceptable in an ubiquitous system environment.

Clustering algorithms, part of the non-learning approach, do not meet our requirements or goals, either. They attempt to group similar documents together instead of assigning one document to a cluster. We could supply the category with some labeled positive and negative examples in addition to the incoming document and depending with which it clusters, determine to which set it belongs. Several clusters would be created if the negative set consisted of various topics. This is not a major problem since the data is labeled. However, it beats our orthogonal concepts-based design. We could have simply chosen a linear set of categories and applied clustering algorithms for the classification of the news articles. On the other hand, if the negative set consisted of only one topic, we run the risk of either creating a cluster for the news article or assigning it to the wrong one.

On a different note, the machine learning approach is well suited for our design. Its most popular classifiers are are NB, TFIDF/Rocchio, SVM, and k-NN. In a series of experiments, [Ragas and Koster, 1998] observe that a Bayesian classifier continues learning until it makes almost no mistakes. However, training over 600 documents has negative effects as the performance curve either remains steady or drops. This is still more reliable than a TFIDF Rocchio-based classifier that stops learning after 150 documents and keeps a steady 15% error-rate.

On the other hand, the performance of both SVM and k-NN classifiers is significantly better than that of NB [Yang and Liu, 1999], and according to [Zhang and Oles, 2001], "Naive Bayes is consistently worse than the other algorithms", namely to regression methods and Support Vector Machines. Two clarifications need to be made. First, regression methods, like symbolic classifiers and neural networks, are

discarded due to their labor intensive nature. Second, in the presence of Naive Bayes, which is well suited for binary classification tasks, SVM and k-NN classifiers are not the best choice for our orthogonal concepts-based design. Also, Naive-Bayes is scalable to large volumes of data as it does not suffer from quadratic training time as Support Vector Machines do and it does not require us to customize our application by experimenting with different k parameters. Furthermore, Naive Bayes achieves a performance of 70–80 % and is "a core methodology in text classification, mainly due to its simplicity and computational efficiency" [Meretakis *et al.*, 2000].

An implementation for Naive Bayes, among other algorithms, is available in the *Bow* toolkit [McCallum, 1996]. The latter was designed and written, in 1996, by Andrew McCallum. It includes a library of C code, *libbow*, which is useful for writing statistical text analysis, language modeling, and information retrieval programs. The toolkit also provides front-ends for document classification (*Rainbow*), document retrieval (*Arrow*), and document clustering (*Crossbow*). Naturally, we were interested in *Rainbow*, the "executable program that does document classification." Although *Bow* was mostly designed for classification by Naive Bayes (NB), it supports other methods, namely, TFIDF/Rocchio, Maximum Entropy, Support Vector Machines (SVM), Probabilistic Indexing, and k-Nearest Neighbour (k-NN).

Naive Bayes falls under the inductive learning approach. In other words, there is the need for a training phase during which *Rainbow* reads tokenized documents, and as decribed by McCallum, a model containing their statistics is written to disk. The tokenization process consists of (1) reading in document d, (2) segmenting it into sections whose seperate identity is significant for categorization, (3) tokenizing each section that contains text, (4) optionally do stemming, (5) optionally use a stoplist, and (6) output a tokenized representation r of the document d from which the list of tokens in each selection can be determined [Zhang and Oles, 2001].

Rainbow provides us with several handy tokenization options. We opt for stemming, i.e., the extraction of the stem form, the use of the system's SMART[3] stoplist, which consists of 524 common words and avoids indexing unimportant terms, as well as a "skip-html" option, which skips all characters between "<" and ">" and is thus useful for lexing HTML files. In addition to tokenization, there is usually the possibility of feature selection. However, this is neither supported in *Bow* nor is it part of

[3]Simple Modular Architecture Research Tool

our requirements.

Our classification approach requires a distinct model to be built for each of the nine orthogonal concepts-based categories in our set. Once the models are built, Rainbow performs the classification task over the nine categories. It is worth mentioning that the same tokenizing procedure used over the training data should also, for consistency, be applied over the incoming testing data [Zhang and Oles, 2001]. However, before starting out with document tokenization, and for both training and testing phases, data needs to be provided. In the section that follows, we describe our training and testing corpora.

3.2.2 Training and Testing Corpora

Training and testing data should be similar in both format and structure. Exceptions put aside, when conducting their experiments in the field of text classification, researchers often use a re-known collection of news articles, such as Reuters, the Wall Street Journal, or the Associated Press Newswire, among others. The corpus is commonly split into two parts, one for training and another for testing. For our text summarization purposes, we were essentially interested in the categorization of the news articles that appeared in the DUC 2003 corpus. Therefore, we trained on a collection that resembled it: the DUC 2002 corpus. The latter consists of 567 news articles distributed over 59 groups, also called clusters.[4] Likewise, the DUC 2003 corpus consists of 60 directories and 624 documents. A topic is associated with each cluster and there are, on average, ten documents per cluster. The news sources from which the documents originated and their distributions are shown in Table 3.4.

However, the DUC 2002 corpus is neither sufficient in terms of training data nor in terms of topic coverage, but given the overhead in collecting and annotating data, we consider the 20 Newsgroups corpus provided with the Bow toolkit. The most important topics in the collection are related to computers, recreational activities, scientific domains, and political issues. Although the corpus does not support all categories covered by news articles, it serves as a supplement to a more reliable and comprehensive training corpus. A model built from the latter is capable of recognizing that an article about "Microsoft", for example, targets the corporation and not

[4]Although the corpus is said to contain 567 news articles, we have only worked with 533 available documents.

Source	DUC 2002	DUC 2003
Associated Press Newswire (AP)	332	266
Chinese Xinhua Newswire (XIE)	–	106
Financial Times (FT)	39	–
Foreign Broadcast Information Service (FBIS)	19	–
Los Angeles Times (LAT)	83	–
New York Times (NYT)	–	252
San Jose Mercury News (SJMN)	44	–
Wall Street Journal (WSJ)	16	–
Total Number of Documents	533	624

Table 3.4: News sources for DUC 2002 and DUC 2003

its softwares, i.e., BUSINESS rather than TECHNOLOGY or *comp.os.ms-windows.misc.* Also, that an article about the "SuperBowl" should belong to a more general sportive category and not to the available *rec.sports.baseball* or *rec.sports.hockey.*

Intuitively, the fact that the documents originated from different news sources should not affect the performance of the classifier as much as training on newsgroups and testing on news articles would; newsgroups messages and news articles differ in structure, length, and possibly vocabulary, as newsgroup messages could be less formal. However, this is overcome given the fact that our strategy is based on the "bag of words" approach, i.e., the training data will be stripped to its word-tokens and the essential ones that represent the category will be filtered.

Nonetheless, what might play a decisive role when it comes to testing and evaluation is the small number of training documents. Naive Bayes needs 600 training documents, per category, to reach perfection [Ragas and Koster, 1998]. As we have already established, the newsgroups will not be able to support all categories. For example, they do not account for disasters. Furthermore, some topics we would want to cover might not be present at all in the 533 available documents. Therefore, in some cases, we considered a collection of 290 WALL STREET JOURNAL texts to supplement the training set. Thus, our training corpus consists of the DUC 2002 corpus, the 20 Newsgroups, and a selected collection of texts from the WALL STREET JOURNAL.

Recall that a category consists of a positive set as well as a negative set. Although $P(d \in \overline{C}) = 1 - P(d \in C)$, the classifier needs to train on both positive and negative sets to realize that. Note that our choice of categories and their structure is influenced by the available data, i.e., to a certain extent, the design was resource-driven. The list of clusters used for training the different positive sets of categories is available in

Appendix C. However, it does not display the documents used to train the negative complement sets. We mention the following: (1) all DUC 2002 documents that are not used to train the positive set are used for the negative set and (2) if we don't use 20 Newsgroups or a Wall Street Journal articles in the positive set, we do not necessarily include them for negative training. For instance, the positive ARTS examples are selected from the DUC 2002 corpus, and the remaining DUC 2002 articles, as well as all 20 Newsgroups and WALL STREET JOURNAL ones, represent the negative examples and allow for training the complement category. On the other hand, the complement of category *politics* only uses articles from the DUC 2002 corpus.

Table 3.5 shows the number of documents used for training, and the subscripts N and W indicate training over 20 Newsgroups or WALL STREET JOURNAL documents, respectively. Note that the "*" next to some categories indicates that their positive set consists of several categories.

cat.	**Positive Set**	**Negative Set**
Politics	3216_N	641
Business	326_W	531
Science*	11056_N	11511_N
Health	1014_N	19556_N
Sports	2029_N	18537_N
Arts	83	20779_{WN}
Natural Disasters*	112	711_W
Events*	395	147
People	142	122

Table 3.5: Number of training documents for different categories

3.3 Multi-Dimensional Categorization

We have a total of nine disjoint micro-classifiers, each specialized in a distinct area, and the idea behind testing an article is to see whether it falls on the positive or negative side of each of these given areas. However, this design is not as rigid as one might think, since testing an article with the micro-classifier for category *Politics*, returns a probability x for the positive set, and y for the negative set. Here, $0 \leq x \leq 1$, $0 \leq y \leq 1$, and $x + y = 1$. Furthermore, $x = \sum_{i=1}^{n} x_i$, where x_i is the probability for the positive category i and n is the number of positive categories. For example, the probability x that document d is about a natural disaster is computed as follows:

$x = P_{disaster}(d) + P_{storm}(d) + P_{earthquake}(d) + P_{flood}(d)$. Also, the probability that the document is not about a natural disaster is simply $1 - x$.

This orthogonal concepts-based design allows for a multi-dimensional and multi-labeled categorization. An article can be represented in a k-dimensional space using any of the attributed positive or negative labels. In what follows, we consider the initial set of categories and models and attempt to evaluate the performance of the micro-classifiers, both in terms of precision and recall. By introducing appropriate changes, we will also see to what extent the structure of our categories and their training affect these results.

We test our orthogonal-based multi-labeled approach on the complete DUC 2003 corpus. The choice of data was highly dependent on the summarization application.

3.3.1 Gold Standard and Delta Matrix

In order to evaluate the performance of our nine micro-classifiers, we need to have a gold standard for comparison. In the community, there are none readily available for our particular task and so we manually developed our own.

As previously established, every article is individually reviewed and classified by each of the nine micro-classifiers, and is attributed to either the category or its complement. Given the criteria for articles to appear under certain categories, a human annotator reviews each document and assigns a probability of 1 if it belongs to the category and of 0 if it does not. At all times, the probability of the document belonging to the category plus that of its complement is equal to 1, i.e., the negative set gets the complement probability of the positive set.

The gold standard we build is representsed by a [624 x 24] (rows x columns) matrix, filled with 0's and 1's. Every row represents an entry for an article, and the columns represent the different categories, as well as their complements. Table 3.6 shows part of the gold standard classification for a "Microsoft on trial" article in the DUC 2003: d31033t cluster. For example, it reads that the document is about BUSINESS, PEOPLE, and COMPUTERS.[5] We exclude the entries for categories ARTS, DISASTERS, HEALTH, and SPORTS. The probability that the document belongs to any of these categories is null.

When developing the gold standard, and even when training, we are manually

[5]The translations for the letters are presented in Table 3.7.

...	B	NB	...	EM	ES	NE	...	P	NP	POL	NPOL	T	TC	TS	NT	...
...	1	0	...	0	1	0	...	1	0	0	1	0	1	0	0	...

Table 3.6: Gold standard entry for document NYT19981023.0251

assigning a probability of 0 or 1 to every category and its complement. However, this binary restriction is a double-edged sword. It allows us to be consistent and objective; an article either belongs to a category or does not. But when an article swings quarter or halfway between the positive and the negative set, we are faced with a dilemma. For instance, assume that 25% of an article on a natural disaster is about economical consequences. This would not necessarily entail the total attribution of the document to the BUSINESS category; assigning a probability of 1 is nonsensical. Else, when classifying a document only similar to the remaining 75%, we might get a false and high probability that the document is about business. Yet, by assigning a probability of 0, we rule out the business presence.

Ideally, we want both micro-classifiers for BUSINESS and NATURAL DISASTERS to recognize the relevant topic segments in the document and return the proper probability for both categories. For instance, 0.25 for business meaning that 25% of the article is about business. However, in reality, these segments overlap. Human annotators, due to their subjective nature, cannot remain consistent segmenting the text or assigning intermediate probabilities. This sacrifices the accuracy of our results when we compute the Delta Matrix. The latter is our third [624 x 24](rows x columns) matrix. It contains the difference between the micro-classifiers' output and the gold standard and it is necessary to compute accuracy, error, precision, and recall. All three matrices have the same format. Table 3.7 shows the entries for three articles in each of the three matrices.

The absolute values of the numbers in the delta matrix range between 0 and 1 and represent the difference existing between the gold standard classification and that of Rainbow. For example, the $|\delta_{[1,13]}| = 0.08$ tells us that classifying document APW19990519.0113 with the micro-classifier for category HEALTH differs by 0.08 from its gold standard categorization.[6] Low difference values indicate a closer classification to the gold standard, i.e., better results.

[6]The symbol Δ represents the delta matrix and δ an entry in the matrix.

cat.	APW19990519.0113			NYT19980610.0335			XIE19961212.0032		
	★	⋒	Δ	★	⋒	Δ	★	⋒	Δ
Arts (A)	0	0	0	0	0	0	0	0	0
Not Arts (NA)	1	1	0	1	1	0	1	1	0
Business (B)	0	0	0	0	0	0	0	1	1
Not Business (NB)	1	1	0	1	1	0	1	0	-1
Disaster (D)	0	0	0	0	0	0	0	0	0
Disaster.Earthquake (DE)	0	0	0	0	0	0	0	0	0
Disaster.Flood (DF)	0	0	0	0	0	0	0	0	0
Disaster.Storm (DS)	0	0	0	0	0	0	0	0	0
Not Disaster (ND)	1	1	0	1	1	0	1	1	0
Event.Single (ES)	1	1	0	0	0.19	-0.24	0	0.94	0.94
Event.Multiple (EM)	0	0	0	1	0.76	0.19	1	0.06	-0.94
Not Event (NE)	0	0	0	0	0.05	0.05	0	0	0
Health (H)	1	1	0	0	0	0	1	1	0
Not Health (NH)	0	0	0	1	1	0	0	0	0
People (P)	1	0.92	-0.08	1	1	0	0	0.31	0.31
Not People (NP)	0	0.08	0.08	0	0	0	1	0.69	-0.31
Politics (Pol)	0	0	0	1	0	-1	0	0	0
Not Politics (NPol)	1	1	0	0	1	1	1	1	0
Technology (T)	1	1	0	0	0	0	0	0.12	0.12
Technology.Computers (TC)	0	0	0	0	0	0	0	0	0
Technology.Space (TS)	0	0	0	0	0	0	0	0	0
Not Technology (NT)	0	0	0	1	1	0	1	0.88	-0.12
Sports (S)	0	0	0	0	0	0	0	0	0
Not Sports (NS)	1	1	0	1	1	0	1	1	0

Table 3.7: Gold-standard (★), rainbow (⋒), and Δ-matrix entries for three documents

3.3.2 Difference and Accuracy

The delta matrix gives us the means to compute precision and recall for the micro-classifiers. But before looking at the results of these evaluation metrics, we report some firsthand observations on (1) difference distribution across documents, categories, and sources, and (2) category accuracy.

1. Documents

The classification of 26.12% of the documents in the test set is equal to that of the gold standard for all categories. We say that these documents achieve an overall null difference since all their delta matrix fields are null. For instance, document NYT19980610.0335, in Table 3.7, does not have an overall null difference. For two of its categories, EVENTS and POLITICS, the automatic classification was not as expected. The worst classification of a document differs from that of the gold standard in four categories. However, this seldom happens. More often, a document is classified

no. cat.	0	1	2	3	4
doc.	26.12%	45.67%	22.76%	4.97%	0.48%

Table 3.8: Frequency of documents and number of categories for which $|\delta| > 0$

differently over one or two categories. The distribution is represented in Table 3.8.

Although we are concerned with the proper multi-label categorization of an article over all the micro-classifiers, it is nonsensical to look at its average difference across all categories. The latter, which ranges between 0 and 0.11, does not indicate with which order the automatic classification differs, when it does, from that of the gold standard. Instead, we consider a collection of data based on the average non-zero difference per document. It is obtained by summing the differences that occur for every category in which $|\delta| > 0$ and dividing by the number of categories, which ranges between 1 and 4. This is simple in the case of binary categories such as PEOPLE, POLITICS, and SPORTS, where $P(cat) = 1 - P(\overline{cat})$ and $|\delta(cat)| = |\delta(\overline{cat})|$. In the case of non-binary categories, like DISASTERS, EVENTS, and SCIENCE, that have multiple positive categories, the probability is distributed over one or more categories (the positive set, the negative, or a combination of both) and the complement is the sum of the probabilities from the remaining categories. Table 3.7 shows the classification results for document XIE19961212.0032, where $P(T) + P(NT) = 1 - (P(TC) + P(TS))$. Here, $|\delta(T) + \delta(TC)| = |\delta(TS) + \delta(NT)|$. We consider the maximum difference in absolute value. For example, document APW19991020.0140 has the following entry for category EVENTS in the delta matrix: $\delta(SE) = -0.45$, $\delta(ME) = 0.19$, and $\delta(NE) = 0.26$. When computing the average difference for the document and later when computing that of the category, we choose $max|(\delta_i)|$, in this case $|\delta(SE)|$, which is equal to the sum of the remaining others.

At this point, the idea is to look at the performance of the category as a whole as it compares to others, rather than account for its macro-differences over the multiple positive categories as well as the negative one.

Table 3.9 shows the average difference ranges and the frequency of documents occuring for each. More than half of the set has an absolute average difference ≥ 0.8, which is to say, the micro-classifiers behaved, at least, 80% differently than expected; either assigning a document to a category where it does not belong or vice versa. There is also a large percentage of documents (33.49%) that cluster at small average difference values.

range	0–0.19	0.2–0.39	0.4–0.59	0.6–0.79	0.8–0.99	1
doc.	33.49%	0.96%	7.36%	4%	8.17%	45.99%

Table 3.9: Frequency of documents for average difference ranges

2. Categories

	Arts	Business	Disasters	Events	Health	People	Politics	Sci|Tech	Sports
doc. freq.	1.92%	8.81%	3.84%	39.74%	3.20%	15.22%	24.52%	6.57%	2.72%
accuracy	98.08%	91.19%	96.16%	60.26%	96.8%	84.78%	75.48%	93.43%	97.28%
avg. diff	0.87	0.78	0.95	0.83	0.95	0.85	0.85	0.84	0.94

Table 3.10: Accuracy and average difference for all categories

Table 3.10 shows the percentage of test documents whose automatic classification is not equal to that of the gold standard. The lower the percentage of documents with $|\delta| > 0$ for a category, the better is the accuracy of its micro-classifier.[7] The latter is computed by summing the number of documents for which $\delta = 0$ and dividing by the total number of documents in the test set. For instance, we can deduce that categories *Arts*, *Sports*, *Disasters*, *Sci|Tech*, and *Business*, will have higher accuracies than categories *People*, *Politics*, and *Events*. Accuracy results are also reported in Table 3.10.

When a category has multiple positive sets (e.g., *Disasters* and *Science*), we obtain different accuracy measures for each of the positive or negative sub-categories. Here, it is not as simple as $\delta(cat) = \delta(\overline{cat})$. We choose the smallest accuracy value as a representative of the category, i.e., pick $min(a_p, a_n)$, where a_p and a_n are the accuracies for the positive and negative sets, respectively, and $a_p = min(a_{p1}, a_{p2}, .., a_{pn})$. For example, in Table 3.11, the accuracy for category *Disasters* is obtained as follows: $min(a_p, a_n) = min(min(100, 99, 97, 96), 96) = min(96, 96) = 96$.

	Positive				**Negative**
sub-category	Dis	Dis.Earthquake	Dis.Flood	Dis.Storm	Not Disaster
accuracy	1	1	.979	.971	.963

Table 3.11: Accuracy for category *Disasters*

Furthermore, for documents with $|\delta| > 0$, we compute the average difference with

[7] In reality, when computing the accuracy for a cateogry we look at $|\delta| > 0.05$.

which the micro-classifiers behave differently than expected.[8] The results are also reported in Table 3.10.

Range	Arts	Business	Disaster	Events	Health	People	Politics	Sci|Tech	Sports
.01-.09		5	1	24	1	10	10	3	1
.10-.19	1			5		2	6	3	
.20-.29		2		5			1		
.30-.39	1	2		2		3	2		
.40-.49				3			2		
.50-.59		1		4		1	2	1	
.60-.69		1		1		1	1		
.70-.79		1		2		1	1	1	
.80-.89		2		5		1	3	2	
.90-.99		6	1	21		9	10	1	
1.00	10	35	22	176	19	68	114	30	16
Total	12	55	24	250	19	95	153	41	17

Table 3.12: Document frequency and difference ranges across categories

A low frequency of documents with $|\delta| > 0$ for a category indicates the decent performance of its micro-classifier. Intuitively, we would expect the best-performing micro-classifiers to have a low average difference. However, this does not always need to be the case. What is interesting to see is that high document frequency correlates with a low average difference and vice-versa. For instance, categories *Disasters* and *Sci|Tech* have the highest average difference, 0.95 and 0.94, respectively. On the other hand, categories with worse performing micro-classifiers, like *Events*, *People*, and *Politics*, have some of the lowest average difference, 0.83, 0.85, and 0.85, respectively.

Indeed, Table 3.12 shows a clustering around high-pitched difference values in well performing categories, while we have a more even and balanced distribution in the worst performing categories. It seems that the more uniform the distribution of non-zero difference, the worse the performance of the category. Given this observation, we can use the table to rank categories in terms of descending performance. Note that we are measuring performance in terms of accuracy. In descending order, these would be: *Event*, *Politics*, *People*, *Business*, *Sci|Tech*, *Disaster*, *Arts*, and finally, *Sports*.

3. Sources

The DUC 2003 corpus consists of articles from three different news sources: 42.62% of the documents are provided by the Associated Press Newswire (APW), 40.38% by

[8]Since we do not account for the average difference of a document across all categories, we also discard looking at the average difference of a category across all documents.

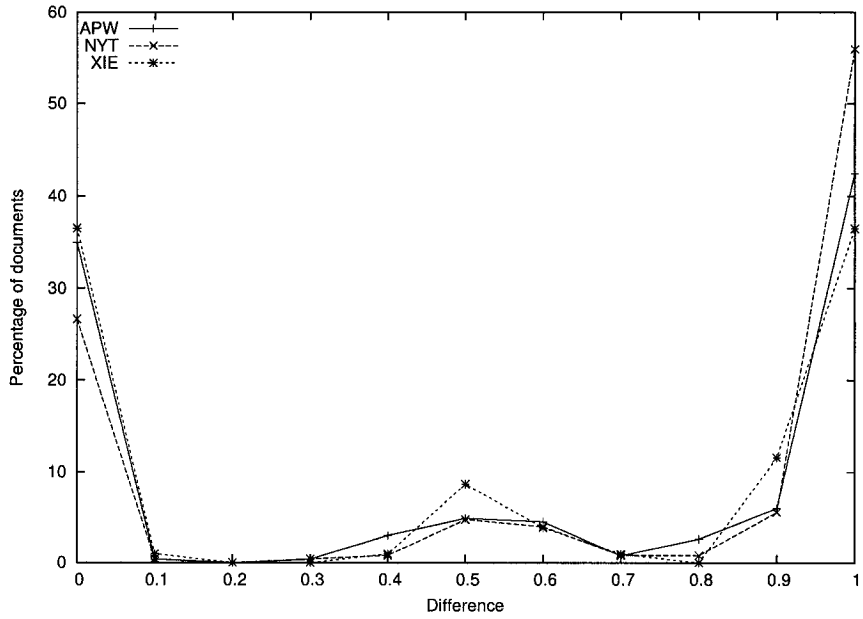

Figure 3.1: Distribution of difference per news source overall documents

the New York Times (NYT), and 16.98% by the Chinese Xinuha Newswire (XIE). We plot, for every news source, the difference distribution over all documents, in order to verify whether or not the classification is source sensitive. The similar patterns in Figure 3.1 show that there are no significant differences across news sources.

3.3.3 Precision and Recall

Since it has been established that accuracy and error are insensitive to variations [Jackson and Moulinier, 2002], we choose to evaluate the performance of each micro-classifier using alternate and more reliable metrics, such as precision and recall. We compute these for four different thresholds. A document is assigned to a category if the absolute difference between its gold standard and its automatic classification is ≤ 0.4, ≤ 0.3, ≤ 0.2, and ≤ 0.1, i.e., the probability assigned by Rainbow must be $\geq 60\%$, $\geq 70\%$, $\geq 80\%$, and $\geq 90\%$, respectively.

Precision is the fraction of documents found and correct over the total documents found, and recall is the fraction of documents found and correct over the total documents correct. For example, assume we want to evaluate the performance of the *Health* micro-classifier:

1. Our test set consists of five documents: d1, d2, d3, d4, and d5.

2. In our gold standard, d2, d4, and d5, are health documents, and

3. the micro-classifier assigned d1 and d4 to category *Health*.

The document found and correct is d4. The total documents found are d1 and d4. The total documents correct are d2, d4, and d5. Thus, precision and recall are: $p = \frac{\text{documents found and correct}}{\text{total documents found}} = \frac{1}{2}$ and $r = \frac{\text{documents found and correct}}{\text{total documents correct}} = \frac{1}{3}$.

The low precision means attributing the wrong documents to the category, while low recall means missing out on documents that should belong to the category.

In reality, we look at the gold standard to determine which documents belong to the category, i.e., the total documents correct. For each of these documents, if $\delta \geq \Theta$, then we increment the number of found and correct documents. Similarly, we increment the number of documents found whenever the automatic classification $d_{\text{m}} \geq x$. The values for Θ and x are as follows:

1. $\Theta = 0.1$, $x = 90\%$,

2. $\Theta = 0.2$, $x = 80\%$,

3. $\Theta = 0.3$, $x = 70\%$, and

4. $\Theta = 0.4$, $x = 60\%$.

Table 3.13 shows the results of the micro-classifiers for our initial set of categories. In the table and the analysis that follows, we consider δ to be in absolute values.

1. Arts

There were 10 doccuments in the DUC 2003 test set about cartoonist Charles Schultz, his death, and his work. In the gold standard, these articles were attributed to category *Arts*. However, the micro-classifier did not recognize any of them. Instead, it nearly assigned an article about Pinochet, the former Chilean dictator, to this category. There are two things to be said. First, although the category was mostly trained on artistic persons, the micro-classifier did not attribute any document about a person to *Arts*, rather to its complement. This is probably due to the solid training of the latter, which furthermore achieves high measures of precision (0.983) and recall (0.996). Note that precision is

Categories	$\|\delta\| \leq 0.1$		$\|\delta\| \leq 0.2$		$\|\delta\| \leq 0.3$		$\|\delta\| \leq 0.4$	
	Prec.	**Recall**	**Prec.**	**Recall**	**Prec.**	**Recall**	**Prec.**	**Recall**
Arts	0	0	0	0	0	0	0	0
Not Arts	0.983	0.996	0.983	0.998	0.983	0.998	0.983	1
Business	0.747	0.893	0.736	0.893	0.730	0.893	0.730	0.893
Not Business	0.981	0.925	0.981	0.929	0.981	0.929	0.979	0.932
Disaster	1	1	1	1	1	1	1	1
Disaster.Earthquake	1	1	1	1	1	1	1	1
Disaster.Flood	1	0.434	1	0.434	1	0.434	1	0.434
Disaster.Storm	0.833	0.606	0.833	0.606	0.833	0.606	0.833	0.606
Not Disaster	0.967	0.994	0.965	0.994	0.965	0.994	0.965	0.994
Event.Single	0.882	0.513	0.884	0.526	0.886	0.532	0.882	0.535
Event.Multiple	0.684	0.729	0.680	0.733	0.680	0.742	0.678	0.746
Not Event	0.433	0.918	0.417	0.918	0.407	0.918	0.404	0.918
Health	0.961	0.581	0.961	0.581	0.961	0.581	0.961	0.581
Not Health	0.971	0.998	0.971	0.998	0.971	0.998	0.971	0.998
People	0.969	0.863	0.966	0.865	0.966	0.865	0.967	0.869
Not People	0.615	0.863	0.615	0.863	0.612	0.863	0.610	0.871
Politics	0.951	0.541	0.941	0.559	0.947	0.565	0.942	0.569
Not Politics	0.736	0.967	0.735	0.970	0.732	0.970	0.730	0.970
Technology	0.526	0.833	0.454	0.833	0.434	0.833	0.416	0.833
Sci|Tech.Computers	0	0	0	0	0	0	0	0
Sci|Tech.Space	0.767	1	0.767	1	0.767	1	0.767	1
Not Sci|Tech	0.981	0.954	0.982	0.959	0.982	0.959	0.982	0.959
Sports	1	0.542	1	0.542	1	0.542	1	0.542
Not Sports	0.975	1	0.975	1	0.975	1	0.975	1

Table 3.13: Precision and recall

constant for this category, for the different thresholds, while recall increases until it reaches 100% at $\delta \leq 0.4$. Second, better and more training data would have improved the poor performance of *Arts* since 83 documents are too few and would have worked better with a k-NN classifier.

2. Business

The best precision and recall values for category *Business* are 0.747 and 0.893, respectively, and are obtained for $\delta \leq 0.1$. Precision decreases as δ increases while recall remains fixed. Some of the topics that caused this 0.747 precision include "the Salt Lake city bribery", "E-coli", "the capital punishment", "banning cell phones usage in cars", "Philippine's typhoons", and "Egyptian interpol." Although these were attributed to *Business* in the automatic classification but not in the manual, we are not necessarily looking at a misclassification. On the contrary, the micro-classifier was able to recognize the business sections in these articles. Recall that a human annotator, in a similar situation, would only

assign 1 if the document was mostly about business.

For the complement category, precision and recall are consistently high, 0.981 and 0.925. They vary lightly across the different values of δ. We can notice the trade-off between both for $\delta \leq 0.1$ and $\delta \leq 0.4$; when precision drops from 0.981 to 0.979, recall jumps from 0.925 to 0.932. There were 531 documents used for training the complement category. This number is close to the 600 documents needed for a Bayesian classifier to reach perfection.

3. Disasters

In this category, there are multiple positive sets: general disasters, earthquake disasters, floods, and storms. For *Disaster* and *Dis.Earthquake* there are no disasters or earthquakes to be found in our test set, and the micro-classifier does not assign wrong documents to the category. For *Dis.Flood*, precision is also 100% but the micro-classifier recognizes less than half the documents that are about a flooding disaster. There is a 20% increase in recall for *Dis.Storm* but precision drops to 0.833. Here, better recall is probably due to the larger number of training instances used. However, for both sub-categories *Dis.Flood* and *Dis.storm*, documents about "Storms and floods in the Mozambique" were not recognized. There are no changes to be noted for the measures of precision and recall for different values of δ.

When it comes to the complement category, best results are obtained for $\delta \leq 0.1$; precision is 0.967 and recall 0.994. Precision decreases by an order of 0.002 when we increase δ but recall remains fixed.

4. Events

The optimal r-cut overall branches for this category is 0.4. There are different r-cuts for the different branches, 0.3 for *Event.single*, 0.4 for *Event.multiple*, and 0.1 for *Not Event*. The best precision is recorded for *Event.Single*, followed by *Event.Multiple*, and *Not Event*. That of the latter is very low (0.404). Recall behaves in exactly the opposite fashion. It is low (0.535) for *Event.Single*, higher (0.746) for *Event.Multiple*, and very decent (0.918) for *Not Event*.

5. Health

The micro-classifier for category *Health* is consistent in its performance over the different values of Θ. We obtain for the positive set a precision of 0.961 and a recall of 0.581. Of the 42 documents pertaining to this category, only those about "Schizophrenia" and "E-coli" were detected. Only 10% of the rest, which were mainly news about doctors without borders and abortion doctors, were classified in *Health*. The category, which has a large number of training instances, seems to be well trained to recognize an illness but not people or events relevant to the domain.

The precision for the complementary category is nearly equal to that of the positive set (1% difference). Recall, 0.998, is nearly perfect.

6. People

Precision for this category's positive set is among the highest ones recorded so far (0.969). Also, like the *Events* micro-classifier, the precision of its negative set is low. Most documents about people are attributed to the negative set. Furthermore, this is the only category in which the measures of recall for the positive set are very similar to those of the negative set, 0.863 and 0.865, and 0.869 and 0.871, respectively.

7. Politics

The micro-classifier for *Politics* scores high on precision for the positive set but low on recall. Mostly, articles relating political discussions, opinions, elections, and stories about political figures, were not detected. Topics include "Tribal tensions in Africa", "Libyan problems", "Czech republic joins NATO", "Capital punishment", "U.S. presidential elections", "Brazilian elections", "Pinochet", and "Monica Lewinsky and Bill Clinton". A reason why the latter was not detected is because the classifier has not been trained to recognize that the names "Bill Clinton" and "Monica Lewinsky" belong to the political scene; either the information became available post-training or the resources were not present altogether. Adding the information will not make a significant difference since Naive Bayes trains on word occurrences.

On the other hand, the performance of the complement category is low in terms of precision but high in recall. The best measures of precision and recall are recorded for $\delta < 0.3$.

8. Sci|Tech

The performance of the *Sci|Tech* micro-classifier scores very high on recall, unlike the low precision value for the general *Technology* category. However, this is again due to the strict binary assignment manual annotators are binded to. Articles on "E-coli", "Microsoft on trial", and "Y2K", were automatically attributed to this category rather than manually. In reality, there is a fair scientific percentage in these articles. For instance, we manually attributed "Microsoft on trial" articles to category *Sci|Tech.Computers*. This was not detected and since there were no other scientific and computer-based articles, both precision and recall for this branch were null. SCI|TECH.SPACE has a perfect recall and an average performance of 0.767 because it picked up articles about "the construction of three gorges dam in China."

Similarly to most complement categories, that of *Sci|Tech* scores very high for both precision and recall. The best measures are recorded for $\delta < 0.1$.

9. Sports

The micro-classifier for category *Sports*, like that of *Health*, is consistent in its performance over the different values of Θ. Precision for the positive set is perfect but recall is 0.542. Articles about the "Rosebowl" event and very few on the "Asian games", were assigned to category *Sports*. The micro-classifier missed most of the remaining "Asian games" documents and those on the Salt Lake City bribery scandal. Again, there is a weakness when meta-material is to be detected. This can only be solved with proper training. For instance, using the Asian games documents to train the *Sports* category will improve the performance of its micro-classifier, in the future.

The precision and recall for the negative set are very high, 0.975 and 1, respectively.

Table 3.14 summmarizes our observations for precision and recall. The "T" marks the best values for precision and recall, i.e., $\geq$ 90%, and the asterisk marks values $\geq$ 80%. We would like to point out that for all the complementary categories, which have very high values of precision and recall, i.e., at least 90%, and to the exception of *Business*, which only counts 531 negative training documents, we have a very large number of training instances (in the thousands).

The table also summarizes the r-cut values for the initial set of categories. The micro-classifiers for categories *Health* and *Sports* have a constant performance over the different values of Θ. We will consider the default to be $\Theta = 0.1$. Note that the differences are not very significant. In case we would like to consider one threshold for all categories, we should consider $\Theta = 0.1$. Else, we can pick the best threshold for each category. In later sections, we will look at the f_1 measures for the different categories in order to determine more accurately the best value of Θ.

	Positive		**Negative**		**R-cut**
cat.	prec.	recall	prec.	recall	**threshold**
Arts			⊤	⊤	0.4
Business		⊤	⊤	⊤	0.1
Disasters	⊤		⊤	⊤	0.1
Events			⊤	⊤	0.3
Health	⊤		⊤	⊤	const.
People	⊤	⊤		*	0.4
Politics	⊤			⊤	0.3
Sci|Tech		⊤	⊤	⊤	0.1
Sports	⊤		⊤	⊤	const.

Table 3.14: Categories with best precision and recall

3.3.4 Data Sensitivity

We have seen that the performance of the micro-classifiers is highly dependent on the training data. The question is: Does the testing data have any effect on the results of the classification? In order to answer this, we experiment with lengthier documents to see whether or not the measures of precision and recall are affected.

The DUC 2003 corpus consists of 624 documents distributed over 60 clusters. News articles belonging to the same cluster relate common events and are concatenated to form one long document, i.e., there are now 60 instances in the testing corpus. The training data is left unaltered and we test on the initial set of categories and their structure.

1. Difference

The [624 x 24] (rows x columns) delta-matrix obtained for multi-documents is very different from that which was computed for short-documents; it consists mostly of 0s

or 1s. This is because, in all cases, the micro-classifiers either assign a probability of 1 or 0. There are no intermediate rankings; this demonstrates a degree of certainty given the amount of data tested. Furthermore, in 56.7% of the cases, there is a misclassification of a document over one or two categories; the worst classification of a document differs from that of the gold standard in two categories only, as opposed to four, which we obtained when classifying shorter documents. Note that, here, a long document is, on average, ten times the length of a short document. Similarly, in 43.3% of the cases, which is again 1.65 times more than what has been gathered for shorter documents, we have a 100% correct classification.

2. Accuracy

We record for 98.3% of the documents an accuracy above 80%. In particular, 85% of the documents achieve at least 90% accuracy, which is 15% more than what has been obtained for short documents.

cat.	Arts	Business	Disasters	Events	Health	People	Politics	Sci|Tech	Sports
short	0.98	0.91	0.96	0.590	0.96	0.84	0.75	0.93	0.97
long	0.98	0.93	0.98	0.80	0.96	0.98	0.88	0.83	0.95

Table 3.15: Accuracy for short and long documents

Table 3.15 displays the accuracy results for both short and long documents. We observe that there is:

1. No accuracy increase for categories *Arts* and *Health.*
2. Low accuracy increase (1%-4%) for categories *Business*, *Disaster*, and *Sports.*
3. High accuracy increase (8%-14%) for categories *Events*, *People*, and *Politics.*
4. Accuracy decrease (0.93%-0.83%) for category *Science*

Categories that a priori perform well have little margin for improvement while those that do less well have a much greater one. This is similar to overtraining a classifier. At one point more training data will make no difference if not worsen the performance. Also, in our case, lengthening the documents proved to be efficient. It allowed the classifier to be more accurate. However, this could have also backfired and the news articles might have as well been classified under many false categories.

Categories	short: best r-cut		long: all r-cut	
	Precision	Recall	Precision	Recall
Arts	0	0	0	0
Not Arts	0.983	1	0.983	1
Business	0.747	0.893	0.714	1
Not Business	0.981	0.925	1	0.92
Disaster	1	1	1	1
Disaster.Earthquake	1	1	1	1
Disaster.Flood	1	0.434	1	0.5
Disaster.Storm	0.833	0.606	1	1
Not Disaster	0.967	0.994	0.982	1
Event.Single	0.886	0.532	0.904	0.633
Event.Multiple	0.680	0.742	0.678	0.826
Not Event	0.407	0.918	0.411	1
Health	0.961	0.581	1	0.5
Not Health	0.971	0.998	0.965	1
People	0.969	0.863	0.955	0.877
Not People	0.615	0.863	0.588	0.909
Politics	0.947	0.565	0.95	0.655
Not Politics	0.732	0.970	0.738	1
Technology	0.526	0.833	0.5	1
Sci|Tech.Computers	0	0	0	0
Sci|Tech.Space	0.767	1	0.75	1
Not Sci|Tech	0.981	0.954	0.981	0.963
Sports	1	0.542	1	0.666
Not Sports	0.975	1	0.982	1

Table 3.16: Precision and recall for long documents

3. Precision and recall

We compare the precision and recall obtained for long documents to those obtained at different r-cut thresholds, in the previous section for short documents. There are no significant differences to be noted for categories *Business*, *Health*, *Politics*, *Sci|Tech*, and *Sports*. In general, the measures of precision and recall slightly increase and reach total precision or recall. For category *Events*, these measures increase more obviously.

We also observe that there is no change whatsoever for category *Arts*, but for categories *Dis.Storm* and *Dis.Flood*, the measures of precision and recall increase. The *Disasters* micro-classifier is trained to recognize events, such as "Storms and floods in the Mozambique." However, the single documents in this cluster did not contain enough evidence to be attributed to *Dis.Storm* and *Dis.Flood*. But, when combined together, they did. On the other hand, the *Arts* micro-classifier does not seem to be trained properly to recognize articles on "Charles Schultz." Therefore, there is no change no matter how long the documents is.

On a different note, the precision for both positive and negative sets of category *People* slightly decreases with this length.

3.3.5 Structure Sensitivity

We turn our attention to the training data and the structure of the categories and determine if and how these two affect the performance of the classifiers.

Most of the categories have the same structure; they are trained over one set of positive examples and one set of negative examples. Exceptionally, categories *Disasters*, *Events*, and *Sci|Tech*, consist of multiple positive categories (partly due to the availability of the training data) as well as the complementary ones:

1. DISASTERS: Disaster, Dis.Earthquake, Dis.Flood, Dis.Storm, Not Disaster;
2. EVENTS: Multiple Event, Single Event, Not Event;
3. SCI|TECH: Technology, Sci|Tech.Computers, Sci|Tech.Space, Not Technology;

The idea is to look at these categories, which have low values of precision, recall, or both, and check whether or not re-structuring them by editing (adding or deleting) their sub-categories and re-distributing the data, results in a better or worse classification. Furthermore, we will consider modifying category *Politics* for which positive recall is 0.541. We note that for every new set of categories and automatic classification we run, we re-build an appropriate gold standard from which we compute the delta matrix to examine the results.

1. Disasters

Unlike all other *Disasters* sub-categories, for which the values of precision and recall are very high, the recall for *Disasters.Storms* and *Disasters.Floods* is 0.434 and 0.606, respectively. There are no alternate ways of sub-categorization or training data re-distribution that would yield a better performance. The only possible structure is one that consists of a single positive set, i.e., *Disaster* and *Not Disaster*. The training can be found in Table D.1, in Appendix D.

The accuracy of the binary structure is 95.99%. For the positive set, this structure results in a higher recall (0.660) than the lowest one obtained for the orignial

set. Also, precision is better than the minimum precision, which was recorded for *Disasters.Storm*. Table 3.17 displays these differences.

	Original Set					**Binary Set**	
	Disaster	D.Earthquake	D.Storm	D.Flood	Not Disaster	Disaster	Not Disaster
accuracy	1	1	.971	.979	.963	.959	.959
precision	1	1	.833	1	.967	.853	.970
recall	1	1	.434	.606	.994	.660	.985

Table 3.17: Differences in performance for *Disasters*

2. Events

Unlike the other DUC categories, *Events*, with its ambiguous definitions, is hard to learn, and consequently, classify. They distinguished between (1) *Single Events* and (2) *Multiple Events*. Documents belonging to (1) are about an event that occurs in any domain and created within at most a seven day window, like natural disasters. Documents in (2) group multiple distinct events of single type without a limit on the time window. It was hard to realize which *Events* sub-category a document belonged to when constructing the gold standard matrix. Unlike the DUC 2003 clusters, the DUC 2004 ones weren't labeled and judging documents independently from their cluster is not simple in terms of deciding whether or not the news article revolves around one or several main events.

So, it does not necessarily come as a surprise that the classifier also has a hard time deciding. We look for improvement in defining another sub-category structure based on time-stamps and re-distribute the training data accordingly:

1. MULTIPLE TIME-STAMPED EVENT (E.MTS): This category is very similar to that of *Multiple Events*. Here, documents group multiple events that occur at different points in time and that pertain to the same topic ("Mc Donald's in Moscow", "German Reunification", ...)

2. SINGLE TIME-STAMPED EVENT (E.STS): The document is about a main event that occured at a particular point in time (a natural disaster, a terrorist attack, ...) and its ramifications, i.e., the discussions that pertain to it without a limit on the time window that we previously had in *Single Events*

3. NOT TIME-STAMPED EVENT (E.NTS): As opposed to *Single Events* and *Multiple Events*, in which events occur at explicit, particular, and defined points in time, this newly introduced concept covers events that do not have a specific time-stamp. In other words, unlike in *Single Events*, the reader does not identify *when* an event has occurred, rather that it *has* occurred and is possibly still taking place. These documents are characterized by their past and progressive tense. Documents of this type are typically political such as "Chinese manifestations", "Philippine's socio-politico-economy", "Trouble reports for India and Kashmir", etc.

4. NOT EVENT: Documents in which we cannot distinguish an event.

The training data repartition can be found in Table D.2 in Appendix D. The maximum accuracy obtained for the modified set is 85.25% (recorded for the E.STS category) and the minimum is 61.85% (recorded for both the E.MTS and NOT EVENT categories).

In terms of precision and recall, the performance of these categories is by far worse than that of the original set. The highest values reached (70%) are recorded for recall of the E.STS and the complementary category. All other values are insignificant.

One problem with the EVENTS-classifier is more its inability to distinguish between single and multiple events rather than recognize whether a text describes an event or not. This could be due to our use of stemming which treats all noun-forms similarly. The authors in [Riloff, 1995] argue that plural nouns denote general events while singular nouns denote specific events.

Since over-refining the categories hurts the performance, we stress on simplicity and train only for binary concepts, i.e., sub-categories EVENT and NOT EVENT, as in BUSINESS, PEOPLE, and SPORTS. The accuracy (62.5%) is overall lower than both the original and modified sets. The values for precision and recall are much higher than those reported for the modified set of categories, to the exception of the complemtary precision, which is as low as 23.5%. In fact, the positive category performs better than the negative one. Precision reaches 97.4%, but recall only 61.5%. The performances of the original, modified, and binary set of categories for EVENTS are reported in Table 3.18.

	Original Set			Modified Set				Binary Set	
	single	mult.	not event	e.sts	e.mts	e.nts	not event	event	not event
accuracy	.751	.698	.807	.852	.618	.748	.618	.625	.625
precision	0.882	0.684	0.433	0.217	0.377	0.005	0.160	0.974	0.235
recall	0.513	0.729	0.918	0.709	0.419	0.006	0.702	0.615	0.777

Table 3.18: Differences in performance for *Events*

	Original Set		Modified Set			
	Pol.	Not Pol.	Pol.world	Pol.terrorism	Pol.misc	Not Pol
accuracy	.767	.767	.794	.932	.860	.647
precision	0.947	0.732	0.236	0	0.199	0.552
recall	0.565	0.970	0.362	0	0.406	0.896

Table 3.19: Differences in performance for *Politics*

3. Politics

The micro-classifier for category *Politics* scores very high on precision for the positive set (0.947), but very low on recall (0.565). Most political documents are not detected. The problem could be due to the variety of political articles. We examine the performance of the micro-classifier after adding new branches: *Pol. World*, *Pol. Terrorism*, and *Pol.Miscellaneous*. The training data distribution is displayed in Table D.3, in Appendix D.

Articles pertaining to political figures are considered miscellaneous. The multi-dimensional classification will detect that these documents are about politicians.

The accuracy of the modified category is higher than that of the original set. However, it is lower for the binary set. The results for this sub-categorization are much worse than those obtained for the original one. The only decent performance of this micro-classifier is recorded for the recall of the negative category (0.896). All other measures are below 55%. None of the terrorism articles were recognized, i.e., "Bombings", "Explosions in Israel", and "Terrorist attacks on U.S. embassies." To verify how sensitive the micro-classifier can be to the training instances, some of the clusters used to train the initial category *Politics* were not used for this modified one. Topics include: "East Germany under Honeker", "Margaret Thatcher retirement", "Soviet Sakharov", "Japan's emperor", "Mandela's imprisonment", and "USSR politics." Table 3.19 displays the differences between the performances of both the original and the modified *Politics* categorization.

4. Sci|Tech

We revisit the idea of training sensitivity in the context of category *Sci|Tech*. Recall the redundant data in its general *Technology* category; all scientific documents are used for training. The sub-category somehow acts as a parent. The idea is to allow a document that doesn't fall under *Sci|Tech.Computers* or *Sci|Tech.Space*, to still belong to category *Sci|Tech*. Recall is high, however, precision for this branch is only 0.541. In order to check whether deleting the redundant data makes a difference, we train a second time, restricting documents of certain types to their relevant sub-categories, i.e., the training is similar to that of the original *Disasters* category. Since specific training data was available, we added categories *Sci|T.electronics* and *Sci|Tech.Medicine*. The training can be found in Table D.4, in Appendix D.

1. COMPUTERS & INTERNET (TECH.COMP): Remains unchanged from the original set. It holds more than 5000 newsgroup articles related to computer and Internet subjects. It remains unchanged from that of the original set.

2. ELECTRONICS (TECH.ELE): In the original structure, the *sci.electronics* newsgroup was added to the general category *Technology*. Here, we allow it to stand alone given the abundant number of articles it contains.

3. MEDICINE (TECH.MED): The *sci.med* newsgroup was also part of the general *Sci|Tech* category in the original set. In this modified structure, it stands alone.

4. SPACE (TECH.SPACE): Remains unchanged from the original set. The category is trained on the *sci.space* newsgroup and relevant DUC 2002 clusters.

5. NOT SCI|TECH: Remains unchanged from the original set.

Note that the current structure assumes that if a document is scientific, it definitely has to belong to one of the above-mentioned sub-topics. This might create a problem. The accuracy for the categories of the modified set are similar to, and as high as, those of the original set.

The performance of the modified set does not yield good results for the newly introduced categories. The articles manually attributed to category *Sci|Tech.Electronics* were related to cell-phones, however, this was not detected by the automatic classification. Furthermore, the performance for the computer-based category also remained

	Original Set				Modified Set					Binary Set	
	tech	t.comp	t.space	not.t	t.comp	t.ele	t.med	t.space	not.t	tech	not.t
accuracy	.969	.983	.979	.940	.947	.983	.996	.979	.905	.937	.937
precision	0.526	0	0.767	0.981	0.027	0	0.201	0.242	0.890	0.793	0.949
recall	0.833	0	1	0.954	0.029	0	0.956	0.939	0.978	0.418	0.985

Table 3.20: Differences in performance for *Sci|Tech*

unchanged as no new material was added in the training to allow the micro-classifier to detect that articles about the "Y2K" and "Microsoft on trial" are generally about computers. Only one of the "Y2K" articles was recognized with a probability of 0.88. The surprising change is noted for category *Sci|Tech.space*, whose precision drops from 0.767 to 0.242. Recall for this category, however, like that of *Sci|Tech.medicine*, remains above 90%. The performance of the complementary category is the most decent with 0.890 precision and 0.939 recall.

As we tried with *Events*, we study a simple binary classification for *Sci|Tech*, i.e., one with no multiple positive sub-categories. The accuracy drops to 93.75%. The precision for the positive set is higher than the highest one obtained for any of the original or modified set's positive categories. However, the performance suffers when it comes to recall, with only 0.418. On the other hand, precision and recall for the complement category is 0.949 and 0.985, respectively. The measures are better than those obtained for the modified set and similar to those of the original set. The differences in performance for category *Sci|Tech* is presented in Table 3.20.

3.3.6 Binary Structure

The key to accurate classification results lies in the quality of the training data. Sub-categories should be created depending on the granularity of the latter and on the intended task. The training data for categories *Arts*, *Business*, *Health*, *People*, and *Sports*, is not sufficient to create sub-categories. Doing so would hurt the performane of their respective micro-classifiers, probably more in terms of recall than precision. Before expanding *Arts*, for instance, its micro-classifier has to be able to recognize that "Charles Schultz" articles are about an artist.

We are able to sub-categorize natural disasters into general disasters, floods, storms, and earthquakes. The performance of the micro-classifier does not decrease when we move from the original to the binary set. Choosing whether or not to create

a sub-category also depends on what we are testing for. Therefore, if we want to distinguish between the disasters, we adopt the original structure. Else, we use the binary structure.

The choice for the EVENTS sub-categorization is between that which was provided for the DUC 2002 clusters (single and multiple events) or the simple binary-based structure. Their performance is similar; we record low recall for the positive set and low precision for the complementary category. For the time being, we are interested in determining if a document is about an event, rather than distinguishing between the types of events, especially that their definitions are still somewhat ambiguous. Furthermore, the precision and recall for the positive category of the binary set is better than those obtained for the original set.

Modifying category *Politics* was as penalizing as modifying category *Events*. Here, the distribution of the training data is the reason behind the poor performance of the micro-classifier. We have noticed that strict training results in high precision while lenient training mostly sacrifices both precision and recall. To illustrate this idea we refer to categories *Dis.Storm*, *Pol.world*, and *Pol.miscellaneous*.

This also better explained when taking the original *Sci|Tech* category as an example. The fact that it groups many types of scientific articles results in low values of precision. In the complement category, the same training hurts its recall. To the exception of category SCI|TECH.COMPUTERS, which results in null precision and recall values, the original structure has a decent overall performance. However, the precision of the positive category in the binary set is better than that of the original. Recall, on the other hand, is not.

The recommendation is to start off with two complementary concepts and sub-categorize the positive one if needed with appropriate and valid data. And unless the data really sets itself as a distinct cluster, one should not create a seperate category for it in the positive set. For example, we could add *Sci|Tech.Space* to the positive set of the binary *Sci|Tech* category and train it with the same documents that resulted in 0.767 precision and 1 recall. The data used for training is deleted from the positive set. However, parts of these articles might sometime contribute to recognize a scientific document, which is not necessarily about space.

It is also important to mention that accuracy is quite insensitive to variations. Comparing accuracy for different sub-categorizations is not a good evaluation method.

The accuracy for most of the modified sets is similar to that of the original set, while precision and recall were totally different, revealing a poor performance for the modified micro-classifiers.

Categories	$\|\delta\| \leq 0.1$	$\|\delta\| \leq 0.2$	$\|\delta\| \leq 0.3$	$\|\delta\| \leq 0.4$
Arts	0	0	0	0
Not Arts	0.989	0.990	0.990	0.991
Business	0.813	0.806	0.803	0.803
Not Business	0.952	0.952	0.954	0.954
Disaster	0.744	0.744	0.736	0.736
Not Disaster	0.977	0.977	0.977	0.977
Event	0.753	0.765	0.769	0.773
Not Event	0.360	0.366	0.373	0.371
Health	0.724	0.724	0.724	0.724
Not Health	0.984	0.984	0.984	0.984
People	0.912	0.912	0.912	0.915
Not People	0.718	0.718	0.716	0.717
Politics	0.637	0.701	0.703	0.709
Not Politics	0.835	0.835	0.834	0.833
Sci|Tech	0.547	0.557	0.557	0.557
Not Technology	0.966	0.966	0.966	0.965
Sports	0.702	0.702	0.702	0.702
Not Sports	0.987	0.987	0.987	0.987

Table 3.21: F_1 for orthogonal categories

For the original set of categories, we looked at the measures of precision and recall and attempted to determine the r-cut threshold. In what follows, we report the f_1 measure of the orthogonal based micro-classifiers and attempt to determine the s-cut threshold for the different categories. Table 3.21 shows the detailed f_1 measures for different values of δ and over all categories. The f_1 measures for categories *Arts*, *Business*, *Health*, *People*, *Politics*, and *Sports*, are computed from the precision and recall obtained in Table 3.13. Those of *Disasters*, *Events*, and *Technology*, use the precision and recall obtained for the binary structure in Tables 3.17, 3.18, and 3.20.

Table 3.22 summarizes the results; an f_1 measure above 90% is represented by a "⊤", one above 80% is represented by "**", and one above 70% by "*". The last column in the table shows the s-cut threshold, which is determined automatically. For the unchanged categories ARTS, BUSINESS, HEALTH, PEOPLE, and SPORTS, the s-cut is the same as the r-cut. For the modified category DISASTERS, the precision decreases when $|\delta|$ increases, recall remains fixed, and the s-cut for the binary structure is also

like the r-cut obtained for the original one. The s-cut for POLITICS drops to 0.4 instead of 0.3 and that of TECHNOLOGY also drops to 0.2 instead of 0.1. The majority of 0.4 s-cut thresholds shows that the f_1 measure performs better when $|\delta|$ takes on higher values.

cat.	**Positive**	**Negative**	**S-cut**
Arts		$\top$	0.4
Business	**	$\top$	0.1
Disasters	*	$\top$	0.1
Events	*		0.4
Health	*	$\top$	const.
People	$\top$	**	0.4
Politics		**	0.4
Sci|Tech		$\top$	0.2
Sports	*	$\top$	const.

Table 3.22: Categories with best f_1

We will rely on this binary structure in the next chapter to determine whether or not there are correlations between document categories and noun-phrase coreference chains. For this upcoming analysis, it is only important to know whether a document is about a topic or is not since we will not be concerned with more detail, such as particular disasters or sub-events. Furthermore, since the best s-cut for this structure is 0.4, we attribute documents to categories if their automatic classification $d_{\text{m}} \geq 60\%$.

Chapter 4

Categories and Coreference Chains

ERSS [Witte *et al.*, 2004] is the CLaC Laboratory's summarization system that has been used to participate in DUC 2003 and DUC 2004 [Bergler *et al.*, 2003, Witte *et al.*, 2004]. One of its major components is Fuzzy-ERS, a coreference resolution system that uses fuzzy logic to group NPs extracted from a document into coreference chains, "ordered sets of NPs that refer to the same entity" [Witte and Bergler, 2003]. These are crucial for building the summary of single documents; the initial[1] idea was to choose the longest NP from the longest chains until the length limit of the summary is reached.

These short summaries were also headed by the most salient classification of the document, which was obtained with a simple decision tree algorithm. An example summary would be:

> PEOPLE: construction project, Schultz's work, voices, a repository, his "Peanuts" strip

This chapter attempts to determine whether Fuzzy-ERS is category sensitive or insensitive. We are motivated by the idea that potential category/np-chains patterns allow for the implementation of new ideas that would enhance the summaries. In order to look for possible correlations between a document's classification results and its noun-phrase coreference chains, we must (1) define how and with which frequency single[2] category labels were assigned to the documents, (2) gather the total number of

[1]This simple strategy was applied for very short summaries such as those required for Task 1 in DUC 2003. Other strategies were implemented later.

[2]For summarization purposes, we are interested in multi-label categorization. Here, however, we are mainly concerned with single labels to draw conclusions particular to the category.

chains per document and the frequency with which chains of different lengths occur, and (3) check for the existence of correlations between both categories and their associated np-chains.

4.1 Category Labels

In the previous chapter, we came to the conclusion that 0.4 was the best s-cut threshold for the binary set. Therefore, in the following analysis, document d is labeled with category c if the probability that d belongs to c is greater than or equal to 60%. Table 4.1 shows the number of documents (from the 624 available testing docs) that have been assigned to the different categories. It also reports the minimum and maximum probabilities for documents belonging to the category. Note that for all categories, we have documents that have taken a maximum probability value of 1, i.e., $d_{\text{m}} = 1$. None of the documents in the cluster was labeled with ARTS and therefore we note the null entry. On the other hand, PEOPLE appears in 74.51% of the documents with a starting probability as low as 65%. Another recurring category seems to be EVENTS. It has been assigned to more than half of the documents in the corpus. However, based on previous observations, these categories are not the ones with the best performance; SCI|TECH has a high starting probability of 85%, and categories HEALTH and SPORTS both have a minimum probability of 1.00.

	Documents		**Probability**	
cat.	number	percentage	min.	max.
Arts	0	0	–	–
Business	126	20.19%	0.74	1.00
Disasters	42	6.73%	0.70	1.00
Events	366	58.65%	0.63	1.00
Health	26	4.16%	1.00	1.00
People	456	74.51%	0.65	1.00
Politics	174	27.88%	0.62	1.00
Sci|Tech	31	4.96%	0.85	1.00
Sports	19	3.04%	1.00	1.00

Table 4.1: Category assignment and probabilities

Furthermore, category EVENTS distributes more evenly over the different probability instances than any other category. POLITICS follows closely behind, but like PEOPLE, it has gaps between the instances that appear in the [20–80]% range, and

clusters most of its documents either below 20% or above 80%. On the other hand, like DISASTERS, SCI|TECH, and SPORTS, BUSINESS clusters most of its non-zero probability documents above 85%, very few appear below, and the rest fall in the complementary category.

4.2 Coreference Chains Statistics

We present some basic notions about coreference chains: (1) how is the length of a coreference chain measured, (2) how frequent are coreference chains in a document or a set of documents, and (3) how do length and frequency of chains correlate. Then, we examine the occurrence of singleton and non-singleton chains across documents, clusters, and categories.

4.2.1 Length and Frequency

Coreference chains within a document vary in length and frequency. For example, document XIE19980217.0051, which appears in Table 4.2 has a total of 28 chains (including singletons): 19 chains of length 1, 6 chains of length 2, 2 chains of length 3, and 1 chain of length 6. When we say length of the chain we mean the number of NPs that it consists of. Note that NPs are attributed to a chain depending on a certain fuzzy setting. For instance, a fuzzy setting $\gamma = 0.6$ is more lenient than a $\gamma = 1$ setting. As more NPs are associated together, the length of a chain increases, and the frequency of its occurrence becomes rarer. Figure 4.1 illustrates this idea. Note that the lengthiest chains, above 70 NPs, only appear in documents of type *People* and *Politics*.

Document	*totalchains*	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$
XIE19980217.0051	28	19	6	2	–	–	1

Table 4.2: Example of chains per document

Table 4.3 shows the frequency of chains with different lengths as they occur across the set of categories. We exclude *Arts* as none of the documents in the testing set were labeled as such. The first column displays the different *lengths* chains can have; there are no chains of lengths 34, 35, 42, 44, 48, 49, 53–71, 73, 75, and 77–95. There are also no chains lengthier than 96 NPs.

	BUS		DIS		Events		Health		People		POL		Sci|Tech		Sports	
L	SF	DF	SF	DF	SF	DF	SF	DF	SF	DF	SF	DF	SF	DF	SF	DF
1	100	61.4	100	49.47	100	64.4	100	65.8	100	65.8	100	66.4	100	64.0	100	45.47
2	100	11.1	100	16.3	99.7	11.4	100	11.5	99.7	11.5	100	11.9	100	11.9	100	18.5
3	97.6	4.58	97.6	6.70	94.2	4.64	100	5.76	94.1	4.82	91.4	4.86	96.7	4.83	100	8.9
4	82.5	2.75	100	3.33	78.1	2.09	80.7	2.66	78.5	2.74	78.1	2.80	70.9	2.59	100	3.9
5	69.8	1.84	85.7	2.13	66.3	1.82	69.2	1.83	64.6	1.89	76.4	1.51	64.5	1.7	89.4	3
6	49.2	1.61	69.1	1.82	45.3	1.64	50	1.69	45.6	1.59	51.7	1.64	51.6	1.81	73.6	2.14
7	33.3	1.38	50	1.57	33.8	1.35	38.4	1.6	34.2	1.39	37.9	1.43	35.4	1.36	47.3	1.55
8	25.3	1.43	45.2	1.63	22.6	1.44	23.1	1.5	25	1.38	25.8	1.44	22.5	1.42	57.8	1.72
9	20.6	1.10	40.4	1.11	20.4	1.25	23.1	1.33	21.4	1.29	21.8	1.26	16.1	1.2	47.3	1.22
10	11.9	1.06	19.1	1	9.83	1.16	15.3	1.25	10.7	1.12	8.04	1.14	16.1	1	26.31	1
11	8.73	1.27	16.6	1.28	9.28	1.14	11.5	1.33	9.86	1.15	10.3	1.16	6.45	1	31.5	1
12	11.1	1	26.1	1	7.92	1.10	11.5	2	8.55	1.07	8.04	1	3.22	1	31.5	1
13	9.52	1.08	23.8	1.1	6.83	1.12	11.5	1.33	7.67	1.11	9.19	1.06	3.22	1	26.3	1.2
14	4.76	1	14.2	1	6.28	1	3.84	1	6.14	1.03	4.59	1.12	3.22	1	15.7	1
15	7.14	1	14.2	1	5.73	1.09	3.84	1	6.14	1.07	4.02	1.14	6.45	1	26.3	1
16	5.55	1	7.14	1	5.19	1	-	-	5.04	1	3.44	1	3.22	1	5.26	1
17	3.17	1	7.14	1	5.46	1.05	3.84	1	5.04	1.04	4.02	1	3.22	1	5.26	1
18	4.76	1.16	4.76	1.5	.81	1.33	-	-	1.09	1.2	.57	2	3.22	1	10.5	1.5
19	3.96	1	7.14	1	.03	1	-	-	3.5	1	2.29	1	6.45	1	5.26	1
20	3.96	1	9.52	1	.01	1	3.84	1	2.19	1	2.87	1	-	-	21.1	1
21	1.58	1	4.76	1	.27	1	-	-	.87	1	.57	1	-	-	10.5	1
22	-	-	-	-	.54	1	-	-	.87	1	.57	1	-	-	-	-
23	.79	1	2.38	1	.81	1	-	-	1.09	1	1.72	1	-	-	5.26	1
24	-	-	-	-	1.69	1	-	-	1.09	1	1.72	1	3.22	1	-	-
25	1.58	1	2.38	1	1.09	1	-	-	1.31	1	-	-	-	-	5.26	1
26	.79	1	-	-	.27	1	-	-	.43	1	1.14	1	-	-	-	-
27	-	-	-	-	1.69	1	-	-	1.31	1	.57	1	-	-	-	-
28	-	-	-	-	-	-	-	-	.21	1	.57	1	-	-	-	-
29	-	-	-	-	.27	1	-	-	.21	1	-	-	-	-	-	-
30	-	-	-	-	.54	1	-	-	.43	1	-	-	3.22	1	-	-
31	-	-	-	-	.27	1	-	-	.43	1	.57	1	-	-	-	-
32	-	-	-	-	.27	1	-	-	.65	1	1.72	1	-	-	-	-
36	.79	1	2.38	1	.54	1	-	-	.43	1	-	-	-	-	5.26	1
37	-	-	-	-	-	-	-	-	.43	1	.57	1	-	-	-	-
38	-	-	-	-	-	-	-	-	.21	1	-	-	-	-	-	-
39	.79	1	2.38	1	-	-	-	-	.21	1	-	-	-	-	5.26	1
40	.79	1	2.38	1	-	-	-	-	.21	1	-	-	-	-	5.26	1
41	-	-	-	-	.54	1	-	-	.43	1	1.49	1	-	-	-	-
43	-	-	-	-	-	-	-	-	.21	1	.57	1	-	-	-	-
45	-	-	-	-	-	-	-	-	.21	1	-	-	-	-	-	-
46	-	-	-	-	.27	1	3.84	1	.21	1	-	-	-	-	-	-
47	-	-	-	-	-	-	-	-	.21	1	.57	1	-	-	-	-
50	.79	1	2.38	1	.27	1	-	-	.21	1	.57	1	-	-	5.26	1
51	-	-	-	-	.27	1	-	-	.65	1	.57	1	-	-	-	-
52	-	-	-	-	.27	1	-	-	-	-	-	-	-	-	-	-
72	-	-	-	-	-	-	-	-	.21	1	-	-	-	-	-	-
74	-	-	-	-	-	-	-	-	.21	1	.57	1	-	-	-	-
76	-	-	-	-	-	-	-	-	.21	1	.57	1	-	-	-	-
96	-	-	-	-	-	-	-	-	.21	1	.57	1	-	-	-	-

Table 4.3: Set and document frequency for chains of different length across categories

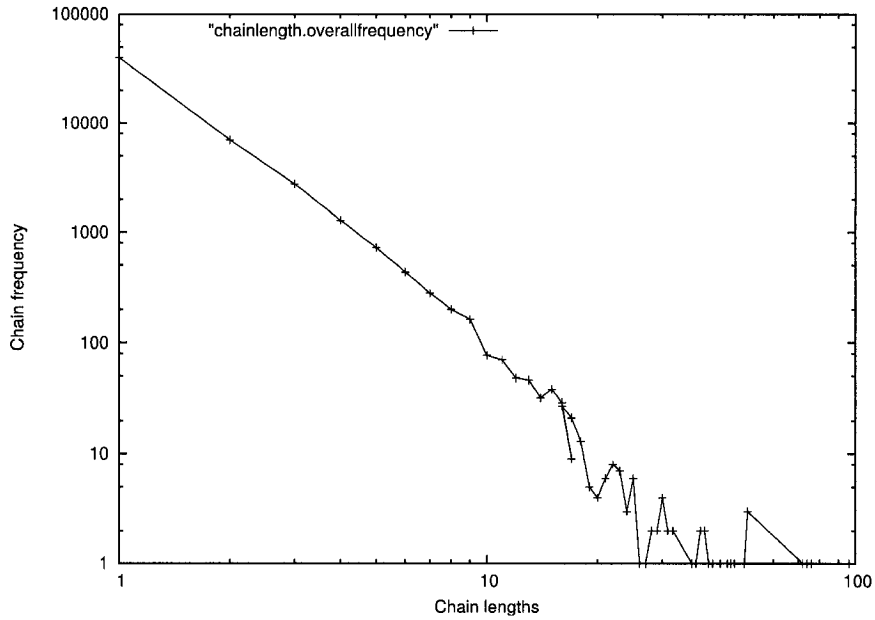

Figure 4.1: Frequency of chains of different lengths

The second column is Set Frequency (SF) which shows the percentage of documents in the set of documents that are assigned to the category (with probability above 60%) and that have chains of $length(x)$. The categories that spread out across the highest number of different chain lengths are: PEOPLE, POLITICS, and EVENTS. Category BUSINESS is at the limit between the first three and SPORTS, DISASTERS, and SCIENCE, which spread out less but with more significant numbers, i.e., 5.26% of SPORTS documents have chains of length 36 while only 0.43% of PEOPLE documents have chains of the same length. HEALTH comes in last in terms of uniform spreading. Set frequency is inversely proportional to chain length. For all categories, the gap between $SF(x)$ and $SF(x+1)$ decreases as we move from length x to $x+1$.

The third column is Document Frequency (DF) and it measures the average number of chains of $length(x)$ that appear in one of the set documents. For example, for category BUSINESS, when chains of length 1 occur, they do so with a frequency of 61.4 chains per document. Similarly, when chains of length 2 occur, they do so with a frequency of 11.1 chains per document. This also means that the sum of the instances in the column is not equal to the average number of chains per document for the category. Like set frequency, document frequency is also inversely proportional to chain length.

Furthermore, it is interesting to see that documents about DISASTERS or SPORTS have the lowest number of chains, 66.47 and 60.58 respectively, while documents belonging to any other category have an average number of 83–89 chains per document. Also, this high number mostly accounts for singleton chains; 49.47 for DISASTERS and 45.47 for SPORTS. Others categories maintain a number of singleton chains that varies between 61 and 66, per document. The natural question that we raise is whether or not the number of chains per document and across the different categories is linked to its length. Table 4.4 represents the average number of chains (singleton and non-singleton) and word-tokens per document, across different categories. In Section 4.2.2, we will see that singleton chains account, across categories, for 76% of the total number of chains in a document. Therefore, plotting word-tokens against the total number of chains, singleton chains, or non-singleton chains will yield the same correlation. We can roughly say that a 100 word-tokens lengthier document results in about 20 more chains (15 singletons and 5 non-singletons), while 35 word-tokens (558.97–592.94) do not make a significant difference, i.e., at most 3.46 more or even less chains, since the lengthiest document does not necessarily have, on average, the maximum number of chains (e.g., SCI|TECH). The lengthiest chains (above 70 NPs) appear in the 1st, 2nd, and 14th lengthiest documents. Their frequency is shown in Table 4.5. In general, the lengthier the document, the more chains we gather, which explains the low number of chains for categories DISASTERS and SPORTS.

	BUS	**DIS**	**Events**	**Health**	**People**	**POL**	**Sci|Tech**	**Sports**
no. of chains	83.45	66.47	86.35	89.61	88.01	88.99	86.15	60.58
Word-tokens	573.60	422.35	558.97	561.5	578.06	586.35	592.93	457.789

Table 4.4: Total number of chains and word-tokens per document across categories

Rank	Document	Word-tokens	$l=1$	...	$l=72$	$l=74$	$l=76$	$l=96$
1	NYT20000304.0128	3309			–	–	–	1
2	NYT19981005.0386	2170			–	1	1	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
14	NYT19980605.006	1672			1	–	–	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 4.5: Frequency of the lengthiest chains appearing in the lengthiest documents

4.2.2 Singleton Chains

Singleton chains consist of only one NP. Their frequency in a document represents the number of NPs that were not associated with any other. Figure 4.2 shows a consistent correlation between the total number of chains in a document and the singleton chains. Most chains in a document are singleton chains. Researchers usually omit singleton chains when running multiple statistics over coreference chains. However, we will take a brief look at their particularities across documents, topic clusters, and categories, before studying those of non-singleton chains, i.e., chains with length strictly greater than 1.

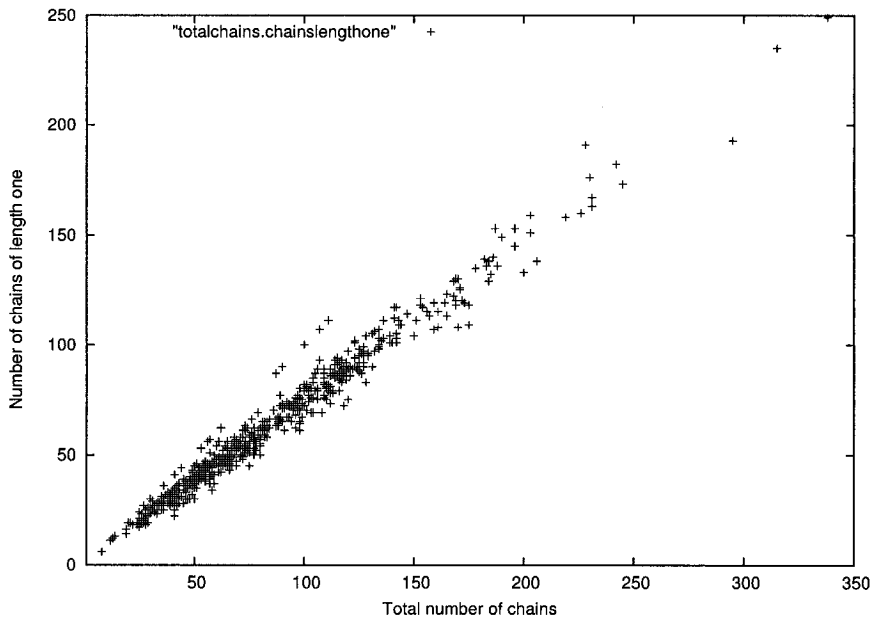

Figure 4.2: Singleton chains vs. total chains overall documents

1. Documents

Figure 4.3 plots the percentage of singleton chains over all documents:

1. At least 54% of the chains in a document are singletons. The minimum has been recorded for document NYT19981021.0066, which belongs to the "Microsoft on trial" BEP cluster.[3]

[3]BEP reads Business, Events, and People. This is the result of the multi-label categorization for the cluster. Each initial represents a category, i.e., A for *Arts*, B for *Business*, D for *disasters*, E for

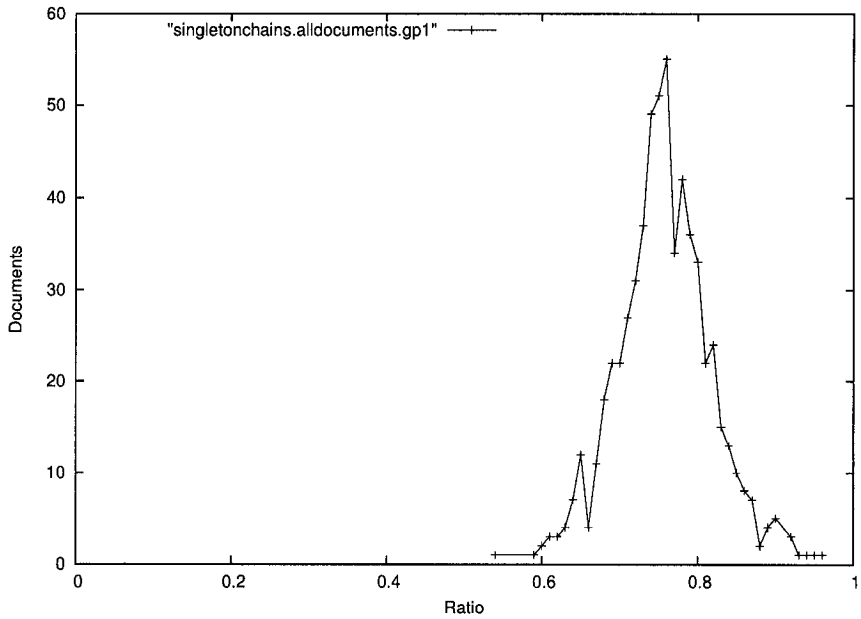

Figure 4.3: Percentage of singleton chains over all documents

2. At most 96% of the chains in a document are singletons. The maximum has been recorded for document APW20000426.0203, which belongs to the "Rosebowl" EPS cluster.

3. On average 76% of the chains in a document are singletons.

The [70–79]% range spans across the largest number of news articles. There are less singletons that make up [60–69]% of the total number of chains per document than those that fall in the [80-89]% range. Figure 4.6 shows the exact percentage of documents distributed over these different ranges:

Range	50–59	60–69	70–79	80–89	90–99	100
Documents	0.3%	13.94%	61.69%	22.11%	1.92%	0%

Table 4.6: Document distribution over different ratio ranges

2. Clusters

We sort the documents by the original topic clusters they were assigned to and compute the average percentage of singleton chains per document for every cluster. The

events, H for *health*, P for peope, P for *Politics*, T for *science*, and S for *sports*. If the multi-label contains only one P, it is always *People*. PP stand for *People* and *Politics*.

minimum percentage of singleton chains in a document and per cluster is 69%. The maximum is 83%. Few clusters have less than 72% or more than 78% singleton chains in a document.

Table 4.7 lists some clusters and the average percentage of singleton chains in their respective documents. We notice that there are no specific correlations between the topic of a cluster and the average percentage of singleton chains per document. For example, category BEP occurs at 70%, 75%, and 81%, DE occurs at 71% and 76%, and PP occurs at 72%, 76%, and 79%.

Percentage	Cluster
69%	BEPP: Census;
70%	BEP: Microsoft on trial;
71%	DE: Flood disaster;
72%	PP: Democrats vs. Clinton; EP: Bombings;
73%	BE: ATM fees; EHPT: Scizophrenia;
74%	PP: Capital Punishment; EHT: E-coli;
75%	BEP: Online Auctions; EPS: Rosebowl;
76%	DE: Cold deaths; PP: Abortion doctors killed;
77%	ET: Hubble; EPP: Turkish and Syrian tensions;
78%	EPS: Asian games; BET: Three gorges dam in China;
79%	PP: Croatia; EP: Kidnaps in Chechnya;
80%	P: Charles Schultz; EP: Canada's northern breakup;
81%	BEP: Y2K;
82%	EP: Palestinians flights;
83%	E: Homeless people;

Table 4.7: Average percentage of singleton chains per document, sorted by DUC-cluster

3. Categories

	BUS	DIS	Events	Health	People	POL	Sci|T	Sports
min.	0.54	0.59	0.54	0.65	0.54	0.61	0.60	0.62
max.	0.90	0.93	0.96	0.80	0.96	0.94	0.86	0.96
avg.	0.74	0.75	0.76	0.74	0.76	0.77	0.76	0.76

Table 4.8: Percentage of singleton chains per category

In order to make sure that there are no correlations between categories and percentage of singleton NPs, we display the distribution of chains with *length* = 1 across the set of categories in Table 4.8. For example, 60–86% of chains in SCI|TECH documents are singletons. Category HEALTH has a slightly narrower range (65–80%) of

News Source	Documents		Singleton Chains		Non-Singleton Chains				
	number	perc.	total	percentage	total	perc.	min.	max.	avg.
APW	266	42.62%	14199	76.25%	4421	23.75%	1	52	16.6
NYT	252	40.38%	21942	73.72%	7866	26.28%	3	102	31.21
XIE	106	16.98%	3704	76.27%	1152	23.73%	3	34	10.86

Table 4.9: Source characteristics

chains in these type of documents are singletons. The fact that its minimum number of singleton chains is the highest among all categories means that it will spread out over chains of different length least of all. This revisits the earlier Document Frequency concept from Table 4.3. We record for categories BUSINESS, EVENTS, and PEOPLE, the lowest percentage (54%) occurence for chains of $length = 1$, per document. The highest is 96% for EVENTS, PEOPLE, and SPORTS, closely followed by POLITICS, DISASTERS, and finally BUSINESS.

Encountering a low percentage of singleton chains per document for a category means the subject of the document is somewhat fixed and more NPs tend to cluster together. This makes sense if we take, for example, PEOPLE, DISASTERS, and even EVENTS. However, we record for these same categories a very high percentage of singleton chains, which refutes the idea that depending on the category, we might have more or less singleton chains. Indeed, the percentage of singleton chains in a document depends on the constituents of the document itself.

4.2.3 Non-Singleton Chains

Singleton chains are omitted from the analysis of coreference chains, since a chain of length $l = 1$ is not exactly considered a coreference chain. Therefore, in the following section, we look at the particularities of non-singleton chains, across documents, topic clusters, and categories.

1. Documents

The data on which we ran the Fuzzy-Coreferencer parallel to our micro-classifiers consists of three types of news sources: the ASSOCIATED PRESS NEWSWIRE (APW), the NEW YORK YIMES (NYT), and the CHINESE XINUHA NEWSWIRE (XIE). Table 4.9 shows the distribution of the 624 testing documents over these news sources as well as the average percentage of singleton and non-singleton chains per document. Those

with the lowest number of chains (one chain per document) are ASSOCIATED PRESS news articles describing disasters (e.g., "Sweden fire" and "Philippine typhoons"), sports (e.g., "Rosebowl"), and politics (e.g., "Turkish politics" and "Bosnian war crimes"). The document with the highest number of chains (102 chains per document) is a NEW YORK TIMES article on "U.S presidential elections." Overall, the CHINESE XINHUA has the least number of chains per document (10.86 chains per doc). This is possibly due to the shorter nature of the articles. On average, there are 21.53 chains per document.

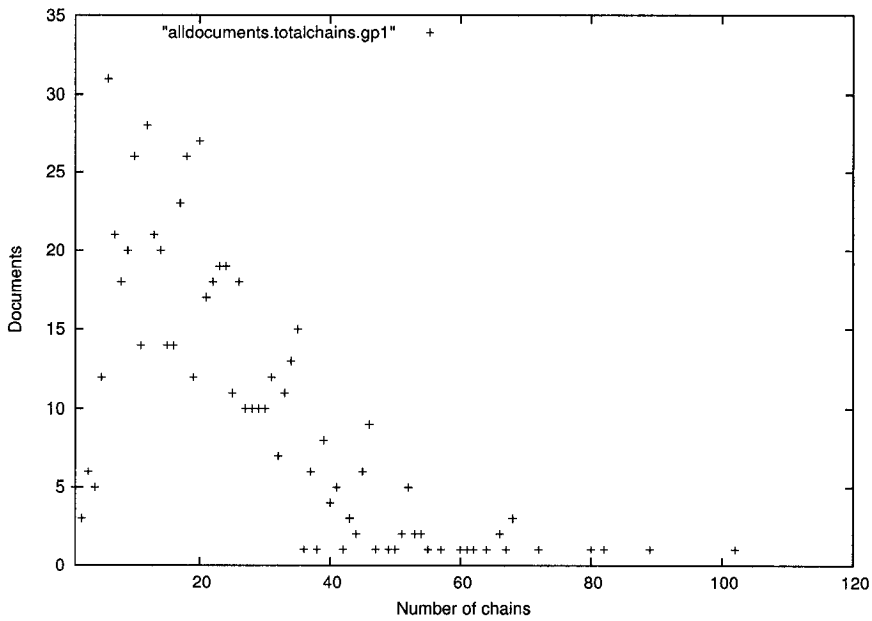

Figure 4.4: Number of non-singleton chains per document

Range	0–9	10–19	20–29	30–39	40–49	50–59	60–69	70–79	80–89	90–99	100–109
Doc. (%)	19.39	31.73	25.64	13.46	5.12	2.24	1.60	0.16	0.48	0	0.16

Table 4.10: Percentage of documents in different chains per document ranges

Figure 4.4 plots the total number of non-singleton chains versus the number of documents in which these appear. The percentage of documents decreases as the range increases; it indeed drops (from 31.73% to 25.64%) by a factor of 6.09% when we move from the [10–19] range to the next and similarly (from 25.64% to 13.46%) by a factor of 12.18% when we move from the [20–29] range to the next. Table 4.10 shows the percentage of documents distributed over the different ranges.

2. Clusters

range	≤ 9	10–14	15–19	20–24	25–29	30–34	≥ 35
no. clusters	2	8	15	9	10	8	2

Table 4.11: Number of clusters at different ranges of average number of non-singleton chains per document

The documents in the DUC 2003 corpus were originally provided in 60 different topic clusters. In order to calculate the average number of chains per document for each cluster, we sum the total number of chains over all documents in the cluster and divide by the cluster size, i.e., the number of documents in the cluster. Although the size of a cluster varies from 8 to 14 documents, most clusters contain 10 or 11 documents. If we detect more chains in a lengthier cluster, then we maintain a similar average. If we don't, the average number of chains per document in a cluster of size x will be less than the average in a cluster of size $x - 1$ and vice versa.

Table 4.2.3 shows the number of clusters at different ranges of average number of non-singleton chains per document. As we can see, documents roughly fall in the [10–30] chains per document range. The cluster with the lowest number (8.1) of chains per document is the BET cluster: "Three gorges dam in China." That with the highest number (44.3) is a PP cluster: "Democrats vs. Clinton." Table 4.12 lists examples of topic-clusters at different numbers of chains per document. Like our previous observation on singleton chains and topic-clusters, we find that there is no particular pattern between non-singleton chains and topic-clusters. These can have an average number of chains of different length.

3. Categories

Here, we sort documents by single-label categories and attempt to verify whether or not there are any correlations between the category and the document's coreference chains. Figures 4.5 and Table 4.13 show the distribution of the total number of chains per document across the categories.

The total numbers of chains for categories *Disasters* and *Sports* cluster between 1 and 34 chains per document. There is only one exception for *Disasters* where 72 chains were encountered for a document. The next wider range in which documents are clustered is that of *Business*. Chains vary between 3 and 64, most are just below

no. of hains	Clusters
8	BET: Three gorges dam in China;
9	DE: Cold deaths;
10	EPP: Tribal tensions in Africa; EPS: Asian games;
11	PP: Monica and Bill; BEP: Y2K;
12	EPP: Turkish and Syrian tensions;
14	BE: Vietnam's cashew; PP: Turkish politics;
15	DE: Flood disaster; BEP: Realizing the Euro; EP: Kidnaps in Chechnya;
16	EHT: E-coli; EP: Iran elections;
17	BE: Swissair crash;
18	DE: Philippine typhoon; EPS: Rosebowl; EP: Bosnian war-crimes;
19	BE: ATM fees; EP: Salt-Lake City bribe;
21	P: Charles Schultz; PP: Pope about Croatia;
22	ET: Hubble; EPP: Explosion in Israel; BEP: Brazil elections;
24	PP: Pinochet;
25	BE: Banning cell-phones in cars; EPP: Terrorist attacks on U.S. embassies;
26	EP: Doctors without borders;
27	PP: Branch Davidian siege;
28	DE: Caribbean storms; BEP: Online auctions;
29	EPT: Neutrinos;
30	EHPT: Schizophrenia; ET: Meteor showers; BEP: Global economy;
32	BEPP: Census;
33	EP: Teaching evolution in schools;
34	E: Homeless people;
41	EP: Bombings;
44	PP: Democrats vs. Clinton;

Table 4.12: Truncated number of non-singleton chains per document for multi-label clusters

40. Categories *Sci|T* and *Health* somewhat have the same repartition. Documents belonging to *Politics*, *Events*, and *People*, have their total number of chains clustering tightly in that order until they approximately reach 80 chains per document. We record for *People* and *Politics* some faint activity above 100 chains per document.

	Business	DIS	Events	Health	People	POL	Sci|T	Sports
min.	3	1	1	7	1	1	4	1
max.	72	64	82	68	102	102	67	34
avg.	22.01	17	21.91	23.73	22.25	22.58	22.12	15.10

Table 4.13: Min., max., and avg. number of non-singleton chains per document across categories

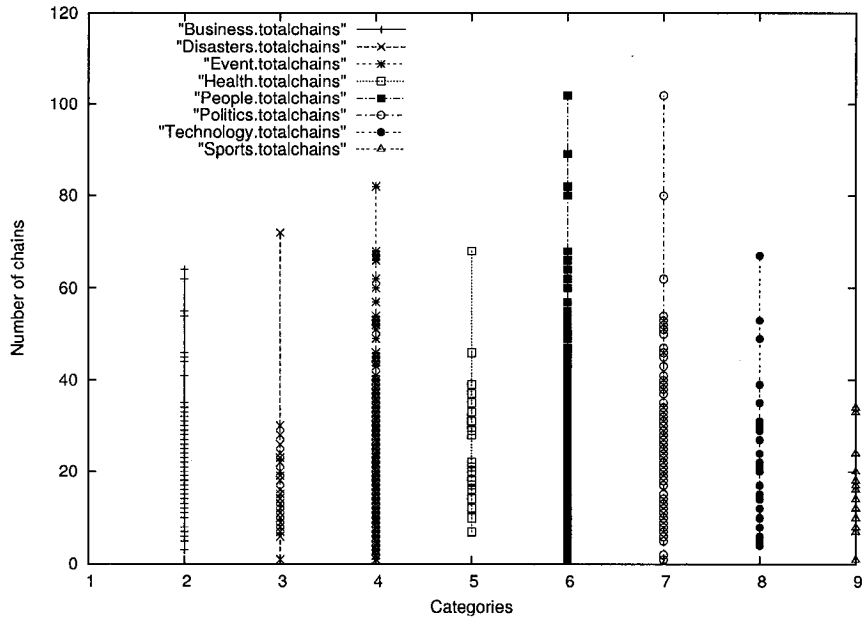

Figure 4.5: Total number of non-singleton chains per document across categories

4.3 Chains and Multi-label Category Patterns

In multi-label text classification, a document can be assigned to more than one category. By collecting the different labels for every document, we are closer to defining its content. We have 34 distinct category combinations of the 2^{10} possible ones. We present these as well as cluster categories in the two following sections.

1. Documents

Table 4.14 lists the distinct 34 multi-labels which were assigned to the set of 624 DUC 2003 documents, as well as their respective min., max., and avg. number of chains per document. The gap between the min. and max. number of chains, per pattern, ranges from 0 for multi-label BEPT to 101, for multi-label PP. Table 4.15 shows the gaps for each of these multi-labels. The higher the gap, the less reliable the correlation. The average number of chains for these patterns vary from as low as 1 chain per document, for multi-labels D, EPS, P, and PP, to 102 chains per document, for multi-label PP. We notice that the highest averages are all a function of categories EP. The other observation is that the percentage of singleton chains across patterns cluster around the 70s with two instances in the 80s, for multi-labels BPT and T, and

Doc-label	Doc.		Clusters		Non-Singleton Chains (L)			Singleton Chains (%)		
	number	perc.	number	perc.	min	max.	avg.	min.	max.	avg.
B	21	3.36%	9	15%	5	35	22.28	0.66	0.81	0.73
BDT	1	0.16%	1	1.66%	6	6	6	0.76	0.76	0.76
BE	9	1.44%	5	8.33%	6	31	14.44	0.69	0.82	0.77
BEH	2	0.32%	1	1.66%	16	17	16.5	0.69	0.77	0.73
BEHP	1	0.16%	1	1.66%	22	22	22	0.7	0.7	0.7
BEHPT	1	0.16%	1	1.66%	29	29	29	0.74	0.74	0.74
BEHT	1	0.16%	1	1.66%	31	31	31	0.74	0.74	0.74
BEP	44	7.05%	13	21.66%	3	54	23.22	0.54	0.86	0.74
BEPP	3	0.48%	1	1.66%	46	62	51.33	0.61	0.64	0.62
BEPT	2	0.32%	1	1.66%	20	20	20	0.75	0.76	0.76
BET	2	0.32%	2	3.33%	6	15	10.5	0.65	0.81	0.73
BH	1	0.16%	1	1.66%	7	7	7	0.76	0.76	0.76
BP	33	5.28%	11	18.33%	6	64	22.63	0.65	0.83	0.74
BPP	2	0.32%	1	1.66%	20	25	22.5	0.70	0.73	0.72
BPT	2	0.32%	1	1.66%	5	6	5.5	0.79	0.86	0.83
BT	1	0.16%	1	1.66%	8	8	8	0.75	0.75	0.75
D	41	6.57%	7	11.66%	1	72	18.26	0.59	0.93	0.75
DH	1	0.16%	1	1.66%	20	20	20	0.65	0.65	0.65
E	18	2.88%	13	21.66%	3	40	19.6	0.63	0.9	0.77
EH	2	0.32%	2	3.33%	7	35	21	0.71	0.78	0.74
EHP	9	1.44%	1	1.66%	12	68	27.7	0.65	0.79	0.73
EHPT	1	0.16%	1	1.66%	14	14	14	0.76	0.76	0.76
EP	117	18.75%	30	50%	3	68	22.41	0.63	0.87	0.76
EPP	116	18.58%	25	41.66%	5	54	20.92	0.65	0.85	0.76
EPS	17	2.72%	2	3.33%	1	34	15.17	0.69	0.96	0.77
EPT	7	1.12%	2	3.33%	17	49	28.57	0.69	0.83	0.73
ES	2	0.32%	1	1.66%	12	17	14.5	0.63	0.76	0.69
ET	12	1.92%	3	5%	4	67	26.33	0.60	0.86	0.76
H	5	0.80%	2	3.33%	10	31	16.2	0.71	0.80	0.75
HP	1	0.16%	1	1.66%	28	28	28	0.75	0.75	0.75
P	51	8.17%	21	35%	1	89	19.64	0.65	0.95	0.79
PP	51	8.17%	17	28.33%	1	102	25.52	0.64	0.94	0.71
T	1	0.16%	1	1.66%	10	10	10	0.84	0.84	0.84
NONE	46	7.37%	11	18.33%	4	52	20.30	0.61	0.87	0.76

Table 4.14: Document multi-labels and coreference-chains

three in the 60s, for multi-labels BEPP, DH, and ES.

2. Clusters

A multi-label for the cluster is obtained by grouping together all distinct categories to which the cluster's documents are assigned. There are 16 multi-label patterns for the clusters, three of which do not occur in the previous document multi-label patterns. These are EHPT, DE, and EHT. For example, in a cluster, some documents might be assigned to *Events* and others to *Disasters*, and while no document has a DE tag, the

Gap	Document multi-labels
0–9	B; BEH; BEPT; BET; BPP; ES;
10–19	BEPP;
20–29	BE; EH; H;
30–39	E; EPS; EPT;
40–49	EPP; none;
50–59	BEP; BP; EHP;
60–69	EP; ET;
70–79	D;
80–89	P;
90–99	–
≥100	PP

Table 4.15: Gaps between min. and max. no. of chains for different document multi-labels

Cluster-label	Clusters		Non-Singleton Chains (L)			Singleton Chains (%)		
	number	perc.	min	max.	avg.	min.	max.	avg.
BEPP	1	1.66%	32.4	32.4	32.4	0.69	0.69	0.69
EHPT	1	1.66%	30.25	30.25	30.25	0.73	0.73	0.73
BE	4	6.66%	14	25.7	19.35	0.73	0.75	0.74
BP	2	3.33%	21.2	30.6	25.9	0.74	0.74	0.74
EHT	1	1.66%	16	16	16	0.74	0.74	0.74
EPT	1	1.66%	29.1	29.1	29.1	0.74	0.74	0.74
BEP	6	10%	11.7	30.4	23.05	0.70	0.81	0.75
DE	4	6.66%	9.6	28	17.65	0.71	0.77	0.75
EPP	7	111.66%	0.6	25.9	16.85	0.73	0.8	0.76
EPS	2	3.33%	10.6	18.2	14.4	0.75	0.78	0.76
PP	10	16.66%	11.1	44.3	24.48	0.72	0.79	0.76
EP	12	20%	10.5	41.5	21.94	0.72	0.82	0.77
ET	2	3.33%	22.2	30.3	26.25	0.77	0.77	0.77
P	2	3.33%	21.2	24.2	22.7	0.74	0.8	0.77
BET	1	1.66%	8.1	8.1	8.1	0.78	0.78	0.78
E	4	6.66%	9.6	34.1	21.67	0.75	0.83	0.78

Table 4.16: Cluster multi-labels and coreference-chains

cluster does. This is not computed on the appended documents, as we did when we tested for length sensitivity in Section 3.3.4. Here, the multi-label for a cluster is the sum of labels for individual documents in the cluster. Table 4.16 lists the different categories that can be found across any of a given cluster's document.

Here, the categories that cluster the most frequently together are EP, PP, EPP, and BEP, and regardless the average number of chains per document, per multi-label cluster, singleton chains still represent at least 69% of the total number of chains and at most 78%. Also, clusters with a large average number of chains, have *People* or *Events* labeled documents. For example, BEPP (32.4), EHPT (30.25), and EPT (29.1).

Table 4.17 lists the different cluster multi-labels and gives some topic examples.

Cluster multi-label	Topics
BEPP	d115f: Census;
EHPT	d100a: Schizophrenia;
BE	d110d: ATM fees; d111d: Banning cell-phones use in cars; d122h: Vietnam's cashew; d30016t: Swissair crash;
BP	d31031t: Congress resolves spending; d31041t: Philippine's airline;
EHT	d109d: E-coli;
EPT	d125i: Neutrinos;
BEP	d103b: Online auctions; d127j: Realizing the Euro; d129i: Y2K; d30025t: Brazil elections; d30048t: Global economy; d31033t: Microsoft on trial;
DE	d101a: Flood disaster; d31010t: Cold deaths; d31011t: Philippine's typhoon; d31028t: Caribbean storms;
EPP	d118g: African tribal tensions; d30005t: Terrorist attacks on U.S. embassies; d30028t: Turkish & Syrian tensions;
EPS	d106c: Rosebowl; d30020t: Asian games;
PP	d107c: Capital punishment; d113e: Monica L. & Bill Clinton; d30003t: Pinochet; d31009t: Turkish politics;
EP	d105b: Salt-Lake city bribe; d121h: Egyptian Interpol; d30040t: Palestinian flights; d30042t: Libyan problems;
ET	d112e: Hubble; d30012t: Meteor shower;
P	d102a: Charles Schultz; d108c: U.S. presidential elections;
BET	d120g: Three gorges dam in China;
E	d117f: Getting rid of homeless people; d119g: Bangladesh & India talks; d128j: Storms & floods in Mozambique; d31022t: Sweden fire;

Table 4.17: Cluster multi-labels and topics

Chapter 5

Conclusion

Text categorization is the task of assigning a document to a category that reflects its content. It revealed to be beneficial in the context of automatic document summarization. Appending the document's label(s) to the ERSS-based summary, as we did for Task 1 in DUC 2003, resulted in good measures of usefulness, while those of coverage were below average [Over, 2003].

After surveying different news sources and their categorization schemas, we decided to adopt nine general categories into which we could classify the documents: 1) arts, 2) business, 3) health, 4) politics, 5) science, 6) sports, 7) disasters, 8) events, and 9) people. We were inspired to create the first six given the news sources' sections. The last three were DUC 2002 based categories.

Instead of combining all concepts into one comprehensive structure, and building a common classifier for all, we decided to create nine disjoint micro-classifiers, each built for a separate concept and its complement. In other words, in order to achieve a multi-label and multi-dimensional classification of an article, the latter was sequentially processed over these nine micro-classifiers. For each, the article was either attributed to the positive or the negative set of examples, i.e., it either belonged to the category or to its complent.

The best approach for these independent binary micro-classifiers was a statistical Naive Bayes based one. Training corpora includes the DUC 2002 corpus, the 20 NEWSGROUPS corpus, a collection of 290 WALL STREET JOURNAL articles, and some web-collected articles. The micro-classifiers were tested on the DUC 2003 corpus. The latter groups 624 news articles from different sources: the ASSOCIATED PRESS

NEWSWIRE, the CHINESE XINUHA TIMES, and the NEW YORK TIMES.

Their performance was evaluated using both measures of precision and recall. In general, the negative set always had a better performance than the positive set. Low precision is mainly due to four reasons: insufficient training data, structure, binary restriction, and terminology. There are fewer reasons behind low recall, namely, the lack of evidence, and the undetected meta-knowledge. The original set of categories, which included sub-categories for *Disasters*, *Events*, and *Science*, was used for DUC 2003's Task 1 [Bergler *et al.*, 2003]. We carried several experiments to see whether or not the micro-classifiers were sensitive to any length, structure, training, or threshold parameters.

The classification of lengthier documents is slightly better than that of shorter ones. The extensive terminology in a lengthier document plays an important role for the micro-classifiers, which are able to detect the topics of the documents more accurately. However, the micro-classifiers turned out to be more sensitive to training rather than structure. For example, the performance of the micro-classifiers for originally sub-categorized *Disasters* and *Sci|Tech* is very decent. The same performance is achieved when the categories are not sub-categorized. However, when there are too many or too little training instances, precision and recall are affected. However, this is due to the (non)availability of the training data. An example of a bad performing category due to training is *Arts*.

We also examined r-cut and s-cut thresholds. The first was applied to the original set of categories, and the second to the binary set. We noticed that the best r-cut threshold was overall reached for $\Theta = 0.1$, i.e., in order to achieve best precision and recall, documents should only be assigned to a category if the probability that they belong to this category is above 90%. However, the s-cut threshold over the binary set was 0.4, i.e., a document can be assigned to a category if the probability that it belongs to this category is above 60%.

Based on a the s-cut threshold for the binary set, we collect 34 distinct multi-labels for the set of 624 testing documents. A document can be categorized simultaneously into at most five different categories, such as *Business*, *Events*, *Health*, *People*, and *Science*. We count only one document with this assignment pattern. There are five other four-labels distributed over seven documents. More frequent are three, two, and one-labeled patterns. We also count 46 documents that are not carrying a tag.

Categories	Arts	BUS	DIS	Events	Health	People	Politics	Sci|Tech	Sports
Arts	X	X	X	X	X	X	X	X	X
BUS	X								X
DIS	X			X		X	X		X
Events	X		X						
Health	X						X		X
People	X		X						
Politics	X		X		X		X	X	X
Sci|Tech	X						X		X
Sports	X	X	X		X		X	X	X

Table 5.1: Unencountered category associations

The "X" in Table 5.1 marks which categories were never assigned to the same document. The most frequent associations were EP, EPP, P, PP, NONE, BEP, D, BP, B, and E. *Business* is one of the most prominent categories which appears alone and attaches to all other categories except *Arts* and *Sports*. Category *Disasters* is only associated with *Business*, *Science*, and *Health*. Surprisingly, there is no DE record, i.e., no disaster document is classified as an event, while the event-classifier was trained to recognize disasters as events. The latter, like *Business*, glues to all other categories to the exception of *Arts* and *Disasters*. *Health* does not associate with *Politics* or *Sports*. Another passe-partout category is that of *People*. However, there are no instances of it with *Disasters*. *Politics* is rare. It either clusters with *Business*, *Events*, or *People*. Finally, *Science* does not attach to *Politics* or *Sports*, while the latter only associates with *Events* or *People*. Note that the only two categories that do not stand alone are *Politics* and *Sports*.

Given the single and multi-label attributions of documents to categories, we study their correlations with noun-phrase coreference chains. The idea is to explore whether or not patterns between a document's categories and its coreference chains exist, in which case, new ideas could be considered for summarization.

Some collected observations include the following: (1) the length of a chain is inversely proportional to the frequency of its occurrence, (2) the lengthiest chain appears in the lengthiest document, and (3) categories *Disasters* and *Sports* have the shortest documents, on average, and consequently, the least number of chains.

We account for two types of chains: singleton and non-singleton. Singleton chains consistently represent 75% of the chains in a document and are not considered to be

coreference chains, per se, in the community. Thus, when carrying analysis over coreference chains, researchers often omit these chains of length 1. As for non-singleton chains, there are on average 22–23 per document, regardless the latter's category. In other words, no correlations can be inferred at this stage.

Thus, there are no optimizations possible for the summarization system from the category and NP chain approach. We therefore propose, for future research, to base the extraction of NPs on the document's category. We can compare our results on the DUC 2004 corpus using ROUGE [Over, 2003, Lin, 2004].

Bibliography

[altavista, 2004] AltaVista. `http://www.altavista.com/`, 2004.

[Apte *et al.*, 1994] Chidanand Apte, Fred Damerau, and Sholom M. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, July 1994. Special Issue on Text Categorization.

[Bergler *et al.*, 2003] Sabine Bergler, René Witte, Michelle Khalifé, Zhuoyan Li, and Frank Rudzicz. Using knowledge-poor coreference resolution for text summarization. In *Workshop on Text Summarization*, Document Understanding Conference (DUC), Edmonton, Canada, May 31–June 1 2003. `http://duc.nist.gov/`.

[Bergler, 1997] Sabine Bergler. Towards reliable partial anaphora resolution. In *Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, Madrid, Spain, 1997. Proceedings of a workshop sponsored by the Association for Computational Linguistics.

[Chai *et al.*, 2002] Kian Ming Adam Chai, Hwee Tou Ng, and Hai Leong Chieu. Bayesian online classifiers for text classification and filtering. In Micheline Beaulieu, Ricardo Baeza-Yates, Sung Hyon Myaeng, and Kalvero Jarvelin, editors, *Proceedings of the 25th Annual International ACM SIGIR-02 Conference on Research and Development in Information Retrieval*, pages 97–104, Tampere, Finland, August 2002. ACM Press.

[Chakrabarti *et al.*, 1998] Soumen Chakrabarti, Byron Dom, and Pitor Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of ACM SIGMOD-98 International Conference on Management of Data*, pages 307–318, Seattle DC, USA, June 1998. ACM Press.

[Chen and Dumais, 2000] Hao Chen and Susan Dumais. Bringing order to the web: Automatically categorizing search results. In *Proceedings of ACM CHI-2000 Conference on Human Factors in Computing Systems*, volume 1 of *Bringing Order Out of Chaos*, pages 145–152, Hague, Netherlands, April 2000. ACM Press.

[Cohen and Singer, 1996] William W. Cohen and Yoram Singer. Context-sensitive learning methods for text categorization. In *Proceedings of the 19th Annual International ACM SIGIR-96 Conference on Research and Development in Information Retrieval*, Categorization, pages 307–315, Zurich, Switzerland, August 1996. ACM Press.

[Cohen and Singer, 1999] William W. Cohen and Yoram Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141–173, April 1999.

[Doandes, 2003] Monia Doandes. Profiling for belief acquisition from reported speech. Master's thesis, Concordia University, Montreal, Canada, April 2003.

[duc, 2004] Document Understanding Conferences (DUC). `http://duc.nist.gov/`, 2004.

[google, 2004] Google. `http://www.google.com/`, 2004.

[Harabagiu and Lacatusu, 2002] Sanda M. Harabagiu and Finely Lacatusu. Generating single and multi-document summaries with gistexter. In *Proceedings of the Second Document Understanding Conference DUC 2002*, Philadelphia PA, USA, July 2002. DUC Workshop held as part of ACL-02 Automatic Summarization Workshop.

[Hersh *et al.*, 1994] William Hersh, Chris Buckley, T.J̃. Leone, and David Hickman. Ohsumed: An interactive retrieval evaluation and new large text collection for research. In W. Bruce Croft and Cornelis J. Van Rijsbergen, editors, *Proceedings of the 18th Annual ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR-94*, pages 192–201, Dublin, Ireland, July 1994. Springer Verlag.

[Hoch, 1994] Rainer Hoch. Using IR techniques for text classification in document analysis. Research Report RR-94-19, German Research Center for Artificial Intelligence, Germany, 1994.

[Jackson and Moulinier, 2002] Peter Jackson and Isabelle Moulinier. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*, volume 5 of *NLP*. John Benjamins Publishing Company, 2002.

[Jackson, 1999] Peter Jackson. *Introduction to Expert Systems*. International Computer Science Series. Addison Wesley Longman, Harlow, England, 3rd edition, 1999.

[Joachims, 2001] Thorsten Joachims. A statistical learning model of text classification for support vector machines. In *Proceedings of the 24th ACM SIGIR-01 International Conference on Research and Development in Information Retrieval*, pages 128–136, New Orleans LA, USA, September 2001. ACM Press.

[Lam and Ho, 1998] Wai Lam and Chao Yang Ho. Using a generalized instance set for automatic text categorization. In W. Bruce Croft, Alistair Moffat, C. J.Ṽan Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR-98 Conference on Research and Development in Information Retrieval*, pages 81–89, Melbourne, Australia, August 1998. ACM Press.

[Lavelli *et al.*, 2002] Alberto Lavelli, Bernardo Magnini, and Fabrizio Sebastiani. Building thematic lexical resources by term categorization. In Micheline Beaulieu, Ricardo Baeza-Yates, Sung Hyon Myaeng, and Kalervo Jarvelin, editors, *Proceedings of the 25th Annual International ACM SIGIR-02 Conference on Research and Development in Information Retrieval*, pages 415–416, Tampere, Finland, August 2002. ACM Press.

[Lee and Myaeng, 2002] Yong-Bae Lee and Sung Hyon Myaeng. Text genre classification with genre-revealing and subject-revealing features. In Micheline Beaulieu, Ricardo Baeza-Yates, Sung Hyon Myaeng, and Kalervo Jarvelin, editors, *Proceedings of the 25th Annual International ACM SIGIR-02 Conference on Research and Development in Information Retrieval*, pages 145–150, Tampere, Finland, August 2002. ACM Press.

[Leopold and Kindermann, 2002] Edda Leopold and Jorg Kindermann. Text categorization with suport vector machines. How to represent text in input space? *Machine Learning*, 46(1/3):423–444, 2002.

[Li and Yamanishi, 2000] Hang Li and Kenji Yamanishi. Text classification using ESC-based stochastic decision lists. In *Proceedings of the 8th International Conference on Information Knowledgment CIKM-99*, pages 122–130, Kansas City MO, USA, November 2000. ACM Press.

[Lin, 2004] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Workshop on text summarization branches out*, Barcelona, Spain, July 2004.

[Liu *et al.*, 2002] Yan Liu, Yiming Yang, and Jamie Carbonell. Boosting to correct inductive bias in text classification. In *Proceedings of CIKM-02, 11th ACM International Conference on Information and Knowledge Management*, pages 348–355, McLean VA, USA, November 2002. ACM Press.

[Macskassy *et al.*, 2001] Sofus Macskassy, Hyam Hirsh, Foster Provost, Ramesh Sankaranarayanan, and Vasant Dhar. Intelligent information triage. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR-01 Conference on Research and Development in Information Retrieval*, pages 318–326, New Orleans LA, USA, September 2001. ACM Press.

[Manning and Schütze, 2001] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* The MIT Press, 2001.

[McCallum and Nigam, 1998] Andrew McCallum and Kamal Nigam. A comparison of event models for Naive Bayes text classification. In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, Madison WI, USA, July 1998.

[McCallum, 1996] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/~mccallum/bow`, 1996.

[Meretakis *et al.*, 2000] Dimitris Meretakis, Dimitris Fragoudis, Hongjun Lu, and Spiros Likothanassis. Scalable association-based text classification. In Arvin Agah,

Jamie Callan, and Elke Rundensteiner, editors, *Proceedings of the 2000 ACM CIKM-00 International Conference on Information and Knowledge Management*, pages 5–11, McLean VA, USA, November 2000. ACM Press.

[Mooney and Roy, 2000] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the 5th ACM Conference on Digital Libraries DL-00*, Full Papers, pages 195–204, San Antonio TX, USA, June 2000. ACM Press.

[msn, 2004] MSN. `http://www.msn.com/`, 2004.

[muc, 2004] Message Understanding Conference (MUC). `http://www.itl.nist.gov/iaui/894.02/related_projects/muc`, 2004.

[newsblaster, 2004] Columbia Newsblaster: Summarizing all the news on the web. `http://www1.cs.columbia.edu/nlp/newsblaster`, 2004.

[Ng *et al.*, 1997] Hwee T. Ng, Wei B. Goh, and Kok L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In Nicholas J. Belkin, A. Desai Narasimhalu, and Peter Willett, editors, *Proceedings of the 20th Annual International ACM SIGIR-97 Conference on Research and Development in Information Retrieval*, pages 67–73, Philadelphia PA, USA, July 1997. ACM Press.

[Nottelmann and Fuhr, 2001] Henrik Nottelmann and Norbert Fuhr. Learning probabilistic datalog rules for information classification and transformation. In *Proceedings of the 10th Annual International ACM CIKM-01 Conference on Information and Knowledge Management*, pages 387–394, McLean VA, USA, November 2001.

[Over, 2003] Paul Over. An introduction to DUC 2003: Intrinsic evaluation of generic news texts summarization systems. In *Workshop on text summarization*, Document Understanding Conference (DUC), Edmonton, Canada, May 31–June 1 2003. `http://duc.nist.gov/`.

[Quinlan, 1993] J.Ross Quinlan. *C4.5: Programming for Machine Learning.* Machine Learning. Morgan Kaufmann, 1993.

[Ragas and Koster, 1998] Hein Ragas and Cornelis H.Ã. Koster. Four text classification algorithms compared on a Dutch corpus. In *Proceedings of the 21st Annual*

International ACM SIGIR-98 Conference on Research and Developmesnt in Information Retrieval SIGIR-98, Posters, pages 369–370, Melbourne, Australia, August 1998. ACM Press.

[Riloff, 1995] Ellen Riloff. Little words can make a big difference for text classification. In *Proceedings of the 18th Annual International ACM SIGIR-95 Conference on Research and Development in Information Retrieval*, Natural Language Processing, pages 130–136, Seatle WA, USA, July 1995. ACM Press.

[Ruiz and Srinivasan, 1999] Miguel E. Ruiz and Padmini Srinivasan. Hierarchical neural networks for text categorization. In Marti AḢearst, Fredric Gey, and Richard Tong, editors, *Proceedings of the 22nd Annual International ACM SIGIR-99 Conference on Research and Development in Information Retrieval*, pages 281–282, Berkeley CA, USA, August 1999. ACM Press.

[Sebastiani, 2002] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002.

[Slonim *et al.*, 2002] Noam Slonim, Nir Friedman, and Naftali Tishby. Unsupervised document classification using sequential information maximization. In Micheline Beaulieu, Ricardo Baeza-Yates, Sung Hyon Myaeng, and Kalervo Jarvelin, editors, *Proceedings of the 25th Annual International ACM SIGIR-02 Conference on Research and Development in Information Retrieval*, pages 11–15, Tampere, Finland, August 2002. ACM Press.

[Surdeanu and Harabagiu, 2002] Mihai Surdeanu and Sanda M. Harabagiu. Infrastructure for open-domain information extraction. Poster Presentation at HLT 2002 Conference, March 2002.

[Thompson, 2001] Paul Thompson. Automatic categorization of case law. In *Proceedings of the 8th International ICAIL-01 Conference on Artificial Intelligence and Law*, pages 70–77, St. Louis MO, USA, May 2001. ACM Press.

[trec, 2004] Text REtrieval Conference (TREC). `http://trec.nist.gov/`, 2004.

[vivisimo, 2004] Vivisimo Custering Engine. `http://vivisimo.com/`, 2004.

[Witte and Bergler, 2003] René Witte and Sabine Bergler. Fuzzy coreference resolution for summarization. In *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 45–50, Venice, Italy, June 23–24 2003. Università Ca' Foscari. `http://rene-witte.net`.

[Witte *et al.*, 2004] Rene Witte, Sabine Bergler, Zhuoyan Li, Michelle Khalifé, Yunyu Chen, Monia Doandes, and Alina Andreevskaia. Multi-ERSS and ERSS 2004. In *Workshop on Text Summarization*, Document Understanding Conference (DUC), Boston MA, USA, May 2004. `http://duc.nist.gov/`.

[Witte, 2002] René Witte. Fuzzy belief revision. In *9th International Workshop on Non Monotonic Reasoning (NMR '02)*, pages 311–320, Toulouse, France, April 19–21 2002. `http://rene-witte.net`.

[Yang and Chute, 1994] Yiming Yang and Christopher G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12(3):252–277, July 1994. Special Issue on Text Categorization.

[Yang and Liu, 1999] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR-99 Conference on Research and Development in Information Retrieval*, Speech IR & Text Categorization, pages 42–49, Berkley CA, USA, August 1999. ACM Press.

[Yang *et al.*, 2000] Yiming Yang, Tom Ault, Thomas Pierce, and Charles W. Lattimer. Improving text categorization methods for event tracking. In *Proceedings of the 23rd Annual International ACM SIGIR-00 Conference on Research and Development in Information Retrieval*, Topic Detection and Tracking, pages 65–72, Athens, Greece, July 2000. ACM Press.

[Yang, 1999] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1:69–90, 1999.

[Yang, 2001] Yiming Yang. A study on thresholding strategies for text categorization. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors,

Proceedings of the 24th Annual International ACM SIGIR-01 Conference on Research and Development in Information Retrieval, pages 137–145, New Orleans LA, USA, September 2001. ACM Press.

[Zhang and Oles, 2001] Tong Zhang and Frank J. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, 4(1):5–31, 2001.

Appendix A

Similarity metrics

Given two binary vectors X and Y, the Euclidean Distance is given by

$$E(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

and the Cosine Measure is given by

$$\cos(X,Y) = \frac{[X \cap Y]}{\sqrt{[X] \times [Y]}}$$

Appendix B

Clustering Algorithms

Clustering algorithms work over *bags*, "objects like sets except that they allow multiple identical items" [Manning and Schütze, 2001], and do not require labeled data to partition a set of objects into groups or *clusters*. However, we distinguish *hard clustering*, in which an object is assigned to only one cluster, from *soft clustering*, which allows degrees of membership and membership in multiple clusters.

In text classification, clustering algorithms attempt to group similar documents under the same cluster, producing either a *hierarchical clustering* or a less informative *flat clustering*. While the latter consists of a number of unrelated clusters, hierarchical clustering presents a hierarchy of clusters related to each other by the common parent-child relations.

B.1 Hierarchical Clustering

Hierarchical clustering is usually based on hard assignment and the tree is built either bottom-up (*agglomerative clustering*) or top-down (*devisive clustering*).

B.1.1 Agglomerative clustering

Agglomerative clustering starts with a separate cluster for each document. At every iteration, a similarity function determines which two clusters should be merged. The algorithm terminates when only one large cluster is left. There are three similarity functions used in agglomerative clustering and illustrative examples can be found in [Manning and Schütze, 2001]:

Single-Link Clustering: Single-link assigns the two most similar members to the same cluster. Its tendency to sometimes follow a chain of large similarities produces an *elongated cluster.*

Complete-Link Clustering: Complete-link assigns the two least similar members to the same cluster. It avoids elongated clusters by focusing on *global context.* Its disadvantage is its $O(n^3)$ time complexity.

Group-Average Clustering: Group-average uses the Cosine measure to compute the average similarity between members and is a compromise between Single-link and Complete-link. It is not only computed more efficiently than Complete-link (i.e. $O(n^2)$) but it also stays away from elongated clusters produced by Single-link.

B.1.2 Devisive clustering

Devisive clustering starts with one cluster that contains all documents and attempts to split the least coherent cluster at every iteration, by using one of above-mentioned similarity functions. Splitting the least coherent cluster is the task of finding two subclusters of the cluster.

B.2 Non-Hierarchical Clustering

As opposed to hierarchical clustering, non-hierarchical clustering employs multiple passes to reallocate documents to the best current cluster. The algorithm starts out with a random partition of clusters and keeps redistributing documents until the curve of improvement flattens or when goodness starts decreasing. Examples of flat clustering algorithms include *K-means*, L_1 *norm* and the *EM algorithm.*

K-means starts out with an initial set of cluster centers called *centroids.* Documents undergo hard assignment to the cluster with the closest center. At the end of every iteration, the center of mass of each cluster has to be recomputed. If there is more than one center with the same distance to a document, the latter is randomly allocated to one of the clusters. Another possibility would be to slightly change the position of the documents in a way that does not allow ties to rise.

L_1 **norm** is a variant of K-means that uses *medoids* rather than centroids as cluster centers. A medoid is one of the objects in the cluster while a centroid (being the mean) is not identical to any of the objects. Both K-means and L_1 norm have $O(n)$ time complexity and require several iterations before the algorithm converges.

The EM algorithm is the soft assignment version of K-means which also starts with a set of cluster centers and tries to estimate the values of the parameters of the model that maximizes the likelihood of the data. EM is short for *Expectation Maximization*. The algorithm is an iterative solution that begins with the E-step and computes cluster membership probabilities. Given the expected values, the M-step follows and computes the most likely parameters of the model.

B.2.1 Sequential Clustering Algorithms

Sequential clustering algorithms were motivated by the Information Bottleneck method (*sIB*), and proposed by [Slonim *et al.*, 2002]. They do not suffer from time and space complexities, unlike agglomerative procedures, which furthermore have no guarantees on finding the solution that is a local maximum of the target function. The method outperforms agglomerative clustering, K-means and its results are comparable to those obtained by a supervised Naive Bayes classifier [Slonim *et al.*, 2002].

Appendix C

Training the micro-classifiers

Politics The category was trained on 14 DUC 2002 clusters and three newsgroups provided in the 20 Newsgroups corpus. The data is displayed in Table C.1.

Politics	
d063j	Bomb blast in Britain
d070f	East Germany under Honeker
d068f	Check point Charlie
d069f	German reunification
d076b	Margaret Thatcher retirement
d082a	Soviet Sakharov
d086d	Kuwait Iraq invasion
d094c	Japan's emperor
d106g	Mandela's imprisonment
d109h	Chinese manifestation
d111h	USSR politics
d114h	Trouble India and Kashmir
d118i	John Tower
d120i	John Major
20NG	talk.politics.guns
20NG	talk.politics.mideast
20NG	talk.politics.misc

Table C.1: Training data for category *Politics*

Business The category was trained on four clusters from the DUC 2002 corpus as well as a selection of 290 Wall Street Journal articles. The data appears in Table C.2.

Sci|Tech The multiple positive categories here were trained on eight newsgroups gathered from the 20 Newsgroups corpus and three DUC 2002 clusters. Data

Business	
d064j	Mc Donald's in Moscow
d066j	Sam Walton and WalMart
d081a	Mining worker's strike
d112h	Maxwell economics
WSJ	290 articles

Table C.2: Training data for category *Business*

Sci|T	d075b	Treatment for heart attacks
	20NG	comp.graphics
	20NG	comp.sys.ibm.pc.hardware
	20NG	comp.windows.x
	20NG	comp.sys.mac.hardware
	20NG	comp.os.ms-windows.x
	20NG	sci.crypt
	20NG	sci.electronics
	20NG	sci.med
Sci|T.Computers	20NG	comp.graphics
	20NG	comp.sys.ibm.pc.hardware
	20NG	comp.windows.x
	20NG	comp.sys.mac.hardware
	20NG	comp.os.ms-windows.x
Sci|T.Space	20NG	sci.space
	d098e	Space missions
	d116i	Discovery shuttle

Table C.3: Training category *Sci|Tech*

redundancy in *Sci|T* is meant for experimental purposes. The data is presented in Table C.3.

Health The category was trained on one DUC 2002 cluster, one newsgroup, and some articles collected from online news sources. Table C.4 lists these clusters.

Health	
d075b	Treatment for heart attacks
20NG	sci.med

Table C.4: Training category *Health*

Sports The category was trained on three DUC 2002 clusters and two newsgroups from the 20 Newsgroups corpus. The data can be found in Table C.5.

Arts The category was trained on five DUC 2002 clusters and some web-collected articles from online news sources' sections such as *Arts* and *Entertainment*. The DUC clusters are shown in Table C.6.

Sports	
d087d	Olympic results
d096c	SuperBowl
d099e	Marathons
20NG	rec.sports.baseball
20NG	rec.sports.hockey

Table C.5: Training category *Sports*

Arts	
d072f	Leonard Bernstein's birthday
d078b	Oscars
d100e	John Lennon
d102e	Lucille Ball
d117i	Booker prize

Table C.6: Training data for category *Arts*

Natural Disasters Three of the different disaster categories here were trained on two DUC 2002 clusters and one was trained on five clusters. Table C.7 lists the clusters for the different categories.

Disasters	d073b	Mt Pinatubo Volcano
	d107g	Drought
Disasters.Earthquakes	d062j	San Fransisco Earthquake
	d092c	Iranian Earthquake
Disasters.Floods	d091c	Flash Floods
	d109h	Chinese Floods
Disasters.Storms	d061j	Hurricane Gilbert
	d067f	Hurricane Andrew
	d083a	Tornado Storms
	d085d	Hurricane Hugo
	d089d	Thunderstorms and Tornadoes

Table C.7: Training data for category *Disasters*

Events The two positive categories here were each trained on 14 distinct DUC 2002 clusters. Table C.8 gives an overview of what constitutes a single or a multiple event.

People The category was trained with 14 biographical clusters from the DUC 2002 corpus. These are displayed in Table C.9.

Single Events	d063j	Bomb Blast in Britain
	d068f	Check Point Charlie
	d074b	Clarence Thomas Hearing
	d080a	Preschool Molest
	d084a	Battleship Explosion
	d086d	Kuwait's Iraqi Invasion
	d092c	Iranian Earthquake
	d095c	School Shooting
	d098e	Space Mission
	d101e	Russian Presidents and Politics
	d104g	Chinese Manifestations
	d110h	John Lennon
	d113h	Explosion in India
	d116i	Discovery Launch
Multiple Events	d064j	Mc Donald's in Moscow
	d069f	German Reunification
	d071f	Dog Lunch
	d075b	Treatment for Heart Attacks
	d078b	Oscars
	d081a	Mining Workers Strike
	d087d	Olympic Results
	d093c	Ferry Sinkings
	d096c	SuperBowl
	d099e	Marathons
	d105g	Soviet Clashes
	d111h	USSR Politics
	d117i	Booker Prize
	d119i	Politics Miscellaneous

Table C.8: Training data for category *Events*

People	
d065j	Dan Quayle
d066j	Sam Walton
d070f	East Germany under Honecker
d072f	Leonard Bernstein's Birthday
d076b	Margaret Thatcher Retirement
d082a	Soviet Sakharov
d090d	Philippine's President
d094c	Japan's Emperor
d100e	John Lennon
d102e	Lucille Ball
d106g	Mandela Imprisonment
d112h	Maxwell Economics
d118i	John Tower
d120i	John Major

Table C.9: Training data for category *People*

Appendix D

Evaluation results

We present the detailed results obtained for precision and recall over the modified categories *Disasters*, *Events*, *Politics*, and *Sci|Tech*. We also show the data that was used for training these categories. In the last section, we show the detailed results for precision and recall over the binary set of categories.

D.1 Training

We first present the training data, then the detailed evaluation results.

Disasters The training data for category *Disasters* is found in Table D.1.

Disasters	d061j	Hurricane Gilbert
	d062j	San Fransisco Earthquake
	d067f	Hurricane Andrew
	d073b	Mt Pinatubo Volcano
	d083a	Tornado Storms
	d085d	Hurricane Hugo
	d089d	Thunderstorms and Tornadoes
	d091c	Flash Floods
	d092c	Iranian Earthquake
	d107g	Drought
	d109h	Chinese Floods

Table D.1: Training data for binary category *Disasters*

Events The training data for category *Events* is found in Table D.2.

Politics The training data for category politics is found in Table D.3.

Event.singletimestamp	d063j	Bomb Blast in Britian
	d067f	Hurricane Andrew
	d072f	Leonard Bernstein's birthday
	d073b	Mt Pinatubo volcano
	d077b	San Fransisco earthquake
	d079a	Hurricane Gilbert
	d080a	Preschool molest
	d084a	Battleship explosion
	d089d	Thunderstorms and tornadoes
	d092c	Iranian earthquake
	d097e	Hurricane Hugo
	d107g	Drought
	d113h	Explosion in India
	d083a	Tornado storms
	d091c	Flash floods
	d095c	School shooting
	d109h	Chinese floods
	d116i	Discovery launch
Event.multipletimestamp	d064j	McDonald's in Moscou
	d068f	Check point Charlie
	d069f	German reunification
	d074b	Clarence Thomas hearing's
	d078b	Oscars
	d081a	Mining workers strike
	d086d	Kuwait Iraqi invasion
	d087d	Olympics results
	d093c	Ferry sinkings
	d096c	SuperBowl
	d098e	Space missions
	d099e	Marathons
	d104g	Chinese manifestations
	d105g	Soviet clashes
	d117i	Booker prize
Event.notimestamp	d082a	Soviet Sakharov
	d090d	Philippine's socio-polico-economy
	d094c	Japan's emperor
	d101e	Russian presidents and politics
	d114h	India and Kashmir trouble
	d118i	Politics John Tower
	d119i	Politics Misc
	d120i	Politics John Major

Table D.2: Training data for modified category *Events*

Sci|Tech The training data for category Sci|Tech is found in Table D.4.

The evaluation results for the modified categories are found in Table D.5

Politics.World	d068f	Check point Charlie
	d069f	German reunification
	d086d	Kuwait Iraq invasion
	d101e	Russian presidents and politics
	d105g	Soviet clashes
	d109h	Chinese manifestation
	d114h	Trouble India and Kashmir
	20NG	talk.politics.mideast
Politics.Terrorism	d063j	Bomb blast in Britain
	d113h	Explosion in India
	20NG	talk.politics.guns
Politics.Miscellaneous	d118i	John Tower
	d120i	John Major
	d119i	Politics Misc
	20NG	talk.politics.misc

Table D.3: Training data for modified category *Politics*

Sci|T.computers	20NG	comp.graphics
	20NG	comp.sys.ibm.pc.hardware
	20NG	comp.windows.x
	20NG	comp.sys.mac.hardware
	20NG	comp.os.ms-windows.x
	20NG	sci.crypt
Sci|T.electronics	20NG	sci.electronics
Sci|T.medicine	20NG	sci.med
	d075b	Treatment for heart attacks
Sci|T.space	20NG	sci.space
	d098e	Space missions
	d116i	Discovery shuttle

Table D.4: Training data for modified category *Sci|Tech*

D.2 Orthogonal categories

The detailed evaluation results for the orthogonal set of categories is found in Table D.6.

Categories	$\delta \leq 0.1$		$\delta \leq 0.2$		$\delta \leq 0.3$		$\delta \leq 0.4$	
	Precision	Recall	Preision.	Recall	Precision	Recall	Precision	Recall
STS E	0.217	0.709	0.216	0.709	0.215	0.709	0.218	0.720
MTS E	0.377	0.419	0.377	0.419	0.374	0.419	0.375	0.423
NTS E	0.005	0.006	0.005	0.006	0.005	0.006	0.005	0.006
NE	0.160	0.702	0.162	0.712	0.162	0.712	0.161	0.712
Pol-W	0.235	0.358	0.238	0.364	0.237	0.364	0.237	0.364
Pol-T	0	0	0	0	0	0	0	0
Pol-M	0.197	0.4	0.200	0.409	0.200	0.409	0.200	0.409
NPol	0.555	0.894	0.554	0.897	0.550	0.897	0.550	0.897
TCI	0	0	0.027	0.029	0.027	0.029	0.027	0.294
TE	0	0	0	0	0	0	0	0
TM	0.201	0.956	0.201	0.956	0.201	0.956	0.201	0.956
TS	0.244	0.939	0.242	0.939	0.242	0.939	0.240	0.939
NT	0.891	0.978	0.890	0.978	0.890	0.978	0.890	0.978

Table D.5: Precision and recall for modified categories *Events*, *Politics*, and *Sci|Tech*

Categories	$\delta \leq 0.1$			$\delta \leq 0.2$			$\delta \leq 0.3$			$\delta \leq 0.4$		
	pre	rec	f_1	pre	rec	f_1	pre	rec	f_1	pre	rec	f_1
Arts	0	0	0	0	0	0	0	0	0	0	0	0
Not Arts	0.983	0.996	0.989	0.983	0.998	0.990	0.983	0.998	0.990	0.983	1	0.991
Business	0.747	0.893	0.813	0.736	0.893	0.806	0.730	0.893	0.803	0.730	0.893	0.803
Not Business	0.981	0.925	0.952	0.981	0.925	0.952	0.981	0.929	0.954	0.979	0.932	0.954
Disaster	0.853	0.660	0.744	0.853	0.660	0.744	0.833	0.660	0.736	0.833	0.660	0.736
Not Disaster	0.970	0.985	0.977	0.969	0.987	0.977	0.969	0.987	0.977	0.969	0.987	0.977
Event	0.974	0.615	0.753	0.974	0.630	0.765	0.972	0.637	0.769	0.969	0.643	0.773
Not Event	0.235	0.777	0.360	0.237	0.805	0.366	0.241	0.833	0.373	0.239	0.833	0.371
Health	0.961	0.581	0.724	0.961	0.581	0.724	0.961	0.581	0.724	0.961	0.581	0.724
Not Health	0.971	0.998	0.984	0.971	0.998	0.984	0.971	0.998	0.984	0.971	0.998	0.984
People	0.969	0.863	0.912	0.966	0.865	0.912	0.966	0.865	0.912	0.967	0.869	0.915
Not People	0.615	0.863	0.718	0.615	0.863	0.718	0.612	0.863	0.716	0.610	0.871	0.717
Politics	0.951	0.541	1.637	0.941	0.559	0.701	0.941	0.562	0.703	0.942	0.569	0.709
Not Politics	0.736	0.967	0.835	0.734	0.970	0.835	0.732	0.970	0.834	0.730	0.970	0.833
Sci|Tech	0.793	0.418	0.547	0.774	0.436	0.557	0.774	0.436	0.557	0.774	0.436	0.557
Not Sci|Tech	0.949	0.985	0.966	0.949	0.985	0.966	0.949	0.985	0.966	0.947	0.985	0.965
Sports	1	0.542	0.702	1	0.542	0.702	1	0.542	0.702	1	0.542	0.702
Not Sports	0.976	1	0.987	0.976	1	0.987	0.976	1	0.987	0.976	1	0.987

Table D.6: Precision, recall, and f-measure, for orthogonal categories