UNIVERSITY OF
LIVERPOOL

Language-independent Pre-processing of Large Documentbases for Text Classification

A thesis submitted in accordance with the requirements of the
University of Liverpool for the degree of Doctor in Philosophy
by

Yanbo J. Wang

Thesis Advisors
Doctor Frans Coenen
Professor Paul Leng

Department of Computer Science
The University of Liverpool

Abstract

Text classification is a well-known topic in the research of knowledge discovery in databases. Algorithms for text classification generally involve two stages. The first is concerned with identification of textual features (i.e. words and/or phrases) that may be relevant to the classification process. The second is concerned with classification rule mining and categorisation of "unseen" textual data. The first stage is the subject of this thesis and often involves an analysis of text that is both language-specific (and possibly domain-specific), and that may also be computationally costly especially when dealing with large datasets. Existing approaches to this stage are not, therefore, generally applicable to all languages. In this thesis, we examine a number of alternative keyword selection methods and phrase generation strategies, coupled with two potential significant word list construction mechanisms and two final significant word selection mechanisms, to identify such words and/or phrases in a given textual dataset that are expected to serve to distinguish between classes, by simple, language-independent statistical properties. We present experimental results, using common (large) textual datasets presented in two distinct languages, to show that the proposed approaches can produce good performance with respect to both classification accuracy and processing efficiency. In other words, the study presented in this thesis demonstrates the possibility of efficiently solving the traditional text classification problem in a language-independent (also domain-independent) manner.

Acknowledgements

On the cover of this Ph.D. thesis, there is only one name printed, however without the help, support, assistance and understanding of others, this thesis would not exist. I am very glad to have the opportunity to write a few words to the people and organisations who provided help to me and made it all possible.

First of all, I would like to thank Universities UK; the China Scholarship Council; and the University of Liverpool. Funding was provided by an Overseas Research Students Awards Scheme (ORSAS) scholarship; a 2006 Chinese Government Award for Outstanding Self-financed Students Abroad; and an International Research Scholarship. They all provided financial support, crucial to my study.

Second, I wish to thank my Ph.D. supervisors Dr. Frans Coenen and Professor Paul Leng. It is my greatest pleasure to be their student. Without their supervision, guidance, understanding and encouragement, there is no doubt that my Ph.D. study would not have been completed. I also wish to thank Professor Leszek A. Gąsieniec, my Ph.D. advisor, who has provided me with valuable advice during my period of study.

Third, I would like to thank two of my previous teachers, Professor Peter Khaiter of the J. E. Atkinson Faculty of Liberal and Professional Studies at York University (Canada), and Mr. John Jose of the Department of Mathematics at Alexander Mackenzie High School (Canada). I also want to thank Dr. Qin Xin of Simula Research Laboratory (Norway), who is a good friend and teacher as well.

Fourth, during my Ph.D. study many members of the Department of Computer Science (at the University of Liverpool) have helped and shown their kindness to me. I am indebted to Katie Atkinson, Irina Biktasheva, Andrey Bovykin, Helen Bradley, Paul Dunne, Judith Lewa, Grant Malcolm, Peter McBurney, Rob Sanderson, Thelma Williams, Prudence Wong, Adam Wyner, Michele Zito, and a number of current and past Ph.D. students including: Kamal

Ali Albashiri, Omar Baqueiro, Farnaz Derakhshan, Chuntao Jiang, Cindy (Ying) Li, Sunchai Meenorngwar, Mark Roberts, Ji Ruan, Chang Su, Maya Wardeh and Xiaohui Zhang.

Fifth, I would like to thank a number of people who have made my life in the UK memorable: Leiwen Chen, Ronghuan Chen, Wen Dong, Zhicheng Dong, Dan Du, Meng Duan, Haifeng Guo, Xiaoguang Guo, Yu He, Yibin Huang, Shengbin Jia, Jiang Jiang, Xiaohan Jiang, Wei Lei, You Li, Hanyu Lian, Yan Liang, Ke Liu, Maybin Muyeba, Rui Sun, Lulu Wang, Xiaojun Wang, Xiaomeng Xie, Xiaofeng Yang, Yao Yao, Jianfei Yu, Sha Yu, Haiyang Zhao, Yu Zhao, Lei Zheng and Xinwei Zheng.

Sixth, during the period of writing my Ph.D. thesis, many people have encouraged me including: Jiaming Gong, Zhesi He, Qihan Jiang, Xiaoming Jiang, Yupeng Jiao, Jiyuan Li, Wei Luo, Yaning Ren, Kang Su, Haoda Sun, Xinyu (Brenda) Tang, Ashley (Yuehnuan) Tsai, Mandy (Miaoying) Tsai, Songpo Wang, Ying Wang, Yitong Wang, Zhao Wang, Haijing Wei, Alice Wu, Changjiang Yu, Karen Zhang and Ying Zhao. I also wish to give special thanks to Jane Tsai, who has provided moral and mental support to me during the past five years.

Finally, I feel that I owe a lot to my parents. I truly understand how much they love me, and wish to thank them for giving it.

To my dearest parents and Jane Tsai

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

The increasing number of electronic documents that are available to be explored on-line has led to text mining [Hotho *et al.*, 2005] becoming a promising school of current research in Knowledge Discovery in Databases (KDD) [Piatetsky-Shapiro and Frawley, 1991], that is attracting more and more attention from a wide range of different groups of people. Text mining aims to extract various types/models of hidden, interesting, previously unknown and potentially useful knowledge (i.e. rules, patterns, regularities, customs, trends, etc.) from sets of collected textual data (i.e. news, mails/e-mails, magazine articles, academic papers, natural language speeches, etc.), where the volume of a collected textual data set can be measured in gigabytes. In text mining, a given textual data set is commonly refined to produce a documentbase, i.e. a set of electronic documents that typically consists of thousands of documents, where each document may contain hundreds of words. One important application of text mining is Text Classification/Categorisation (TC) [Liu *et al.*, 2004] [Sebastiani, 2002] [Sebastiani, 2005] [Yang and Liu, 1999] [Zhuang *et al.*, 2005] — the automated categorisation of "unseen" documents into pre-defined classes/groups. Other common text mining applications include: document clustering [Dhillon *et al.*, 2004], topic detection and tracking [Allan, 2002], text segmentation [Beeferman *et al.*, 1999], text summarisation [Mani and Maybury, 1999], text visualisation [Hotho *et al.*, 2005], etc.

TC, during the last decade, has been well investigated as an intersection of research into KDD and machine learning [Mitchell, 1997] [Michalski *et al.*, 1983] [Michalski *et al.*, 2006]. The distinction is that KDD based TC investigation (such as that undertaken by Antonie and Zaïane [2002]) aims, in general, to apply statistical data mining techniques; while machine learning based TC approaches (such as [Sebastiani, 2002]) focus on various artificial intelligence techniques. Anand *et al.* [1995] argue that the major drawback of machine learning based data investigations is that "*most machine learning algorithms get quite inefficient when it comes to using them with large quantities of data*".

TC algorithms can be broadly divided into two significant groups: (1) single-label, which assigns exactly one pre-defined class to each "unseen" document, and (2) multi-label, which assigns one or more pre-defined class to each "unseen" document. With regard to single-label TC, two distinct approaches can be identified: (i) binary (positive and negative) TC [Sebastiani, 2002] [Wu *et al.*, 2002], where the classes are processed iteratively and for all "unseen" documents it is determined whether the document belongs to the target class or not, and (ii) multi-class TC [Berger and Merkl, 2004], which simultaneously considers all the pre-defined classes and determines to which particular class each "unseen" document should be assigned. The focus of this thesis is the single-label multi-class TC. Furthermore Sun and Lim [Sun and Lim, 2001] divide TC approaches into hierarchical and "flat" (non-hierarchical), where hierarchical classification relates to the assignment of one or more suitable classes from a (pre-defined) hierarchical class space to each "unseen" document. Note that the hierarchical TC is beyond the scope of this thesis.

Text mining requires the given documentbase to be first pre-processed so that it is in an appropriate format (i.e. an "intermediate form" [Tan, 1999]). Thus the process of TC, in general, can be identified as documentbase pre-processing plus Classification Rule Mining (CRM) [Quinlan, 1993] [Liu *et al.*, 1998]. The nature of the pre-processing can be characterised as:

1. **Documentbase Representation**, which designs an application-oriented data model to precisely interpret a given documentbase in an explicit and structured manner; and

2. **Feature Selection**, which extracts the most significant information (text-features) from the given documentbase.

In TC, many documentbase pre-processing mechanisms use specialised language-dependent techniques to identify key words and/or phrases (i.e. stop-word lists, synonym lists, stemming, lemmatisation, part-of-speech tagging, etc.) [Forman, 2003] [Goncalves and Quaresma, 2004] [Goncalves and Quaresma, 2005] [Mladenic, 1999]. These techniques operate well with regard to the accuracy of classification, but are designed with particular languages and styles of language as the target. They are therefore not usually applicable to all languages (e.g. Arabic, Chinese, Spanish). In many applications of TC there may be a need to examine cross-lingual, multi-lingual and/or unknown-lingual textual data collections, where such specialised language-dependent techniques are not available. Bel *et al.* [2003] consider that:

1. The documentation departments of multinational and international organisations may desire to automatically classify a number of cross-lingual business documents; and
2. Many search engines on the web may be expected to automatically classify a number of cross-lingual web documents.

It is also the case that many language-dependent documentbase pre-processing techniques involve deep linguistic analysis, and as a result they have drawbacks in terms of computational efficiency.

The above motivates the research described in this thesis in which we search for ways of performing documentbase pre-processing for TC without involving deep linguistic analysis. In the past few decades traditional TC problems have been well researched. However, only a very few studies have been concerned with such language-independent TC approaches (e.g. [Peng *et al.*, 2003]). The work described in this thesis systematically investigates approaches to single-label multi-class TC that use only statistical and other language-independent techniques for documentbase pre-processing. It seeks to establish whether such approaches can produce acceptable performance with respect to both the accuracy of classification and the efficiency of computation for the TC problem.

1.2 Research Strategy

Rule-based classification systems, in general, begin with a process of Classification Rule Mining (CRM) to identify a set of Classification Rules (CRs) in a given training set of data. One recent approach to CRM is to employ Association Rule

Mining (ARM) [Agrawal and Srikant, 1994] techniques to identify the desired Classification Rules, i.e. Classification Association Rule Mining (CARM) [Ali *et al.*, 1997]. Coenen *et al.* [2005] and Shidara *et al.* [2007] suggest that results presented in the studies of [Li *et al.*, 2001] [Liu *et al.*, 1998] and [Yin and Han, 2003] show that in many cases CARM offers higher classification accuracy than other traditional CRM methods, such as C4.5 [Quinlan, 1993] and RIPPER (Repeated Incremental Pruning to Produce Error Reduction) [Cohen, 1995].

With regard to a number of CRM techniques available, CARM is suggested to offer the following advantages [Antonie and Zaïane, 2002] [Yoon and Lee, 2005]:

1. The approach is efficient during both the training and categorisation phases, especially when handling a large volume of data; and
2. The classifier, built in this approach, can be read, understood and modified by humans.

Furthermore CARM based TC is relatively insensitive to noise data (for example where some document words are misspelt/miswritten). CARM builds a classifier by extracting a set of Classification Association Rules (CARs) from a given set of training instances. Possible CARs are determined by: (i) a large enough *support* — the overall frequency in the training set of instances where the rule applies; and (ii) a large enough *confidence* — the support of the rule in relation to the support of its antecedent. Usually, rules derived from noise in the data will fail to reach these thresholds and will be discarded.

For these reasons it is proposed to use a CARM approach to address the language-independent TC problem. One of the existing CARM frameworks is the TFPC (Total From Partial Classification) algorithm [Coenen and Leng, 2004] [Coenen *et al.*, 2005] [Coenen and Leng, 2007]. TFPC, based on the Apriori-TFP ARM algorithm [Coenen *et al.*, 2004a] [Coenen *et al.*, 2004b], generates CARs (from a given set of training instances) by an efficient use of set enumeration tree structures. Experimental results using this algorithm reported in [Coenen *et al.*, 2005] show that it can achieve high classification accuracy for a range of data sets.

1.3 Contribution of This Thesis

In this thesis, a number of statistical, language-independent, documentbase pre-processing strategies are examined. Some of them have been presented previously in [Wang *et al.*, 2006] and [Coenen *et al.*, 2007]. In total 16 different documentbase pre-processing schemes (see Figure 1.1) are introduced under the "*bag of words*" heading (4 statistical keyword selection methods × 2 potential significant word list construction mechanisms × 2 final significant word selection mechanisms), and 64 schemes (see Figure 1.2) are proposed under the "*bag of phrases*" heading (4 significant phrase identification mechanisms × 16 "bag of words" or significant word identification schemes). These strategies are used in conjunction with the TFPC algorithm to generate classifiers. The experimental work described here evaluates the success of each strategy by examining the accuracy and performance of the classifiers thus derived. The experimental results show that using the common textual datasets presented in two distinct languages, a number of the proposed language-independent documentbase pre-processing strategies produce acceptable classification accuracy whilst being efficient in terms of processing time. In other words, the work presented in this thesis demonstrates the possibility of efficiently solving the traditional text classification problem in a language-independent fashion. This approach is generally applicable to cross-lingual, multi-lingual and/or unknown-lingual document collections.

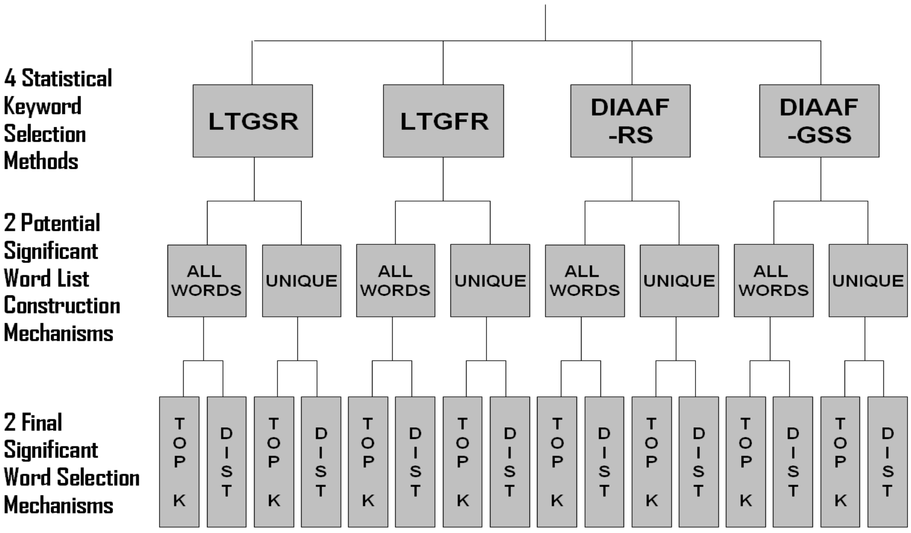

4 Statistical Keyword Selection Methods
2 Potential Significant Word List Construction Mechanisms
2 Final Significant Word Selection Mechanisms
LTGSR
LTGFR
DIAAF -RS
DIAAF -GSS
ALL WORDS
UNIQUE
TOP K
DIST

Figure 1.1: Sixteen "bag of words" documentbase pre-processing schemes

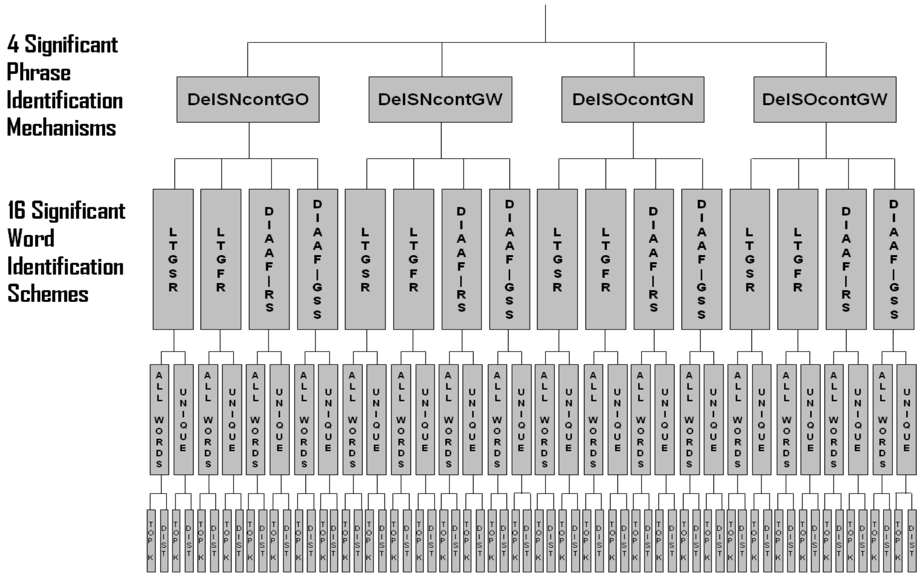


Figure 1.2: Sixty-four “bag of phrases” documentbase pre-processing schemes

1.4 Thesis Outline

The following chapter describes the background of current KDD research with respect to a variety of technologies and/or methodologies in both data mining in general and text mining in particular. In chapter 3 an overview of the existing documentbase pre-processing techniques for TC is provided. The proposed language-independent documentbase pre-processing mechanisms/strategies are presented in chapter 4. In chapter 5, experimental results are obtained using the TFPC CARM algorithm, and demonstrate that the proposed mechanisms/strategies can produce acceptable performance with respect to both the accuracy of classification and the efficiency of computation for TC. Finally overall conclusions are drawn and a number of issues for further research are discussed in chapter 6.

Chapter 2

Background on Knowledge Discovery in Databases

2.1 Introduction

The aim of this chapter is to systematically review research relating to Knowledge Discovery in Databases (KDD) with respect to a variety of technologies and/or methodologies in both data mining and text mining. The organisation of this chapter is as follows. An overview of KDD is presented in section 2.2, where the overall KDD process is described and various KDD "schools" are summarised. In section 2.3 a general description of data mining is provided. Association Rule Mining (ARM), and three well-established ARM approaches are reviewed in section 2.4, while two particular ARM methods, the Apriori algorithm and the Apriori-TFP approach, are described in detail. Classification Rule Mining (CRM) is reviewed in section 2.5, and Classification Association Rule Mining (CARM), particularly the TFPC algorithm, adopted for evaluation purposes in this thesis, in section 2.6. The topic of text mining, as a particular case of data mining, is reviewed in section 2.7, where overall research in this field is summarised. Finally some well established approaches to Text Classification/Categorisation (TC) are presented in section 2.8.

2.2 Knowledge Discovery in Databases

2.2.1 The KDD Overview

Knowledge Discovery in Databases (KDD) [Han *et al.*, 1992] [Piatetsky-Shapiro and Frawley, 1991] [Piatetsky-Shapiro, 2000] has become a popular area of research and development in computer science during the last decade. The concept of *knowledge discovery* was first introduced by Frawley *et al.* [1991] — "*knowledge discovery is the nontrivial extraction of implicit, previously unknown,* *and potentially useful information from data*". The current interest in KDD is fuelled by:

1. The large amount of available data (for any particular application-domain considered, such as bioinformatics, e-commerce, financial investments, geography, marketing and sales, etc.) currently stored electronically. Anand *et al.* [1995] indicate that "*the amount of data being stored in databases has been on the increase since the 1970's partly due to the advances made in database technology since the introduction of the relational model for data by E. F. Codd*"; and
2. The hypothesis that there is likely to exist some hidden knowledge in the form of rules, patterns, regularities, customs, trends, etc. in a set of data, especially when the size of a data set becomes large.

KDD, with its goal of recognising patterns within large volumes of data is a tool with the potential to produce new unknown knowledge.

2.2.2 The KDD Process

KDD refers to the overall process of knowledge discovery. Piatetsky-Shapiro [2000] highlights the difference between KDD and data mining — "*sometimes 'knowledge discovery process' is used for describing the overall process, including all the data preparation and postprocessing while 'data mining' is used to refer to the step of applying the algorithm to the clean data*". The KDD process has been well studied and analysed [Brachman and Anand, 1996] [Fayyad *et al.*, 1996] [Han and Kamber, 2006] [Smith, 2005] [Weiss and Indurkhya, 1997]. It was pointed out in [Ahmed, 2004] that "*the problem of knowledge extraction from large databases involves many stages and a unique scheme has not yet been agreed upon*". One possible outline of the KDD process can be presented as follows.

1. **Problem Specification:** In the first stage of the KDD process, a domain-oriented understanding/specification of the target mining task/application is identified, which clarifies the goal of the application. An application-oriented description of three primitives for KDD was introduced in [Han *et al.*, 1992] — "*three primitives should be provided for the specification of a learning task:*

task-relevant data, background knowledge, and the expected representations of learning results". In the description of task-relevant data, factors such as the preferred size of the data collection, the expected number of data attributes, the required format of stored data, and the determined criteria of data qualification are identified. In the description of background knowledge, a concept hierarchy table and/or a concept tree may be drawn by knowledge engineers or domain experts. Concept hierarchies present the necessary taxonomy-like relationships among concept entities, where a concept entity can be a data attribute or any necessary information that relates to one or more data attributes. The description of the expected representation of learning results clarifies the representation form (e.g. relational form, first-order predicate calculus) for demonstrating the discovered knowledge. Another important aspect that may be involved in the application specification is the suitable criteria of knowledge interestingness measurement [Bayardo and Agrawal, 1999] [Freitas, 2006] [Klemettinen *et al.*, 1994] [Silberschatz and Tuzhillin, 1995]. This stage is often described as the "data mining" stage. However, in this thesis the term "data mining" is used to describe a particular "school" of information mining that concentrates on the mining of data typically presented in tabular form (see section 2.2.3 below for further detail). The term information mining is used here to imply a more generic level of mining.

2. **Resourcing:** The second stage of the KDD process aims to create a suitable set of data on which the target application can be performed. It may be possible to find several large databases available that appear to be task-relevant for the target application, and which were originally built for other purposes and are irrelevant to the target application. With regard to the information obtained from the problem specification stage, it may be known that not all data instances embraced in these large databases are qualified/useful to the target application. Thus extracting a suitable (sub) set of data (for the target application) from each of these available large databases is required.

3. **Data Cleaning:** The purpose of this stage is as Han and Kamber [2006] explain "*to remove noise and inconsistent data*" from a given dataset. De Veaux and Hand [2005] argue that "*anyone who has analyzed real data knows that the*

majority of their time on a data analysis project will be spent 'cleaning' the data before doing any analysis", and "*common wisdom puts the extent of this at 60-95% of the total project effort*". Klein [1998] argues that "*there is strong evidence that data stored in organizational databases have a significant number of errors*", and "*between one and ten percent of data items in critical organizational databases are estimated to be inaccurate*". In this stage, not only the noise and inconsistent data but also the missing and distorted data [De Veaux and Hand, 2005] and data outliers are cleansed. It will be helpful if an automated approach could be applied that is able to accurately detect and correct/remove error data in/from a given data set.

4. **Data Integration:** Definitions of data integration have been provided by Halevy [2001], Hull [1997], Lenzerini [2002] and Ullman [1997]. Basically, "*data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data*". In the previous stages of resourcing and data cleaning, a suitable data set was first extracted from each of the available large databases (sources), and then cleaned so that it can be further processed with other database techniques. In this stage, the cleaned data sets are combined into an integrated data set with a unified data view.

5. **Pre-processing:** In this stage two tasks are involved: (i) data transformation and (ii) data reduction. The data collected may be in an unstructured format, i.e. texts, images, videos, etc. In (i) the collected data is transformed into a structured/semi-structured (e.g. XML, SGML) representation that allows the data to be further operated upon in the KDD process. Then, for simplicity, especially when the volume of the collected data is considered too large, in (ii) the data that seems to be most significant for the target application is selected for further usages, and other data is discarded.

6. **Information Mining:** Information mining is the core stage in the overall KDD process. The purpose of this stage is to identify the most valuable information in the prepared data by utilising "*data analysis and knowledge discovery techniques under acceptable computational efficiency limitations, and produces a particular enumeration of patterns over the data*" [Zhang and Zhou, 2004].

7. **Interpretation and Evaluation of Results:** The validity of each pattern discovered is interpreted and measured. From this the overall quality of the mining performance can be evaluated. Freitas [2006] asks the question "*are we really discovering 'interesting' knowledge from data?*" In fact, not all mined knowledge is significantly interesting to the target application. In this stage, the discovered valuable knowledge is initially interpreted in a user-readable form (especially when the user is strongly involved in the evaluation), where the patterns, rule symbols, and/or variables are precisely and concisely expressed in human language. With respect to the criteria of knowledge interestingness being pre-determined in the problem specification stage, suitable patterns (valuable knowledge) are then caught in this stage.

It can be noted that the above stages are usually applied iteratively; with results of one stage providing feedback that allows improvement to earlier stages.

2.2.3 The KDD Schools

Corresponding to the variety of data formats, KDD research can be divided into different "schools", i.e. data mining, text mining, graph mining, image mining, web mining, music mining, etc.

- **Data Mining** [Bramer, 2007] [Cios *et al.*, 1998] [Dunham, 2002] [Han and Kamber, 2001] [Han and Kamber, 2006] [Hand *et al.*, 2001] [Sumathi and Sivanandam, 2006] [Thuraisingham, 1999] [Witten and Frank, 2005]**:** This school encompasses generic techniques, generally described in terms of database like data, although often adaptable to other forms of data. Some of the work described in this thesis makes use of techniques espoused by this school and are therefore presented in further detail in sections 2.3 ~ 2.6.

- **Text Mining** [Berry, 2004] [Feldman and Sanger, 2006] [Hotho *et al.*, 2005] [Weiss *et al.*, 2004]**:** This KDD school deals with various forms of electronic textual data. As text mining is central to the theme of this thesis a detailed review is provided in sections 2.7 and 2.8.

- **Graph Mining** [Chakrabarti and Faloutsos, 2006] [Cook and Holder, 2006] [Washio *et al.*, 2005]**:** This research school specialises on mining data represented in the form of graphs [Diestel, 2005]. Graph mining [Coenen, 2007] may be categorised into transaction graph mining, which searches for patterns in sets of graphs, or single graph mining, which looks for patterns within a single large graph.

- **Image Mining** [Hsu *et al.*, 2002]**:** Electronic images such as satellite images, medical images, and digital photographs are the data to be manipulated in this research school. Hsu *et al.* [2002] classify two types: (i) image mining that involves domain-specific applications; and (ii) image mining that involves general applications. Research topics include: satellite image (remote sensing) mining [Honda and Konoshi, 2000] [Ding *et al.*, 2002], medical image mining [Antonie *et al.*, 2001] [Pan *et al.*, 2005], image classification [Fan *et al.*, 2003] [Daniel and Ding, 2004], image clustering [Lin *et al.*, 2003] [Wang *et al.*, 2003b], image comparison [Olson, 2003], etc.

- **Web Mining** [Chakrabarti, 2002] [Chang *et al.*, 2006] [Liu, 2007] [Scime, 2005]**:** Web mining concentrates on detecting hidden knowledge from web like data. Three common types of web like data can be identified: web page contents, web hyperlink structures, and users' usage data (server logs). As a consequence, research areas in web mining can be grouped into three divisions/sections [Scime, 2005]: content mining [Lawrence and Giles, 1999] [Navigli, 2005], structure mining [Kosala and Blockeel, 2000] [Getoor and Diehl, 2005] [Hamdi, 2005], and usage mining [Baeza-Yates, 2005] [Baumgarten *et al.*, 1999]. Web content mining is closely related to text mining, since web pages usually contain a significant amount of text. Cooley *et al.* [1997] provide a comprehensive review of the web mining school.

- **Music Mining** [Lin *et al.*, 2004] [Pachet *et al.*, 2001] [Rolland and Ganascia, 2002]**:** Electronic music files such as MIDI, PCM and MP3 are the data required by this research school. One research aspect is music genre classification [Basili *et al.*, 2004] [Cataltepe *et al.*, 2007] [McKay and Fujinaga, 2004] [Scaringella *et al.*, 2006] — the automated assignment of "unseen" digital music records into pre-defined musical genres.

In this thesis, only data mining and text mining will be further considered.

2.3 The Data Mining School

Data mining is "*a multidisciplinary field, drawing work from areas including database technology, machine learning, statistics, pattern recognition, information retrieval, neural networks, knowledge-based system, artificial intelligence, high-performance computing, and data visualization*" [Han and Kamber, 2006]. In the past decade, data mining techniques have been widely applied in bioinformatics [Wang *et al.*, 2005], e-commerce [Raghavan, 2005], financial studies [Kovalerchun and Vityaev, 2000], geography [Miller and Han, 2001], marketing and sales studies [Berry and Linoff, 1997] [Rypielski *et al.*, 2002], etc.

The data mining research school refers to the classical investigation of KDD that takes only database like data as the input. Commonly known database models that are addressed include: relational database tables, transactional databases, etc. In this thesis, three related data mining approaches — Association Rule Mining (ARM), Classification Rule Mining (CRM), and Classification Association Rule Mining (CARM) — are presented in detail in the following sections. All three of these approaches are referenced later in this work.

2.4 Association Rule Mining

Association Rule Mining (ARM), first introduced in [Agrawal *et al.*, 1993], aims to extract a set of Association Rules (ARs) from a given transactional database D_T. An AR describes an implicative co-occurring relationship between two sets of binary-valued transactional database attributes (items), expressed in the form of an "⟨antecedent⟩ ⇒ ⟨consequent⟩" rule. In a marketing context, an archetypal AR can be exemplified as "⟨bread, egg, milk⟩ ⇒ ⟨butter, ham⟩" which can be interpreted as "when bread, egg and milk are purchased together, it is likely that both butter and ham are also purchased". Cornelis *et al.* [2006] suggest that the concept of mining ARs can be dated back to the work of Hájek *et al.* [1966].

More generally, we define ARM as follows. Let $I = \{a_1, a_2, \ldots, a_{n-1}, a_n\}$ be a set of items, and $Ŧ = \{T_1, T_2, \ldots, T_{m-1}, T_m\}$ be a set of transactions (data records), a transactional database D_T is described by $Ŧ$, where each $T_j \in Ŧ$ comprises a set of items $I' \subseteq I$. In ARM, two threshold values are usually used to determine the significance of an AR:

1. **Support:** A set of items S is called an itemset. The support of S is the proportion of transactions T in $\mathcal{T}$ for which $S \subseteq T$. If the support of S exceeds a user-supplied support threshold σ, S is defined to be a Frequent Itemset (FI).
2. **Confidence:** Represents how “strongly” an itemset X implies another itemset Y, where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. A confidence threshold α, supplied by the user, is used to distinguish high confidence ARs from low confidence ARs.

An AR $X \Rightarrow Y$ is said to be *valid* when the support for the co-occurrence of X and Y exceeds σ, and the confidence of this AR exceeds α. The computation of support is:

$$support(X \cup Y) = count(X \cup Y) \,/\, |\mathcal{T}| \,,$$

where $count(X \cup Y)$ is the number of transactions containing the set $X \cup Y$ in $\mathcal{T}$, and $|\mathcal{T}|$ is the size function (cardinality) of the set $\mathcal{T}$. The computation of confidence is:

$$confidence(X \Rightarrow Y) = support(X \cup Y) \,/\, support(X) \,.$$

Informally, “$X \Rightarrow Y$” can be interpreted as: if X is found in a transaction, it is likely that Y also will be found.

In general, ARM involves a search for all valid rules. The most computationally difficult part of this is the identification of FIs. A number of techniques for finding FIs are summarised below, including the TFP (Total From Partial) algorithm (section 2.4.5) on which the TFPC (Total From Partial Classification) algorithm used for evaluation purposes in this thesis is based. Brief mention is made of maximal frequent itemset mining and frequent closed itemset mining as these represent alternative techniques to that used in this thesis.

2.4.1 The Apriori Algorithm

Since its introduction in 1994, the Apriori algorithm developed by Agrawal and Srikant [1994] has been the basis of many subsequent ARM and/or ARM-related algorithms. In [Agrawal and Srikant, 1994], it was observed that ARs can be straightforwardly generated from a set of FIs. Thus, efficiently and effectively mining FIs from data is the key to ARM. The Apriori algorithm iteratively identifies FIs in data by employing the “closure property” of itemsets in the generation of candidate itemsets, where a candidate (possibly frequent) itemset is confirmed as frequent only when all its subsets are identified as frequent in the previous pass. The “closure property” of itemsets can be described as follows: if an itemset is frequent then all its subsets will also be frequent; conversely if an itemset is infrequent then all its supersets will also be infrequent. The Apriori algorithm is as follows.

Algorithm 2.1: The Apriori Algorithm

```
Input: (a) A transactional database $D_T$;
       (b) A support threshold $\sigma$;
Output: A set of frequent itemsets $S_{FI}$;
Begin Algorithm:
    $k \leftarrow 1$;
    $S_{FI} \leftarrow$ an empty set for holding the identified frequent itemsets;
    generate all candidate 1-itemsets from $D_T$;
    while (candidate $k$-itemsets exist) do
        determine support for candidate $k$-itemsets from $D_T$;
        add frequent $k$-itemsets into $S_{FI}$;
        remove all candidate $k$-itemsets that are not sufficiently supported
            to give frequent $k$-itemsets;
        generate candidate $(k + 1)$-itemsets from frequent $k$-itemsets using
            “closure property”;
        $k \leftarrow k + 1$;
    end while
    return ($S_{FI}$);
End Algorithm
```

Note: A k-itemset represents a set of k items.

Example 2.1: The Apriori Procedure

An example that illustrates the process of mining FIs by using the Apriori procedure is provided here. Let $I = \{A, B, C, D, E\}$ be a set of items existing in D_T, where D_T is described by a set of transactions $\mathcal{T} = \{T_1\{A, C, D\}, T_2\{B, C, E\}, T_3\{A, B, C, E\}, T_4\{B, E\}\}$. Assume the support threshold $\sigma = 2$ (or 50% = 2 / (4 records in $\mathcal{T}$)). First of all candidate 1-itemsets are enumerated — $\{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}\}$. The support of each candidate 1-itemset is counted in D_T — $\{\{A\}[2], \{B\}[3], \{C\}[3], \{D\}[1], \{E\}[3]\}$. The support of $\{D\}$ is less than σ, thus $\{D\}$ is infrequent and removed. Secondly, a set of candidate 2-itemsets is generated based on the frequent 1-itemsets — $\{\{AB\}, \{AC\}, \{AE\}, \{BC\}, \{BE\}, \{CE\}\}$. The support of each candidate 2-itemsets is counted in D_T — {{AB}[1], {AC}[2], {AE}[1], {BC}[2], {BE}[3], {CE}[2]}. Hence {AB} and {AE} are deleted. Consequently the candidate 3-itemsets are generated from the frequent 2-itemsets — {{ABC}, {ACE}, {BCE}} are first enumerated; according to the "closure property" of itemsets, only {BCE} is identified as a candidate 3-itemset (all its subsets are frequent). The support of {BCE} is counted in D_T — {{BCE}[2]}. Finally the Apriori procedure is terminated because there is no candidate 4-itemset. The mined set of FIs is returned as {{A}[2], {B}[3], {C}[3], {E}[3], {AC}[2], {BC}[2], {BE}[3], {CE}[2], {BCE}[2]}.

2.4.2 Related Algorithms

Other algorithms that use the Apriori style of operation include: AprioriTid and AprioriHybrid [Agrawal and Srikant, 1994], Partition [Savasere *et al.*, 1995], DHP (Direct Hashing and Pruning) [Park *et al.*, 1995], Sampling [Toivonen, 1996], DIC (Dynamic Itemset Counting) [Brin *et al.*, 1997], CARMA (Continuous Association Rule Mining Algorithm) [Hidber, 1999], etc. Some early algorithms where FIs are generated by enumerating candidate itemsets but do not apply the Apriori generate-prune iterative approach include: AIS (Agrawal·Imielinski·Swami) [Agrawal *et al.*, 1993], OCD (Off-line Candidate Determination) [Mannila *et al.*, 1994], SETM (SET oriented Mining) [Houtsma and Swami, 1995], etc.

A number of algorithms make use of a set enumeration tree structure [Rymon, 1992] to organise itemsets whose support is being counted. These include: [Ahmed *et al.*, 2003], [Coenen and Leng, 2001], [Coenen *et al.*, 2001], [Coenen and Leng, 2002], [Coenen *et al.*, 2004a], [Coenen *et al.*, 2004b], [El-Hajj and Zaïana, 2003], [Goulbourne *et al.*, 2000], [Han *et al.*, 2000], and [Liu *et al.*, 2002]. This approach is relevant to the present work, and will be discussed further below.

2.4.3 Maximal Frequent Itemsets

It is apparent that the size of a complete set of FIs can be very large. The concept of Maximal Frequent Itemsets (MFI) was proposed by Roberto and Bayardo [1998] to avoid the redundant work required identifying all FIs. This approach attempts to identify a set of MFIs of which all other FIs are subsets. The concept of vertical mining has also been effectively promoted in relation to MFI mining [Zaki *et al.*, 1997]. Vertical mining, first mentioned in [Holsheimer *et al.*, 1995], deals with a vertical transaction database D_{TV}, where each database record represents an item that is associated with a list of its relative transactions (the transactions in which it is present). MFI algorithms include: MaxEclat/Eclat [Zaki *et al.*, 1997], MaxClique/Clique [Zaki *et al.*, 1997], Max-Miner [Roberto and Bayardo, 1998], Pincer-Search [Lin and Kedem, 1998], MAFIA (MAximal Frequent Itemset Algorithm) [Burdick *et al.*, 2001], Genmax [Gouda and Zaki, 2001], etc.

2.4.4 Frequent Closed Itemsets

Algorithms belonging to this category extract ARs through generating a set of Frequent Closed Itemsets (FCIs) from D_T. The concept of FCI as explained in [Pei *et al.*, 2000] describes an itemset f that is frequent and $\neg\exists$ itemset $f' \supset f$ and f' shares a common support with f. The relationship between FI, MFI and FCI is that MFI $\subseteq$ FCI $\subseteq$ FI [Burdick *et al.*, 2001]. In this category algorithms include: CLOSET (mining CLOsed itemSETs) [Pei *et al.*, 2000], CLOSET+ [Wang *et al.*, 2003a], CHARM (Closed Association Rule Mining; the 'H' is gratuitous) [Zaki and Hsiao, 2002], MAFIA [Burdick *et al.*, 2001], etc.

2.4.5 The Apriori-TFP Approach

An ARM algorithm that is closely related to this thesis is Apriori-TFP [Coenen *et al.*, 2004a] [Coenen *et al.*, 2004b]. It is the precursor of the TFPC algorithm used for evaluating the work described here. In this subsection the Apriori-TFP approach is described in detail. Apriori-TFP makes use of a structure called a P-tree (Partial-support Tree) developed by Goulbourne *et al.* [2000], that stores partially calculated supports in a set enumeration tree structure and uses it to generate ARs. The P-tree is a summary (pre-processing) of input data, into a "compressed" form, with the inclusion of partial support counts. In Figure 2.1, a complete P-tree that consists of all the subsets of $I = \{A, B, C, D\}$ is drawn. The collection of transactions is given as $\mathcal{T} = \{T_1\{A\}, T_2\{B\}, T_3\{C\}, T_4\{D\}, T_5\{A, B\},$

$T_6\{A, C\}$, $T_7\{A, D\}$, $T_8\{B, C\}$, $T_9\{B, D\}$, $T_{10}\{C, D\}$, $T_{11}\{A, B, C\}$, $T_{12}\{A, B, D\}$, $T_{13}\{A, C, D\}$, $T_{14}\{B, C, D\}$, $T_{15}\{A, B, C, D\}\}$.

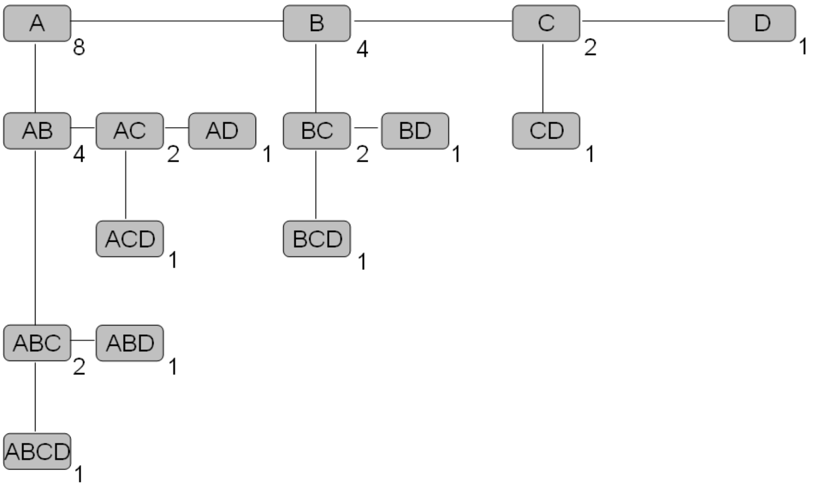


Figure 2.1: Complete P-tree for $I = \{A, B, C, D\}$

From Figure 2.1, it can be seen that the support stored at each tree node in the P-tree is "*an incomplete support total, comprised of the sum of the supports stored in the subtree of the node. Because of the way the tree is ordered, for each node in the tree, the contribution to the support count for that set which derives from all its lexicographically succeeding supersets has been included*" [Coenen *et al.*, 2004a]. In the construction of a P-tree, the given database D_T is scanned record by record. The algorithm of P-tree construction is outlined as follows.

Algorithm 2.2: The P-tree Construction
Input: A transactional database D_T;
Output: A P-tree *PT*;
Begin Algorithm:

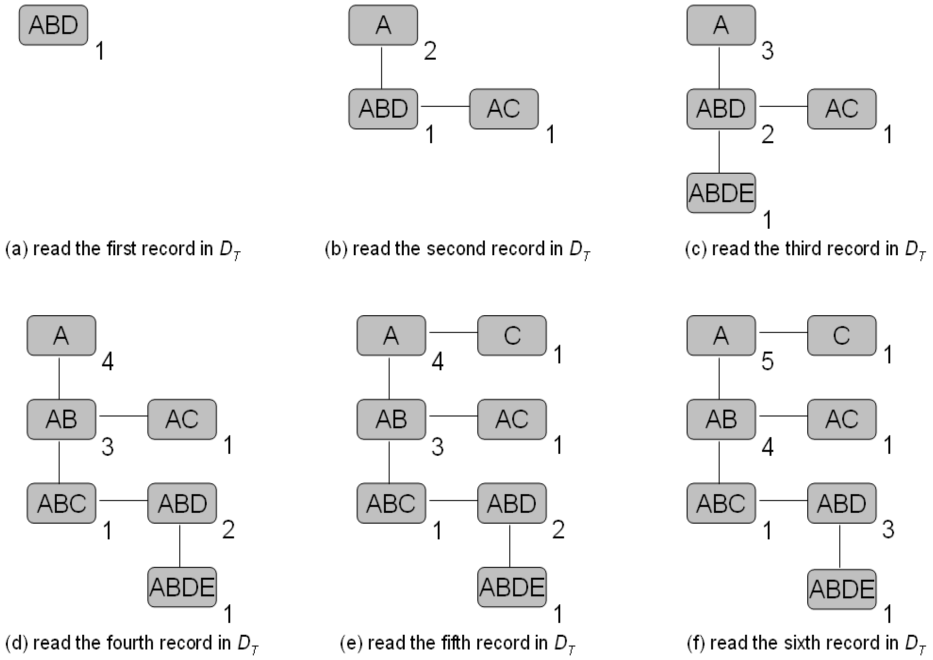
```
(1)   k ← 1;
(2)   PT ← an empty set enumeration tree structure;
(3)   while (the k-th record in D_T exists) do
(4)       traverse PT to find the position of this record;
(5)       if (this record does not exist in PT) then
(6)           create a new node in PT for this record;
(7)       increment the support count for all parent nodes of this new node
              traversed in the course to find the position;
(8)       k ← k + 1;
(9)   end while
(10)  return (PT);
```

End Algorithm

The P-tree generation algorithm calculates the interim support count Q_{β} for each tree node $\beta \in PT, \beta \subseteq I$. The computation of Q_{β} is defined as: $\sum$ (stored support count for β^{-}), where $\forall \beta^{-}, \beta^{-} \in PT, \beta^{-} \subseteq I, \beta^{-} \supseteq \beta$ and β^{-} follows β in lexicographic order. The total support count U_{β} for β can be calculated as: $Q_{\beta} + \sum$ (stored support count for β^{+}), where $\forall \beta^{+}$, $\beta^{+} \in PT$, $\beta^{+} \subseteq I$, $\beta^{+} \supset \beta$ and β^{+} precedes β in lexicographic order.

Example 2.2: The P-tree Generation

An example that illustrates the process of generating a P-tree from a given transactional database D_T is presented in Figure 2.2. Let $I = \{A, B, C, D, E\}$ and *Ŧ* = $\{T_1\{A, B, D\}, T_2\{A, C\}, T_3\{A, B, D, E\}, T_4\{A, B, C\}, T_5\{C\}, T_6\{A, B, D\}\}$. The P-tree generation begins by scanning the first record in D_T, which is $T_1\{A, B, D\}$. A P-tree node *ABD* is created for $T_1\{A, B, D\}$; its support count is initialised as 1 (Figure 2.2 (a)). Secondly $T_2\{A, C\}$ is read — a node *AC* is created with support count 1; since *AC* and the current node *ABD* have a “leading substring” *A*, a dummy node *A* is created where both *ABD* and *AC* are assigned to be the children of *A*; the support count of *A* is calculated as the sum of the supports of its children, which is 2 herein (Figure 2.2 (b)). Thirdly $T_3\{A, B, D, E\}$ is read — creating a node *ABDE* for this record with support 1; simply adding this node as a child of *ABD*; consequently incrementing each support count for *ABD* and *A* by 1 (Figure 2.2 (c)). Fourthly $T_4\{A, B, C\}$ is scanned — the node *ABC* is first created with support 1; the dummy node *AB* is created because it is a “leading string” of *ABC* and *ABD*; both *ABC* and *ABD* are assigned as the children of *AB*; the support count of *AB* is the sum of the supports of *ABC* and *ABD*, which is 3; the support of node *A* is incremented by 1; *ABDE* remains as a child of *ABD* (Figure 2.2 (d)). The fifth record $T_5\{C\}$ is then read — inserting the created node *C* with support 1 as a sibling of the current node *A* (Fig 2.2 (e)). Finally $T_6\{A, B, D\}$ is read — since the node *ABD* already exists in the P-tree, the supports of *ABD* and its parent nodes (*AB* and *A*) are simply incremented by 1 (Fig 2.2 (f)). The constructed P-tree can be further simplified from Figure 2.2 (f), which removes all the unnecessary items from each tree node (i.e. the items duplicated in a child of a parent node). The final form of the constructed P-tree is shown in Figure 2.3.



(a) read the first record in D_T (b) read the second record in D_T (c) read the third record in D_T

(d) read the fourth record in D_T (e) read the fifth record in D_T (f) read the sixth record in D_T

Figure 2.2: An example of P-tree generation

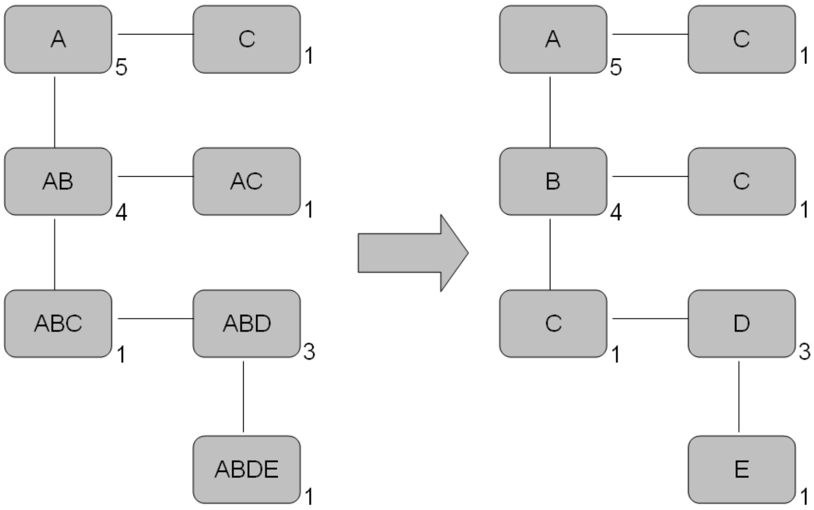


Figure 2.3: The final form of the constructed P-tree

Using the P-tree, the Apriori-TFP algorithm constructs a second set-enumeration tree, the T-tree (Total-support Tree), to contain the total support counts of the frequent itemsets [Coenen *et al.*, 2001]. In Figure 2.4, a complete T-tree that consists of all the subsets of $I = \{A, B, C, D\}$ is drawn. Note that the complete P-tree has already been shown in Figure 2.1. Algorithm 2.3 shows the process for calculating the total support count from partial support counts.

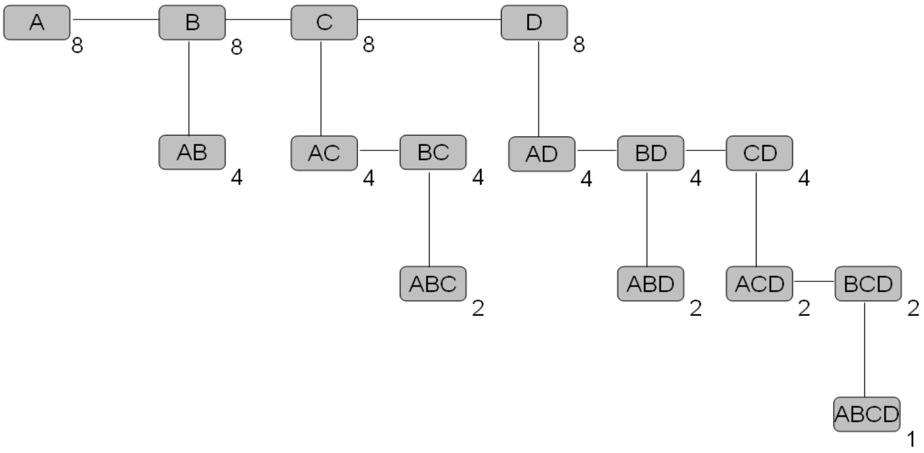


Figure 2.4: Complete T-tree for *I* = {*A*, *B*, *C*, *D*}

Algorithm 2.3: TFP — Compute Total from Partial Supports

```
Input: (a) A P-tree $PT$;
       (b) A T-tree $TT$ (without support values);
Output: $TT$ with total support associated;
Begin Algorithm:
    for each tree node $\beta \in PT$ do
        $\kappa \leftarrow \beta$ – parent($\beta$);
        $\kappa_f \leftarrow$ get the first attribute in $\kappa$;
        starting at node $\kappa_f$ of $TT$ do
        begin  if ($\kappa_f \subseteq \beta$) then
                    add the interim support of $\beta$ to the total support $U\kappa_f$;
                    if ($\kappa_f = \beta$) then
                        exit;
                    else
                        recurse to child node;
               proceed to sibling node;
        end begin
    end for
return ($TT$);
End Algorithm
```

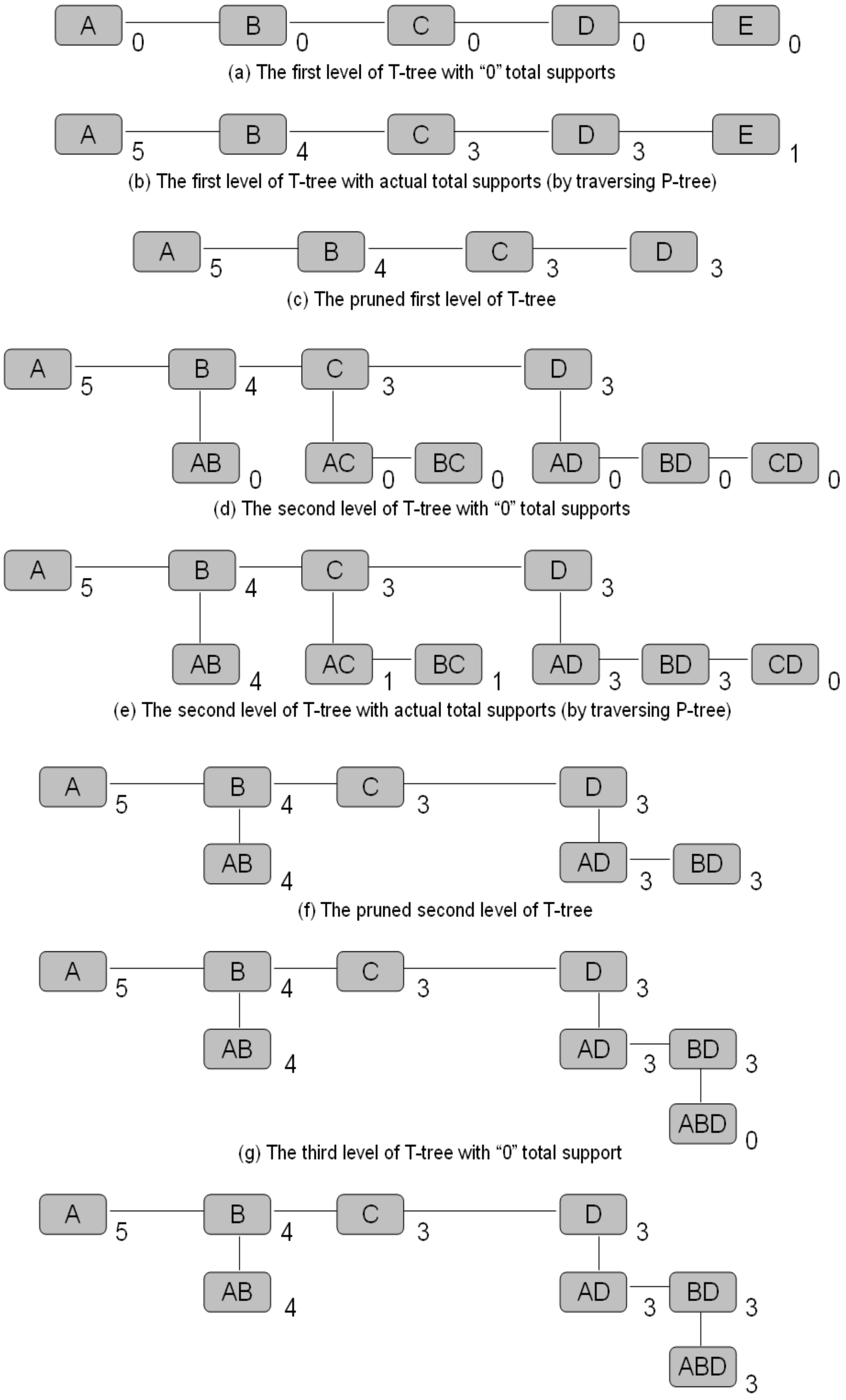
Coenen *et al.* [2001] note that: "*of course, to construct the entire T-tree would imply an exponential storage requirement. In (practice), however, it is only necessary to create that subset of the tree corresponding to the current candidate set being considered*". Thus the concept of Apriori can be applied to build a T-tree based on a P-tree — Apriori-TFP [Coenen *et al.*, 2001] [Coenen *et al.*, 2004b] (Algorithm 2.4).

Algorithm 2.4: Apriori-TFP

```
Input: (a) A P-tree PT;
       (b) A support threshold $\sigma$;
Output: A T-tree TT;
Begin Algorithm:
    $k \leftarrow 1$;
    build the level $k$ of TT;
    while (the level $k$ of TT exists) do
        traverse PT, applying algorithm TFP (Algorithm 2.3) to add interim
            supports of PT nodes to the level $k$ TT nodes generated
            previously;
        remove any level $k$ node in TT if its support < $\sigma$;
        $k \leftarrow k + 1$;
        build the level $k$ of TT;
    end while
    return (TT);
End Algorithm
```

Example 2.3: The T-tree Generation

An example, that illustrates the process of constructing a T-tree based on a given P-tree, is given as follows using the P-tree given in Figure 2.3. Assume the support threshold $\sigma = 3$ (or 50% = 3 / (6 records in $\mathcal{T}$)). A T-tree is generated level by level. The algorithm begins by listing the candidate 1-nodes with their total support counts initialised to 0 (Figure 2.5(a)); then the P-tree is traversed to add the interim support counts of the corresponding P-tree nodes to each candidate 1-node in the T-tree (Figure 2.5(b)). Any unsupported level 1 nodes are then pruned (Figure 2.5(c)). The construction of the first level of the T-tree is now complete. Next the second level of the T-tree is generated — the candidate 2-nodes are enumerated with their total support counts initialised to 0 (Figure 2.5(d)). The P-tree is again traversed to compute the total support for each candidate 2-node in the T-tree (Figure 2.5(e)).



(a) The first level of T-tree with "0" total supports

(b) The first level of T-tree with actual total supports (by traversing P-tree)

(c) The pruned first level of T-tree

(d) The second level of T-tree with "0" total supports

(e) The second level of T-tree with actual total supports (by traversing P-tree)

(f) The pruned second level of T-tree

(g) The third level of T-tree with "0" total support

(h) The entire T-tree generated from P-tree

Figure 2.5: An example of T-tree generation

The nodes *AC*, *BC* and *CD* are infrequent, so these three nodes are pruned from the T-tree (Figure 2.5(f)). Finally the third level of T-tree is built — the only candidate 3-node is *ABD*. The total support count for this candidate node is obtained from the P-tree (Figure 2.5(g)). This last node is frequent but no candidate 4-node can be enumerated, thus the T-tree construction process is complete (Figure 2.5(h)). The constructed T-tree can be further simplified based on the ideas of simplification presented for P-tree in Figure 2.3. In Figure 2.6 the final form of the built T-tree is presented.

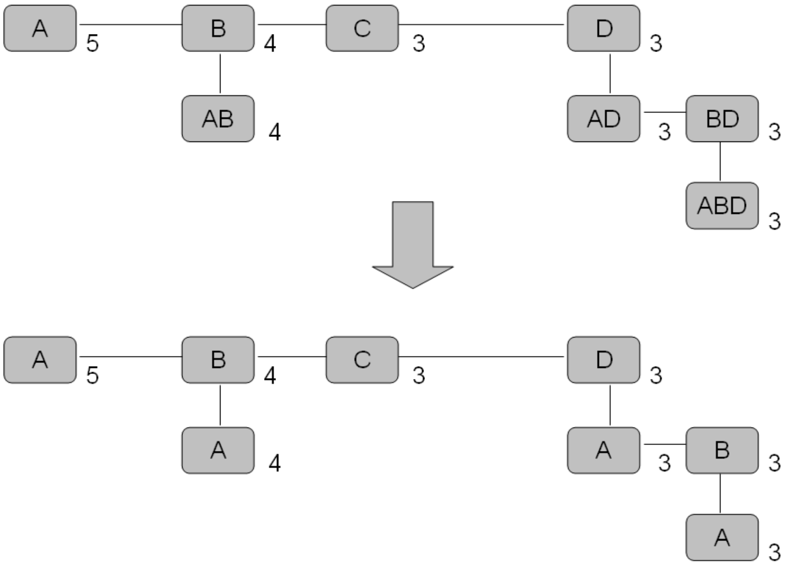


Figure 2.6: The final form of the constructed T-tree

In [Coenen and Leng, 2001] the technique of itemset ordering was introduced in the context of Apriori-TFP where it was found that tree ordering, by descending frequency of attributes, improves the performance.

2.5 Classification Rule Mining

Classification Rule Mining (CRM) [Quinlan, 1993] [Liu *et al.*, 1998] is a technique for identifying Classification Rules (CRs) from a given class database D_C, the objective being to build a classifier to categorise "unseen" data instances/records. Generally D_C is described by a relational database table that includes a class attribute — whose values are a set of pre-defined class labels $C = \{c_1, c_{2, \ldots,} c_{|C|-1,}$

$c_{|C|}$}. The process of CRM consists of two stages: (i) a training phase where CRs are generated from a set of training data instances $D_R \subset D_C$; and (ii) a test phase where "unseen" instances in a test data set $D_E \subset D_C$ are assigned into pre-defined class groups. A D_C is established as $D_R \cup D_E$, where $D_R \cap D_E = \varnothing$. Both D_R and D_E share the same database attributes except the class attribute. By convention the last attribute in each D_R record usually indicates the pre-defined class of this record, noted as the class attribute, while the class attribute is missing in D_E. In the following sub-sections a brief review is given of a number of CRM techniques. Note here that in this thesis the term "rule" is used in a generic way, to refer to any type of classification knowledge representation.

2.5.1 Classification Rule Mining Techniques

Mechanisms on which CRM algorithms have been based include: decision trees [Quinlan, 1993], naïve Bayes [Lowd and Domingos, 2005], *K*-Nearest Neighbour (*K*-NN) [James, 1985], Support Vector Machine (SVM) [Boser *et al.*, 1992], association rules [Liu *et al.*, 1998], genetic algorithm [Freitas, 2002] [Yang *et al.*, 2001], emerging patterns [Dong and Li, 1999] [Dong *et al.*, 1999], neural networks [Han and Kamber, 2001], case-based reasoning [Han and Kamber, 2001], rough sets [Han and Kamber, 2001], fuzzy set theory [Han and Kamber, 2001], etc. In this subsection, four of the most well-known mechanisms used in CR generation are briefly described.

- **Decision Tree Induction:** Where CRs are mined based on a greedy algorithm. The approach can be separated into two stages. In the first stage the tree is constructed from D_R and followed by a tree pruning phase. In the second stage the pruned tree is then used in CR generation. C4.5 [Quinlan, 1993] is the best known decision tree based CRM method and operates by recursively splitting D_R on the attribute that produces the *maximum information gain* to generate the decision tree. This tree is then pruned according to an error estimate. The result is used to classify "unseen" data.

- **Naïve Bayes:** The typical mechanism found in Bayesian CRM approaches such as [Domingos and Pazzani, 1997] is naïve Bayes [Lowd and Domingos, 2005], which has been widely applied in machine learning. The general idea of naïve

Bayes is to make use of knowledge of the probabilities involving attribute values and classes in the training dataset to produce a model of a machine learning application that can then be applied to “unseen” data. The term naïve is used to refer to the assumption that the conditional probability of a database attribute value given a class is independent of the conditional probability of other attribute values given that class. A naïve Bayes classifier [Rish, 2001] is built using D_R, and comprises a set of conditional probabilities for each database attribute and each class $c_i \in C$, so that there are $n \times |C|$ conditional probabilities, where n represents the number of attributes in D_R and $|C|$ is the size function (cardinality) of C. A naïve Bayes classifier also comprises a set of prior class probabilities, one for each class. All these probabilities are then used to classify “unseen” data records in D_E according to Bayes’ theorem.

- ***K*-Nearest Neighbour:** *K*-NN [James, 1985] is a well-known statistical approach used in CRM, and classifies an “unseen” data record $d'_{j'} \in D_E$, by assigning to that record the most frequent class in the set of the K most similar instances to $d'_{j'}$, identified in D_R. To identify the K most similar training-instances for $d'_{j'}$, calculation of the Euclidean distance value between each training data record $d_j \in D_R$ and $d'_{j'}$ is commonly used:

$$distance(d_j, d'_{j'}) = \sqrt{(\Sigma_{\{k = 1 \ldots n\}} (d_{j.k} - d'_{j'.k})^2)} ,$$

where $d_{j.k}$ and $d'_{j'.k}$ are the values of the k-th data attribute in D_C for d_j and $d'_{j'}$.

- **Support Vector Machine:** The objective of using SVM [Boser *et al.*, 1992] is to find a hypothesis $\hat{h}$ which minimises the *true error* defined as the probability that $\hat{h}$ produces an erroneous result. SVM makes use of linear functions of the form:

$$f(x) = w^T x + b ,$$

where w is the weight vector, x is the input vector, and $w^T x$ is the inner product between w and x. The main concept of SVM is to select a *hyperplane* that separates the positive and negative examples while maximising the smallest margin. Standard SVM techniques produce binary classifiers as opposed to multi-classifiers. Two common approaches to support the application of SVM techniques to the multi-class problem are One Against All (OAA) and One Against One (OAO).

2.5.2 Rule Pruning

In CRM a number of approaches have been proposed to prune the generated classification rule set. A popular example is the rule pruning idea presented in the Cover algorithm [Michalski, 1980] which takes D_R (the training data records) as its input and aims to generate a complete set of minimal non-redundant CRs. The Cover process can be explained in the following way: "*The covering set is found by heuristically searching for a single best rule that covers cases for only one class. Having found a best conjunctive rule for a class C, the rule is added to the rule set, and the cases satisfying it are removed from further consideration. The process is repeated until no cases remain to be covered*" [Apte *et al.*, 1994]. The original Cover algorithm was used for rule discovery. Herein, the rule pruning idea of this algorithm is focused. The Cover algorithm is provided below (Algorithm 2.5).

Algorithm 2.5: The Cover Algorithm

```
Input: (a) A training dataset $D_R$;
       (b) An ordered rule set $R$;
       Output: A complete set of minimal non-redundant classification rules $S_{CR}$;
Begin Algorithm:
(1)   $S_{CR} \leftarrow \emptyset$;
(2)   while ($D_R \neq \emptyset$) do
(3)       get first rule $r \in R$;
(4)       remove all records satisfied by $r$ from $D_R$;
(5)       add $r$ into $S_{CR}$;
(6)       remove $r$ from $R$;
(7)       if ($R = \emptyset$) then
(8)           break;
(9)   end while
(10)  return ($S_{CR}$);
End Algorithm
```

Note: In line (4), "satisfied by *r*" means having the same attribute values in both the antecedent ("IF part") and the consequent ("THEN part") of rule *r*.

2.6 Classification Association Rule Mining

An overlap between ARM and CRM is CARM (Classification Association Rule Mining), which strategically solves the traditional CRM problem by applying ARM techniques. The idea of CARM, first introduced in [Ali *et al.*, 1997], aims to extract a set of Classification Association Rules (CARs) from a class-transactional database $D_{C\text{-}T}$. Let D_T be a transactional database, and $C = \{c_1, c_2, \ldots, c_{|C|-1}, c_{|C|}\}$ be a set of pre-defined class labels (as previously defined in section 2.5), $D_{C\text{-}T}$ is described by $D_T \times C$. $D_{C\text{-}T}$ can also be defined as a special class database D_C (as previously defined in section 2.5), where all database attributes and the class attribute are valued in a binary manner — "*Boolean attributes can be considered a special case of categorical attributes*" [Srikant and Agrawal, 1996]. A CAR is a special AR that describes an implicative co-occurring relationship between a set of binary-valued data attributes and a pre-defined class, expressed in the form of an "$X \Rightarrow c_i$" rule, where X is an itemset found in D_T (as "$D_{C\text{-}T} - C$") and c_i is a pre-defined class in C.

As noted in section 1.2 CARM seems to offer a number of advantages over other CRM approaches ([Coenen *et al.*, 2005], [Shidara *et al.*, 2007], [Thabtah *et al.*, 2005], etc.). Coenen and Leng [2007] indicate:

- "*Training of the classifier is generally much faster using CARM techniques than other classification generation techniques such as decision tree (induction) and SVM approaches*" (particularly when handling multi-class problems as opposed to binary problems).
- "*Training sets with high dimensionality can be handled very effectively*".
- "*The resulting classifier is expressed as a set of rules which are easily understandable and simple to apply to unseen data (an advantage also shared by some other techniques, e.g. decision tree classifiers)*".
- In addition Liu *et al.* [1998] suggest that "*Experimental work has also shown that CARM can offer improved classification accuracy*".

2.6.1 CARM Approaches

Broadly speaking, CARM algorithms can be categorised into two groups according to the way that the CARs are generated:

- **Two Stage Algorithms** where a set of CARs are produced first (stage 1), which are then pruned and placed into a classifier (stage 2). Typical algorithms of this approach include CBA (Classification Based Associations) [Liu *et al.*, 1998] and CMAR (Classification based on Multiple Association Rules) [Li *et al.*, 2001]. CBA is an Apriori based CARM algorithm, which: (i) applies its CBA-RG (Rule Generator) procedure for CAR generation; and (ii) applies its CBA-CB (Classifier Builder) procedure to build a classifier based on the generated CARs. CMAR is similar to CBA but generates CARs through a FP-tree [Han *et al.*, 2000] based approach.

- **Integrated Algorithms** where the classifier is produced in a single processing step. Algorithms of this kind include TFPC (Total From Partial Classification) [Coenen and Leng, 2004] [Coenen *et al.*, 2005] [Coenen and Leng, 2007] and CPAR (Classification based on Predictive Association Rules) [Yin and Han, 2003]. TFPC is an Apriori-TFP based CARM algorithm that generates CARs through efficiently constructing both P-tree and T-tree set enumeration tree structures. CPAR is based on the PRM (Predictive Rule Mining) algorithm, and PRM is modified from the FOIL (First Order Inductive Learner) algorithm [Quinlan and Cameron-Jones, 1993].

2.6.2 Case Satisfaction and Rule Selection Mechanisms

Regardless of which particular CARM algorithm is used, a similar set of CARs is always generated from the data, and a classifier is usually presented as an ordered list of CARs. Coenen and Leng [2004] summarise three Case Satisfaction and Rule Selection (CSRS) mechanisms that have been employed in a variety of CARM algorithms for utilising the resulting classifier to classify "unseen" data records. These three CSRS mechanisms are itemised as follows (given a particular case):

- **Best First Rule:** Select the first rule that satisfies the given case according to some ordering imposed on the CAR list. The ordering can be defined according to many different ordering strategies including:

  1. CSA (Confidence Support & size-of-rule-Antecedent) where confidence is the most significant factor and size-of-rule-antecedent the least significant factor (used in CBA, TFPC and the early stage of processing of CMAR),

where size-of-rule-antecedent is measured by the cardinality of the rule antecedent and where the smaller this value is the better;

2. ACS (size-of-rule-Antecedent Confidence & Support), an alternative mechanism to CSA that considers size-of-rule-antecedent the most significant factor (the greater this value is the better) and support the least significant factor;
3. WRA (Weighted Relative Accuracy), which reflects a number of rule interestingness measures as proposed in [Lavrac *et al.*, 1999];
4. LAP (Laplace Accuracy) — as used in CPAR; and
5. χ^2 (Chi-square Testing) — as used, in part, in CMAR; etc.

These approaches are discussed further in section 2.6.3.

- **Best *K* Rules:** Select the first (top) *K* rules that satisfy the given case and then select a rule according to some averaging process as used for example, in CPAR. The term "best" in this case is defined according to an imposed ordering of the form described in Best First Rule.
- **All Rules:** Collect all rules in the classifier that satisfy the given case and then evaluate this collection to identify a class. One well-known evaluation method in this category is WCS (Weighted χ^2) testing as used in CMAR.

2.6.3 Rule Ordering Approaches

As noted above, rule ordering strategies support the Best First Rule CSRS mechanism. The rule ordering is conducted using some scoring mechanism. In some work related to that described here the nature of the scoring mechanisms can be divided into two groups. This work has been subsequently published in [Wang *et al.*, 2007b]. The first group includes the following:

- **CSA:** The CSA rule ordering strategy is based on the well-established "support-confidence" framework of for instance [Delgado *et al.*, 2002] that was originally introduced for AR interestingness measure. CSA sorts all generated CARs in a descending order based on the value of confidence of each CAR. For those CARs that share a common value of confidence, CSA sorts them in a descending order based on their support value. Furthermore for those CARs that share common values for both confidence and support, CSA sorts them in an ascending order based on the size of the rule antecedent.

- **ACS:** The ACS rule ordering strategy is a variant of CSA. It takes the size of the rule antecedent as its major factor (using a descending order — unlike the ascending order used in CSA) followed by the rule confidence and support values respectively. Coenen and Leng [2004] state that ACS ensures: "*specific rules have a higher precedence than more general rules*".

The second group of rule ordering strategies is rule weighting based where an additive weighting score is assigned to each CAR, based on a particular weighting scheme. Examples include:

- **WRA:** The WRA measure [Lavrac *et al.*, 1999] is used to determine the expected accuracy of each CAR. The calculation of the WRA score of a CAR R (as "$X \Rightarrow c_i$") confirmed in [Coenen and Leng, 2004], is:

  $$wra\_score(R) = support(X) \times (confidence(R) - support(c_i)) .$$

  WRA simply sorts all generated CARs in a descending order, based on the assigned WRA score of each CAR.

- **LAP:** The use of the *Laplace Expected Error Estimate* [Clark and Boswell, 1991] can be found in [Yin and Han, 2003]. The principle of applying this rule ordering mechanism is similar to WRA. The calculation of the LAP score of a CAR R is:

  $$lap\_score(R) = (support(X \cup \{c_i\}) + 1) / (support(X) + |C|) ,$$

  where $\{c_i\}$ denotes the 1-itemset form of c_i, and $|C|$ denotes the number of pre-defined classes.

- **χ^2:** χ^2 testing is a well-known technique used in statistics (see for example [Moore and McCabe, 1998]). It can be used to determine whether two variables are independent of one another. In χ^2 testing, a set of observed values O is compared against a set of expected values E — values that would be estimated if there was no dependence between the variables. The value of χ^2 is calculated using:

$$\chi^2\_value = \textstyle\sum_{\{j = 1 \ldots n\}} (O_j - E_j)^2 / E_j \ ,$$

where n is the number of entries in the confusion matrix, which is always 4 in CARM. If the χ^2 value between two variables (the rule antecedent and consequent-class of a CAR) is greater than a given threshold value (for CMAR the chosen threshold is 3.8415), it can be concluded that there is a dependence between the rule antecedent and consequent-class; otherwise there is no dependence. After assigning a χ^2 score/value to each CAR, it can be used as the basis for ordering CARs into descending order.

Yin and Han [2003] suggest that there are only a limited number (perhaps 5 in each class) of CARs that are required to distinguish between classes and should thus be used to make up a classifier. Yin and Han employ LAP to estimate the accuracy of CARs. Incorporating the *K* rules concept of Yin and Han a hybrid support-confidence & rule weighting based ordering approach was developed as part of the

```
Algorithm 2.6: The Hybrid Rule Ordering Procedure
Input: (a) A list of CARs $\Re$ (either in CSA or ACS ordering manner);
       (b) A desired number (integer value) $K$ of the best rules;
Output: A re-ordered list of CARs $\Re^{HYBRID}$ (in a hybrid rule ordering manner);
Begin Algorithm:
    $\Re^{HYBRID} \leftarrow \emptyset$;
    $\Re^{SCORE} \leftarrow \emptyset$;
    for each CAR $\in \Re$ do
        calculate the additive score ($\delta$) for this CAR (in WRA, LAP or $\chi^2$
            manner);
        add (CAR $\oplus\ \delta$) into $\Re^{SCORE}$; // the $\oplus$ sign means "with" an
            additive CAR attribute
    end for
    sort $\Re^{SCORE}$ in a descending order based on $\delta$;
    $\Re^{SCORE} \leftarrow$ select the top $K$ CARs (for each pre-defined class) $\in \Re^{SCORE}$;
    sort $\Re^{SCORE}$ either in CSA or ACS ordering manner; // keep $\Re^{SCORE}$
        consistent with $\Re$
    $\Re^{HYBRID} \leftarrow$ link $\Re^{SCORE}$ at front of $\Re$;
    return ($\Re^{HYBRID}$);
End Algorithm
```

research described here (and published in [Wang *et al.*, 2007b]). The hybrid approach fuses both the CSRS mechanisms of Best First Rule and Best *K* Rules. The overall procedure of the hybrid rule ordering strategy is outlined (see Algorithm 2.6).

From the foregoing, six hybrid rule ordering schemes can be identified (see [Wang *et al.*, 2007b] for further details):

- **Hybrid CSA/WRA:** Selects the Best *K* Rules (for each pre-defined class) in a WRA manner, and re-orders both the best *K* CAR list and the original CAR list in a CSA fashion. The best *K* CAR list is linked at the front of the original CAR list.

- **Hybrid CSA/LAP:** Selects the Best *K* Rules (for each pre-defined class) in a LAP manner, and re-orders both the best *K* CAR list and the original CAR list in a CSA fashion. The best *K* CAR list is linked at the front of the original CAR list.

- **Hybrid CSA/χ^2:** Selects the Best *K* Rules (for each pre-defined class) in a χ^2 manner, and re-orders both the best *K* CAR list and the original CAR list in a CSA fashion. The best *K* CAR list is linked at the front of the original CAR list.

- **Hybrid ACS/WRA:** Selects the Best *K* Rules (for each pre-defined class) in a WRA manner, and re-orders both the best *K* CAR list and the original CAR list in an ACS fashion. The best *K* CAR list is linked at the front of the original CAR list.

- **Hybrid ACS/LAP:** Selects the Best *K* Rules (for each pre-defined class) in a LAP manner, and re-orders both the best *K* CAR list and the original CAR list in an ACS fashion. The best *K* CAR list is linked at the front of the original CAR list.

- **Hybrid ACS/χ^2:** Selects the Best *K* Rules (for each pre-defined class) in a χ^2 manner, and re-orders both the best *K* CAR list and the original CAR list in an ACS fashion. The best *K* CAR list is linked at the front of the original CAR list.

2.6.4 The TFPC Algorithm

The TFPC algorithm is the CARM technique that will be used as the basis for experimental work undertaken in this thesis. In this subsection a detailed description of TFPC is therefore provided.

Several of the above CARM methods (see section 2.6.1) apply coverage analysis (see section 2.5.2 — the Cover algorithm) to prune instances/cases and reduce the number of rules generated in the training phase. It can be demonstrated that coverage analysis, especially when applied to a large $D_{C\text{-}T}$ comprising many items and multiple transactions, includes a significant computational overhead. This is the motivation behind development of an algorithm that directly builds an acceptably accurate classifier without coverage analysis. The TFPC algorithm is directed at this aim. Coenen and Leng [2007] argue that the principal advantage offered by TFPC is that "*it is extremely efficient (because it dispenses with the need for coverage analysis)*".

TFPC is derived from the Apriori-TFP ARM approach (see section 2.4.5). It employs the same structures and procedures as used in Apriori-TFP to the task of identifying CARs in $D_{C\text{-}T}$. For this purpose, pre-defined class labels in $D_{C\text{-}T}$ are considered as items, and set at the end of the item list (ordered in a descending manner based on the item frequency).

In its rule generation process, TFPC adopts the heuristic: "*if we can identify a rule $X \Rightarrow c$ which meets the required support and confidence thresholds, then it is not necessary to look for other rules whose antecedent is a superset of X and whose consequent is c*" [Coenen *et al.*, 2005]. The advantages of employing this heuristic can be listed as follows.

- It "*reduces the number of candidate rules to be considered*" thus "*significantly improving the speed of the rule-generation algorithm*" [Coenen and Leng, 2007].

- It reduces the number of final rules to be generated, so that "*this 'on-the-fly' pruning replaces the expensive pruning step that other algorithms perform by coverage analysis*" [Coenen and Leng, 2007].

- It reduces the risk of overfitting — i.e. the risk of producing a set of rules that perform well on the training data set but do not generalise well to the test data set.

The classifier built by TFPC is finally represented as a list of CARs in a CSA rule ordering fashion (although any other rule ordering strategy may equally be applied — see section 2.6.3). When classifying "unseen" cases TFPC typically uses the Best First Rule CSRS approach (but again any other CSRS mechanism may be employed — see section 2.6.2). The TFPC rule generation algorithm is presented below.

Algorithm 2.7: The TFPC Algorithm

```
Input: (a) A class-transactional database based training data set $D_{C\text{-}TR}$;
       (b) A support threshold $\sigma$;
       (c) A confidence threshold $\alpha$;
Output: A set of class association rules $S_{CAR}$;
Begin Algorithm:
    $D_{C\text{-}TR} \leftarrow$ remove unsupported attributes from $D_{C\text{-}TR}$;
    $D_{C\text{-}TR} \leftarrow$ recast records in $D_{C\text{-}TR}$ so that remaining attributes (except the
            class attribute) are ordered according to frequency;
    create a P-tree $PT$ based on $D_{C\text{-}TR}$; // apply Algorithm 2.2
    create the first level (1-itemsets) of a T-tree $TT$ and count supports by
            traversing $PT$;
    generate the second level (candidate 2-itemsts) of $TT$;
    $S_{CAR} \leftarrow \emptyset$;
    $k \leftarrow 2$;
    while (the level $k$ of $TT$ exists) do
        traverse $PT$, applying algorithm TFP (Algorithm 2.3) to add interim
                supports of $PT$ nodes to the level $k$ $TT$ nodes generated previously;
        remove any level $k$ node in $TT$ if its support < $\sigma$;
        for all remaining level $k$ nodes in branches representing a class do
            generate a CAR $R$ with associated confidence value;
            if (the confidence value ≥ $\alpha$) then
                add $R$ into $S_{CAR}$; // in a CSA rule ordering fashion
                remove corresponding level $k$ nodes from $TT$;
        end for
        $k \leftarrow k + 1$;
        build the level $k$ of $TT$ from the remaining $(k – 1)$-itemsets using the
                closure property of itemsets;
    end while
    return ($S_{CAR}$);
End Algorithm
```

Example 2.4: The TFPC Rule Generation Procedure

An example that illustrates the process of generating CARs using the TFPC CARM approach is as follows. Let {A, B, D, E} be the attribute-set of a given class-transactional database, where each data record comprises some subset of this attribute-set that is labelled with a pre-defined class label, either c_1 or c_2. The training data set of the given database can be assumed as: {{A, B, D, E, c_1}, {A, B, D, c_1}, {A, B, D, c_2}, {A, B, E, c_1}, {A, B, E, c_2}, {A, B, E, c_2}, {A, B, c_1}, {A, B, c_2}, {A, B, c_1}, {A, B, c_2}, {A, D, E, c_1}, {A, D, c_1}, {A, D, c_1}, {A, D, c_1}, {A, D, c_1}, {B, D, E, c_2}, {B, D, E, c_2}, {B, D, E, c_2}, {B, E, c_2}, {D, E, c_2}}. In the TFPC rule generation process, a P-tree is first constructed from this data (Figure 2.7).

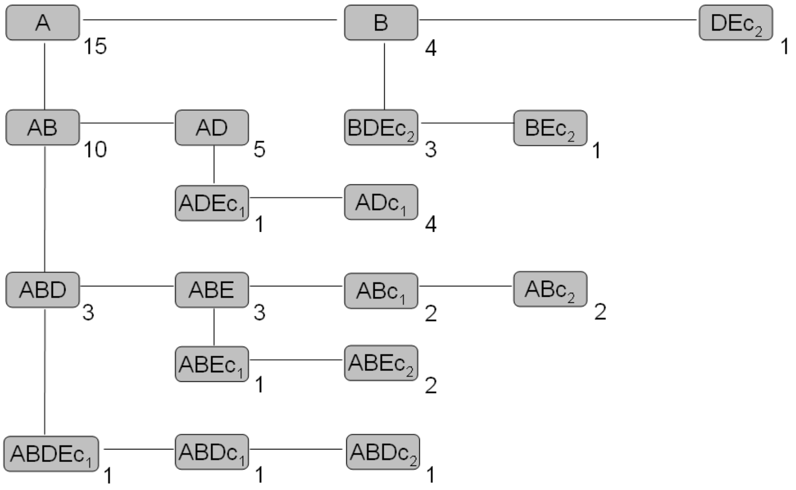


Figure 2.7: The P-tree structure in TFPC

Assuming a support threshold of $\sigma = 6$ records (or 30%) and a confidence threshold of $\alpha = 70\%$, a T-tree is then built level by level, based on the P-tree, as follows:

- **Level 1:** The candidate 1-nodes are {A}, {B}, {D}, {E}, {c_1}, and {c_2}. The P-tree is traversed to add the interim support counts of the corresponding P-tree nodes to each candidate 1-node in the T-tree, e.g. the total support of node B is 14, the sum of the partial supports of node B (4) plus node AB (10); since the total support of each candidate 1-node is greater than σ, no nodes are pruned at this stage. The construction of the first level of the T-tree is complete with {{A}[15], {B}[14], {D}[12], {E}[10], {c_1}[10], {c_2}[10]}.

- **Level 2:** The candidate 2-nodes are enumerated as $\{A, B\}$, $\{A, D\}$, $\{B, D\}$, $\{A, E\}$, $\{B, E\}$, $\{D, E\}$, $\{ A, c_1\}$, $\{B, c_1\}$, $\{D, c_1\}$, $\{E, c_1\}$, $\{ A, c_2\}$, $\{B, c_2\}$, $\{D, c_2\}$, and $\{E, c_2\}$. The P-tree is traversed to compute the total support for each candidate 2-node in the T-tree. Consequently nodes AE, Bc_1, Ec_1, Ac_2, Dc_2 are infrequent, hence they are removed from the T-tree; for nodes Ac_1, Dc_1, Bc_2, and Ec_2, their rule confidence value is calculated — Ac_1[66.67%], Dc_1[58.33%], Bc_2[64.29%], and Ec_2[70%] — where the confidence of $Ec_2 \geq \alpha$, therefore the CAR "$E \Rightarrow c_2$" is added to the rule set, and no further candidate super sets are generated from this node (shown by the dashed line in Figure 2.8). The construction of the second level of T-tree is now complete with $\{\{A, B\}[10], \{A, D\}[8], \{B, D\}[6], \{B, E\}[8], \{D, E\}[6], \{A, c_1\}[10], \{D, c_1\}[7], \{B, c_2\}[9]\}$.

- **Level 3:** The candidate 3-nodes are $\{A, B, D\}$, $\{B, D, E\}$ and $\{A, D, c_1\}$. The total support values are then obtained from the P-tree. Nodes ABD and BDE are infrequent and thus removed from the T-tree. The confidence of node ADc_1 is calculated as 7 / 8 = 87.5%, which is greater than α, thus this CAR "$AD \Rightarrow c_1$" is added in the rule set (shown by the dashed line in Figure 2.8). The entire generation process is now complete (Figure 2.8).

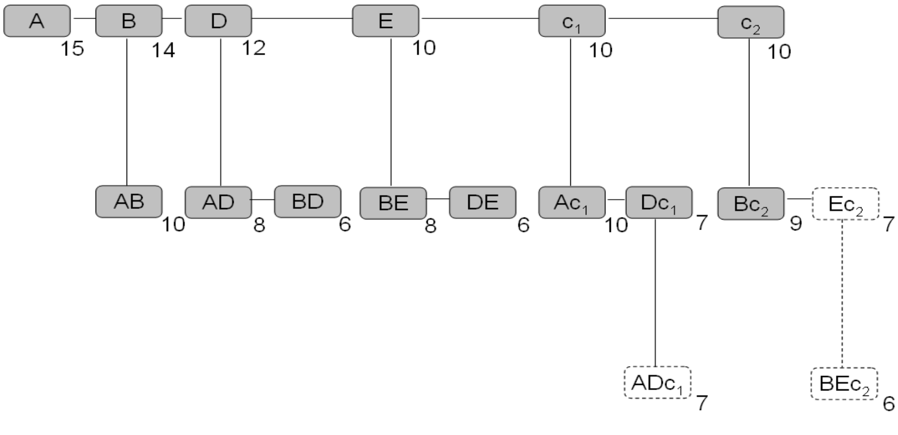


Figure 2.8: The T-tree structure in TFPC

The generated rule set is returned with two CARs: "$E \Rightarrow c_2$" and "$AD \Rightarrow c_1$". Using the CSA rule ordering mechanism, "$AD \Rightarrow c_1$" (confidence = 87.5%) is before "$E \Rightarrow c_2$" (confidence = 70%), in the CAR list that represents the TFPC classifier.

2.7 Text Mining

Text mining — an increasingly important field of research in KDD — applies data mining techniques to textual data collections, and "*aims at disclosing the concealed information by means of methods which on the one hand are able to cope with the large number of words and structures in natural language and on the other hand allow to handle vagueness, uncertainty and fuzziness*" [Hotho *et al.*, 2005]. Typical textual data includes natural language speeches (e.g. dialogues, argumentations), text files (e.g. magazine articles, academic papers), web documents (e.g. web news, e-mails), etc. In text mining, a given textual data collection is commonly refined in documentbase fashion. A documentbase (i.e. a set of electronic documents) usually consists of thousands of documents, where each document may contain hundreds of words.

Broadly speaking, research in text mining can be specified in a number of task-driven areas that include: text classification, document clustering, topic detection and tracking, text segmentation, text summarisation, text visualisation, etc.

- **Text Classification:** Text Classification (TC) [Liu *et al.*, 2004] [Zhuang *et al.*, 2005], also known as Text Categorisation (TC) [Sebastiani, 2002] [Sebastiani, 2005] [Yang and Liu, 1999], is an application of CRM approaches to textual data, and aims to automatically assign "unseen" documents into pre-defined categories (text class-labels). In a machine learning context, TC is recognised as a supervised learning approach that involves a training phase and a test phase [Namburu *et al.*, 2005]. TC is central to the theme of this thesis and is therefore further discussed in section 2.8.

- **Document Clustering:** Document clustering [Dhillon *et al.*, 2004] is concerned with automatically grouping similar documents into text categories without a supervised learning procedure, where (1) the number of categories may be specified or not, and (2) document similarity is about some function on documents, founded on some "distance based" measure and/or some "probability density function". Steinbach *et al.* [2000] indicate that two common document clustering techniques are agglomerative hierarchical clustering and *K*-means.

- **Topic Detection and Tracking:** Topic Detection and Tracking (TDT) refers to "*tasks on analyzing time-ordered information sources, e.g. news wires*" [Rajaraman and Tan, 2001]. The initial motivation of this research was to develop an information system that in the words of Allan [2002] would "*monitor broadcast news and alert an analyst to new and interesting events happening in the world*". TDT comprises two tasks: (i) topic detection — detecting text topics that are previously unknown to the system, where a text topic is "*an abstraction of a cluster of stories that discuss the same event*" [Rajaraman and Tan, 2001]; and (ii) tracking — assigning incoming stories into previously detected text topics. Walls *et al.* [1999] argue that TDT is an extended problem of document clustering, i.e. incremental document clustering [Hammouda and Kamel, 2003].

- **Text Segmentation:** Text segmentation concerns the automatic partitioning of text into coherent segments. It constructs according to Beeferman *et al.* [1999] "*a system which, when given a stream of text, identifies locations where the topic changes*". In the past decade, many studies have indicated that automatically defining similarity between words (i.e. lexical cohesion [Kozima, 1993] [Kozima and Furugori, 1993]) can be very useful in text segmentation investigation [Bestgen, 2006].

- **Text Summarisation:** Text summarisation [Mani and Maybury, 1999] aims to produce a core tool/system that automatically generates a shortened version of some text. Research in this area can be dated back to the late 1950's and early 1960's [Hovy, 2003]. Gagnon and Sylva [2005] categorise text summarisation investigation under three headings:

  1. **Extraction** — where "interesting" sentences in the source text are simply gathered to "*produce what is hoped to be a legible summary*";

  2. **Abstraction** — Mani [2001] indicates that at least some abstract material is not present in the source text, and Gagnon and Sylva suggest that

abstraction may "*start by reducing sentences from the source text, joining sentence fragments, generalizing, etc.*"; and

3. **Text Reduction** — where the size of the source text is reduced by sentence reduction, usually based on a syntactic analysis.

In a machine learning context, Nomoto and Matsumoto [2001] divide text summarisation into two kinds: (i) supervised approaches that "*typically make use of human-made summaries or extracts to find features or parameters of summarization algorithms*"; and (ii) unsupervised approaches that "*determine relevant parameters without regard to human-made summaries*".

- **Text Visualisation:** Text visualisation [Hotho *et al.*, 2005] aims to provide a "*more comprehensive and better and faster understandable information than it is possible by (using) pure text based descriptions*" and which improves and simplifies "*the discovery or extraction of relevant patterns or information*" from textual data. Two major directions under this research are (i) document collection visualisation — where the entire documentbase is targeted to be visualised, usually in a two-dimensional (sometimes in a three-dimensional) projection; and (ii) relation/result visualisation — i.e. visualising keyword-document relations [Hearst and Karadi, 1997], the results of a set of queries [Havre *et al.*, 2001], etc.

This thesis focuses on TC only, other text mining applications will not be considered further.

2.8 Text Classification

Text Classification/Categorisation (TC) according to Fragoudis *et al.* [2005] is "*the task of assigning one or more predefined categories to natural language text documents, based on their contents*". Early studies of TC can be dated back to the early 1960s (see for instance [Maron, 1961]). During the past half-century, a large number (hundreds or maybe thousands) of computer science related publications (research papers and/or academic books) have been concerned themselves with the investigation of this topic. Recent books concerned with TC include [Berry, 2004] and [Berry and Castellanos, 2008].

2.8.1 Various Tasks and Evaluation Measures

In a general context, the TC problem can be separated into two significant divisions:

1. Assigning only one pre-defined category to each ("unseen") natural language text document as in [Cardoso-Cachopo, 2007] and often defined as the non-overlapping or *single-label* TC task; and

2. Assigning more than one pre-defined category to an "unseen" document as in [Feng *et al.*, 2005] and often defined as the overlapping or *multi-label* TC task.

"*A special case of single-label TC is binary TC*" [Sebastiani, 2002], which in particular assigns either a pre-defined category or the complement of this category to an "unseen" document. Thus binary TC has been referred to as the *two-class* (positive and negative) TC approach. In previous TC investigation, many studies have addressed this approach, i.e. [Joachims, 1998], [Sebastiani, 2002], [Wu *et al.*, 2002], etc. In contrast, single-label TC tasks other than the two-class approach are recognised as *multi-class* approaches, and simultaneously deal with all given categories and assign the most appropriate category to each document. Individual studies under this heading include [Berger and Merkl, 2004], [Giorgetti and Sebastani, 2003], and [Wu *et al.*, 2007]. When handling a documentbase (prepared textual data) with more than two pre-defined categories, a sufficient set of binary TC tasks will implement a multi-class TC task with a possibly better accuracy of classification, but a drawback in terms of processing efficiency. It is worth giving further consideration to the following: when dealing with a multi-label TC problem, simply interpreting and solving such problems using a sufficient set of binary TC tasks may result in good performance with respect to both the accuracy of classification and the efficiency of computation.

To evaluate the performance of a TC system (a text classifier and/or its related techniques), classification accuracy has been widely used. Agarwal *et al.* [2007] confirm that "*in a classification problem, the classification system is trained on the training data and effectiveness is measured by accuracy on test data which is the fraction of correctly predicted document-class mappings*". Other measures [Zheng and Srihari, 2003] that have been used in TC evaluation (especially in binary TC) include: precision, recall, the F1 measure, micro-averaging, macro-averaging, etc. Sebastiani [2005] clarifies the reason for applying these evaluation measures rather than using accuracy alone — "*in binary TC applications the two categories c_i and $\overline{c_i}$ are usually unbalanced, i.e. one contains far more members than the other*" — therefore "*building a classifier that has high accuracy is trivial*" (by predicting the majority class in the training set for all test instances). On the other hand, Cardoso-Cachopo and Oliveira [2006] point out that "*measures based on Precision and Recall, like F1 or PRBP (Precision-Recall Breakeven Point) have been widely used to compare the performance of TC models*"; "*however, to evaluate single-label TC tasks, these measures are not adequate*"; "*so, accuracy, the percentage of correctly classified documents, is used to evaluate this kind of tasks*". Many single-label TC studies, such as that of [Berger and Merkl, 2004], especially when handling the multi-class TC problem, support Cardoso-Cachopo and Oliveira's view point.

In this thesis, the particular problem to be focused on is the single-label multi-class TC task; therefore the classification accuracy measure was adopted in the evaluation of the proposed TC related approaches.

2.8.2 TC Framework and the Usage of CARM

Mladenic [1999] divides the overall TC process into two stages: (i) pre-processing of textual data and (ii) mining of classification rules (or classification). Stage (i) comprises document (base) representation and feature selection. In stage (ii) various CRM techniques can be equally well applied. For each individual technique, it can be demonstrated that under a variety of circumstances one particular technique will become more accurate than its alternatives. With respect to the variety of CRM techniques available, classification approaches based on mining association rules (i.e. CARM) have been proposed for application in TC (e.g. [Antonie and Zaïane, 2002] [Baralis and Garza, 2006] [Yoon and Lee, 2005]). Apte *et al.* [1994] indicate that the pre-processing (initial task) stage of TC is to "*produce a list of attributes from samples of text of labelled documents*", and represent sample cases "*in terms of the words or phrases found in the documents*" where "*each case consists of the values of the attributes for a single article, where the values could be either Boolean, i.e., indicating whether the attribute appears in* *the text or does not, or numerical, i.e., frequency of occurrence in the text being processed*". Note that the Boolean valued textual data representation directly accounts for a class-transactional database (see section 2.6) where CARs (Classification Association Rules) can be mined. This suggests that CARM should be utilised in TC. Other points that recommend using CARM in TC are listed as follows.

- The volume of a textual data collection is usually large. As Antonie and Zaïane [2002] argue, a CARM based text classifier "*is fast during both training and categorisation phases*", especially when handling very large databases.

- A CARM based text classifier as Antonie and Zaïane [2002] argue "*can be read, understood and modified by humans*".

- A CARM based text classifier is relatively insensitive to noise data (for example some document words may be misspelt/miswritten) in both the training and categorisation phases. In the training phase, significant classification rules are identified based on sufficiently large values of the rule's support and confidence. When dealing with a large database, a relatively small amount of noise data in the training dataset should not significantly affect the expected classification result. In the categorisation phase, a generated classifier is presented as an ordered list of CARs. When classifying an "unseen" document, the Best First Rule CSRS (Case Satisfaction and Rule Selection) mechanism is usually employed.

Given the above suggested advantages offered by CARM with respect to TC, this approach was adopted in this thesis to support the investigation of language-independent text mining, as described below.

2.8.3 Language-independent Text Classification

One direction in which the traditional TC approach can be extended is to develop a language-independent text classifier that can be globally applied to all languages (e.g. English, Arabic, Chinese, Spanish, etc.). Previous TC techniques operate well with quite high classification accuracy, but most of them have been designed with particular languages and styles of language as the target. In the past decade, only a very few TC studies have addressed this issue (see for example [Damashek, 1995], [Peng *et al.*, 2003]). It is clear that the key in deriving a language-independent text classifier is provided through the language-independent documentbase pre-processing approach.

In previous studies, the $\mathcal{N}$-gram approach (as further detailed in section 3.3.2) has been suggested. However the experimental results of [Peng *et al.*, 2003] indicate that: for different languages (Greek, English, Chinese, and Japanese) and/or tasks (authorship attribution, text genre classification, and topic detection), it appears that different values for $\mathcal{N}$ should be set. This suggests that the $\mathcal{N}$-gram approach is not well suited to language-independent TC. The work described in this thesis thus avoids the usage of the $\mathcal{N}$-gram approach.

In this thesis a number of statistical language-independent documentbase pre-processing techniques are described that apply statistical methods to identify key terms (words/phrases that significantly serve to distinguish between classes) in the documentbase. The proposed documentbase pre-processing techniques in turn establish the approach of language-independent single-label multi-class TC with acceptable performance in terms of both classification accuracy and processing efficiency.

Chapter 3

Documentbase Pre-processing

3.1 Introduction

Natural language documents are comprised of words, punctuation marks and a variety of other symbols such as numbers etc. The words in a document, in turn, are made up of one or more letters (as in for example European languages) or symbols (as in for example Chinese). In either case, a text mining application requires the documentbase to be pre-processed so that it is in an "appropriate format" [Tan, 1999] for the mining purpose. In [Mladenic, 1999] the nature of documentbase pre-processing is characterised as: (i) documentbase representation — that designs an application oriented data structure that precisely interprets a given documentbase in an explicit and structured manner; and (ii) feature selection — that identifies the most significant text-units (text-features) in the documentbase, based on (i).

In this chapter, aspects of documentbase pre-processing, especially for the TC problem, are explored in detail. The organisation of this chapter is as follows. An overview of documentbase representation approaches is provided in the following section, where the Vector Space Model (VSM) is described in detail with regard to TC. In section 3.3 various techniques of feature selection, proposed particularly for TC problems, are reviewed. Finally a summary is presented in section 3.4.

3.2 Documentbase Representation

In documentbase representation, the "bag of *" or Vector Space Model (VSM) [Salton *et al.*, 1975] — where "*" stands for words, phrases, etc. — is considered appropriate for many text mining applications, especially when dealing with TC problems. VSM can be described as follows: given a documentbase *Đ*, each document $D_j \in Đ$ is represented by a single numeric vector, and each vector is a subset of some vocabulary V. The vocabulary V is a representation of the set of text-features (documentbase attributes) that are used to characterise the documents. VSM can be represented either (i) in a binary format as in [Kobayashi and Aono, 2004] — where "*each coordinate of a document vector is zero (when the corresponding attribute is absent) or unity (when the corresponding attribute is present)*"; or (ii) in a frequency format — where each coordinate of a document vector is zero (when the corresponding attribute is absent) or the frequency of occurrence (when the corresponding attribute is present).

Mladenic [1999] summarises well-established data structures on which documentbase representation approaches have been based, which, other than VSM, include graphs (e.g. $\mathcal{N}$-gram graph [McElligott and Sorensen, 1993]), lists (e.g. ordered word list [Cohen and Singer, 1996]), etc. Other documentbase representation approaches include string kernels (used in support vector machine based TC approaches) [Lodhi *et al.*, 2002], tensor space model [Liu *et al.*, 2005], etc.

In this thesis the "bag of *" model (VSM) is further considered. Text-features in this model can represent words, wordsets, phrases, concepts, etc.

- **Word:** In a linguistics context, a word is "*a unit of language that carries meaning and consists of one or more morphemes which are linked more or less tightly together; typically a word will consist of a root or stem and zero or more affixes*"[1]. In text mining related research, a word is usually identified as a text-unit (a group of text-characters), separated by punctuation marks, white space and/or wild card characters. A word can be further identified/defined as a "proper" (or "recognised") word if it belongs to (or "seems to belong to" with a consideration of misspelt/miswritten words) at least one of the known languages (i.e. Arabic, Chinese, English, French, Japanese, Spanish, etc.), and is not coupled with any non-language textual component, i.e. numbers, symbols, etc.

[1] http://en.wikipedia.org/wiki/Word_(linguistics)

- **Wordset:** The concept of an item as used in ARM (Association Rule Mining) was introduced in chapter 2. In ARM an itemset is a co-occurring set of items (binary-valued attributes) in a transactional database D_T. When modelling a documentbase *Đ*, where items within different transactions are expressed by distinct words in documents, an itemset corresponds to a wordset. Thus a wordset is an unordered group of two or more words that may co-occur in at least one document in *Đ*.

- **Phrase:** In a linguistics context, a phrase is "*a brief expression, sometimes a single word, but usually two or more words forming an expression by themselves, or being a portion of sentence; as, an adverbial phrase*"[2]. In text mining related research, a phrase can be defined as a "special" wordset permutation, where the words occur in a particular order.

- **Concept:** A concept, in a general case, is "*an abstract idea or a mental symbol, typically associated with a corresponding representation in language or symbology, that denotes all of the objects in a given category or class of entities, interactions, phenomena, or relationships between them*"[3]. In text mining, a concept can be represented by a (proper) word, wordset, phrase, other kinds of text-feature (i.e. text-symbols), and/or their combinations. A concept is generated (summed up) from a given documentbase that represents a group of text-features sharing a "similar" semantic, syntactic, lexical and/or ontological idea.

In TC two approaches are used to define the "bag of *" model — the "bag of words" and the "bag of phrases". This thesis concentrates on these two approaches only.

3.2.1 Bag of Words

The "bag of words" approach has been used in many TC studies, such as [Joachims, 1996], [Joachims, 1998], [Lewis and Ringuette, 1994], [Lewis *et al.*, 1996], [Nigam and McCallum, 1998] and [Wiener *et al.*, 1995]. In this approach, each

[2] http://www.brainydictionary.com/words/ph/phrase202473.html
[3] http://en.wikipedia.org/wiki/Concept

document is represented by the set of words that is used in the document. Information on the ordering of words within documents as well as the structure of the documents is lost. The problem with the approach is how to effectively and efficiently select a limited, computationally manageable, subset of words from the entire set represented in the documentbase.

The "bag of words" approach can take two forms: (i) full-texts — where all punctuation marks (sometimes, all non-alphabetic characters, i.e. numbers, symbols, etc.) are removed from the original documentbase, but other textual components are kept as complete as possible; and (ii) keywords — where significant words that contribute to the TC task are selected by feature selection approach(es).

- **Full-texts:** The full-texts based "bag of words" approach [Joachims, 1996] [Joachims, 1998] [Nigam and McCallum, 1998] retains the "rich" features of the original documentbase, and represents a document as a collection of proper words with no additional information regarding frequency. The main advantage of this approach is its simplicity. However, the "bag of words" may be very large and consequently an overwhelming number of classification rules may be generated, many of which are uninteresting.

- **Keywords:** The keywords based "bag of words" approach [Lewis and Ringuette, 1994] [Lewis *et al.*, 1996] [Wiener *et al.*, 1995] deals with the most significant words that are identified, employing various feature selection mechanisms, in a given documentbase. In this approach, the "bag of words" initially comprises a set of candidate keywords. For each candidate keyword, its significance can be evaluated by employing a feature selection mechanism. Well-established feature selection mechanisms will be further detailed in section 3.3.

3.2.2 Bag of Phrases

Instead of representing a documentbase using only words, a number of TC studies consider the usage of phrases, i.e. [Caropreso *et al.*, 2001], [Fürnkranz, 1998], [Katrenko, 2004], [Peng and Schuurmans, 2003], [Zhang *et al.*, 2005a], etc. In the "bag of phrases" approach, each element in a document vector represents a phrase describing an ordered combination of words appearing contiguously in sequence (sometimes with some maximum word gap). The motivation for this approach is that phrases seem to carry more contextual and/or syntactic information than single words. For example Scheffer and Wrobel [2002] argue that the "bag of words" representation does not distinguish between "*I have no objections, thanks*" and "*No thanks, I have objections*".

In [Lewis, 1992] and [Scott and Matwin, 1999] a sequence of experiments is described comparing the "bag of keywords" (keywords based "bag of words") approach with the "bag of phrases" approach in the TC context. The expectation in both was that the phrase based approach would work better than the keyword approach. However the reverse was discovered! In [Sebastiani, 2002] a number of reasons for this are given: (i) phrases have inferior statistical properties; (ii) phrases have lower frequency of occurrence than keywords; and (iii) the "bag of phrases" includes many redundant and/or noisy phrases. However, it is hypothesised here that the drawbacks of the "bag of phrases" can be overcome by the use of appropriate classification algorithms: it is clear that phrases will be found in fewer documents than corresponding key words, but conversely that they are expected to have a greater discriminating power [Coenen *et al.*, 2007].

The phrases in the "bag of phrases" are typically generated using either: (i) a $\mathcal{N}$-gram — where each sequence of $\mathcal{N}$ ordered and adjacent words in a document is simply identified as a phrase ($\mathcal{N} \leq$ the size of the document); or (ii) morpho-syntax — where ordered and adjacent words within a document are automatically combined into sequences (as phrases), based on the morpho-syntactic information of these words.

- **$\mathcal{N}$-gram:** Traditionally as shown in [Cavnar, 1994] an $\mathcal{N}$-gram is a sequence of $\mathcal{N}$ characters in a piece of text, but can equally be used with respect to words ($\mathcal{N}$-word) as in [Moschitti and Basili, 2004]. The $\mathcal{N}$-word based "bag of phrases" approach [Caropreso *et al.*, 2001] [Fürnkranz, 1998] [Peng and Schuurmans, 2003] generates a set of phrases by simply segmenting each different sequence of $\mathcal{N}$ ordered and adjacent words in a given document D, where $\mathcal{N} \leq |D|$ (the size of D) and the number of generated phrases is $|D| - \mathcal{N} + 1$. The main advantage of this approach as argued by Damashek [1995] and Peng *et al.* [2003] is its language-independency, efficiency and simplicity. However, the main question with respect to $\mathcal{N}$-gram techniques is what should the value of $\mathcal{N}$ be? This remains a current research issue. Note that this problem damages the usefulness of this approach especially with respect to language-independency (see section 2.8.3). Key phrases that significantly serve to classify documents can be further identified in all possible $\mathcal{N}$-gram phrases, based on some feature selection mechanism (see section 3.3).

- **Morpho-syntax:** The morpho-syntax based "bag of phrases" approach [Katrenko, 2004] [Moschitti and Basili, 2004] extracts a set of phrases from a collected document *D*, based on a two-phase procedure. In phase (i), the appropriate morpho-syntactic category (i.e. noun, adjective, verb, adverb, etc.) of each word in *D* is identified using a POS (Part-Of-Speech) tagging technique (see subsection 3.3.1). In phase (ii), ordered and adjacent words are gathered into sequences (each sequence accounts for a phrase), based on some morpho-syntactic patterns. Rajman and Besancon [1998] show that the morpho-syntactic patterns may include, for example: (pattern 1) 'noun (*N*) noun (*N*)', (pattern 2) 'noun (*N*) of (*prep*) noun (*N*)', (pattern 3) 'adjective (*Adj*) noun (*N*)', and (pattern 4) 'adjective (*Adj*) verbal (*Verb*)'. Based on these four patterns, they introduce an iterative phrase extraction procedure that avoids producing a large number of short and/or redundant phrases. For example, "*the sequence 'Secretary/N of/prep State/N George/N Shultz/N' was first transformed into 'Secretary-of-State/N George-Shultz/N' (patterns 2 and 1) and then combined into a unique term 'Secretary-of-State-George-Shultz/N' (pattern 1)*". Given a set of morpho-syntax phrases the key phrases that significantly contribute to the TC task can be identified using some feature selection mechanism (see section 3.3 below).

3.3 Feature Selection

In theory, the textual attributes of a document can include every word/phrase (text-feature) that might be expected to occur in a given documentbase. However, this is computationally unrealistic, so it requires some method of pre-processing documents to identify the key text-features that will be useful for a particular text mining application, such as TC. Feature selection is a well-established research area in information retrieval that aims to select a limited number of text-features from the entire set representing the documentbase. With respect to the classification task, Freitas [2002] categorises feature (attribute) selection methods into two types: the filter approach and the wrapper approach.

- **Filter Approach:** "*Attribute selection is performed without taking into account the classification algorithm that will be applied to the selected attributes*"; and "*in this approach the goal is to select a subset of attributes that preserves as much as possible the relevant information found in the entire set of attributes*".
- **Wrapper Approach:** "*Attribute selection is performed by taking into account the classification algorithm that will be applied to the selected attributes*"; and "*in this approach the goal is to select a subset of attributes that is 'optimized' for a given classification algorithm*".

Note here that the scope of this thesis is concerned with the filter approach rather than the wrapper approach. In a text mining context, basic techniques of feature selection (sometimes referred to as "feature reduction" from the opposite point of view) can generally be divided into two groups: (i) linguistic and (ii) statistical.

3.3.1 Linguistics based Feature Selection

Linguistics based feature selection/reduction methods identify significant text-features depending on the rules and/or regularities in semantics, syntax and/or lexicology. These techniques are designed with particular languages and styles of language as the target, and involve deep linguistic analysis. Typical methods in this group include stop-word lists, stemming, lemmatisation, synonym lists, POS tagging, word sense disambiguation, etc. Some of these, such as the use of stop-word lists and stemming, serve to eliminate words from the possible set of keywords; others serve to directly contribute to significant text-feature identification.

- **Stop-word Lists:** Ahonen-Myka *et al.* [1999] go some way toward addressing the problem of pruning text-features, in particular single words, with reference to a stop-word list. Words contained in a stop-word list, comprising common words which are not expected to contribute to the TC task (e.g. pronouns, conjunctions, articles, common adverbs, etc.) can be removed. In general some 40-50% of words in a document can be removed using a stop-word list, while the remaining words can be selected to be further processed. A stop-word list is a very common feature selection/reduction method. It has been widely applied in many TC approaches, such as [Combarro *et al.*, 2005], [Deng *et al.*, 2002], [Joachims, 1998], [Lam and Ho, 1998], [Lewis and Ringuette, 1994], [Yoon and Lee, 2005], [Zhuang *et al.*, 2005], etc.

- **Stemming:** Stemming [Melucci and Orio, 2003], also referred to as word normalisation [Airio *et al.*, 2004], aims to reduce the number of attributes (represented as words) in a documentbase by normalising words using their morphological root/stem. For example, the words "computed", "computing" and "computes" can be uniformly normalised (stemmed) as "comput". In text mining applications one widely available stemming algorithm is the well-established Porter algorithm first proposed in [Porter, 1980]. Many TC studies make use of Porter stemming, e.g. [Combarro *et al.*, 2005], [Deng *et al.*, 2002], [Hulth and Megyesi, 2006], [Joachims, 1998], [Ozgur *et al.*, 2005], etc. However Zaïane and Antonie [2002] report (in their initial TC testing) that Porter stemming "*does not improve effectiveness significantly*".

- **Lemmatisation:** Plisson *et al.* [2004] assert that lemmatisation is "*an important preprocessing step for many applications of text mining*"; and comment that lemmatisation is "*similar to word stemming but it does not require (the production of the) stem of the word but to replace the suffix of a word, appearing in free text, with a (typically) different word suffix to get the normalized word form*". For instance, words "computed", "computing" and "computes" would be lemmatised as "compute", while stemmed as "comput". With regard to TC problems, it seems that lemmatisation can only slightly improve the classification performance, but has as Bel *et al.* [2003] and Leopold and Kindermann [2002] show drawbacks in terms of processing efficiency.

- **Synonym Lists:** A list of synonyms and near-synonyms (also called a "thesaurus") can, as Senellart and Blondel [2004] argue, be provided by the user or generated automatically from a given documentbase. It can be used to reduce the number of documentbase features (attributes) — words/phrases that share the same semantic meaning should be integrated in one, e.g. “foe” & “adversary”, “close but no cigar” & “losing”, etc. One of the early ideas of automatic identification of synonyms and near-synonyms is the distribution hypothesis [Harrism, 1968], which states that words with similar meanings tend to appear in similar contexts. Zhang *et al.* [2005b] integrate synonyms to reduce the number of documentbase dimensions (attributes) for TC.

- **POS Tagging:** POS (Part-Of-Speech) tagging as discussed in [Brill, 1992] and [Cutting, 1992], and also called grammatical tagging, is a mature Natural Language Processing (NLP) technique. As Zavrel and Daelemans [1999] mention, it aims to assign the appropriate part-of-speech (morpho-syntactic category) to each word in a phrase, sentence, or paragraph. POS tagging can be used to distinguish informative text-features from uninformative ones. Rajman and Besancon [1998] believe that nouns, verbs and adjectives contain the most natural language information, and suggest that for text mining applications a documentbase can be “better” represented by only nouns, verbs, adjectives and their combinations. Rajman and Besancon further present an approach to identify/select key phrases in a documentbase, based on the morpho-syntactic (POS) patterns (as previously described in subsection 3.2.2). Several studies also consider using POS tagging to improve the TC performance (see for example [Goucalves and Quaresma, 2005], [Liao *et al.*, 2003] and [Zhang *et al.*, 2005a]).

- **Word Sense Disambiguation:** Word Sense Disambiguation (WSD) is the automatic process of determining the most appropriate sense of each word in a natural language phrase, sentence, or paragraph. Early interest in this NLP problem can be dated back to 1950’s [Ide and Veronis, 1998], but a unique definition of “word sense” has not yet been agreed upon. In general terms, a word sense can be identified as the word’s morpho-syntactic category (the case of POS tagging [Kelly and Stone, 1975]), its semantic meaning, etc. Ide and Veronis [1998] indicate that the WSD process consists of two stages: (i) “*the determination of all the different senses for every word relevant (at least) to the* *text or discourse under consideration*"; and (ii) "*a means to assign each occurrence of a word to the appropriate sense*". Uejima *et al.* [2003] go some way toward improving TC performance by resolving word semantic ambiguity.

3.3.2 Statistics based Feature Selection

Statistics based feature selection/reduction techniques automatically compute a weighting score for each text-feature in a document. A significant text-feature can be identified when its weighting score exceeds a user-defined weighting threshold. Methods in this group do not involve linguistic analysis but focus on some documentbase statistics. With regard to TC, the common intuitions of various methods in this group can be described as: (i) the more times a text-feature appears in a document the more relevant it is to the class of the document; and (ii) the more times a text-feature appears across the documentbase in documents of all classes the worse it is at discriminating between the classes.

In TC a documentbase *Đ* is usually represented in the binary format (rather than the frequency format) of VSM. As Ozgur *et al.*, [2005] argue, this "*has the advantages of being very simple and requiring less memory*", especially when dealing with a "*high dimensional text domain*". Based on a selected statistical model, a weighting score can be calculated for each text-feature u_h in a document $D_j \in Đ$. Common statistical models that deal with the binary format of VSM include: document frequency, DIA (Darmstadt Indexing Approach) association factor, odds ratio, relevancy score, mutual information, chi-square statistics, correlation coefficient, and GSS (Galavotti·Sebastiani·Simi) coefficient.

- **Document Frequency:** Document Frequency (DF) is the number of documents in which a text-feature u_h appears [Yang and Pedersen, 1997]. It can be defined as:

$$df\_score(u_h) = count(u_h \in Đ) \; / \; |Đ| \, ,$$

where $count(u_h \in Đ)$ is the number of documents containing u_h in *Đ*, and |*Đ*| is the size function (cardinality) of the set *Đ*. This simple function can be used to identify significant text-features in a documentbase, where a significant text-feature is defined as having a "high" DF score greater than some pre-determined (DF) threshold. The basic assumption of DF based feature selection/reduction is that "*rare terms are either non-informative for category prediction, or not influential in global performance*" [Yang and Pedersen, 1997]. With regard to TC, Fragoudis *et al.* [2005] describe an ameliorated DF weighting scheme, namely DF^C (Class based Document Frequency) which takes the classes into consideration. A simple probabilistic formula for DF^C is:

$$df^C\_score(u_h, C_i) = P(u_h \mid C_i)\ .$$

This formula can also be written as: $count(u_h \in C_i) / |C_i|$, where C_i represents a set of documents labelling with a particular text-class, $count(u_h \in C_i)$ is the number of documents containing u_h in C_i, and $|C_i|$ is the size function (cardinality) of the set C_i. This value expresses the probability with which the feature occurs in documents of the given class.

- **DIA Association Factor:** The Darmstadt Indexing Approach (DIA) [Fuhr, 1989] was originally "*developed for automatic indexing with a prescribed indexing vocabulary*" [Fuhr and Buckley, 1991]. In a machine learning context, Sebastiani [2002] argues that this approach "*considers properties (of terms, documents, categories, or pairwise relationships among these) as basic dimensions of the learning space*". Examples of the properties include the length of a document, the frequency of occurrence between a text-feature and a document/class, etc. One of the pair-wise relationships considered is the term-category relationship, noted as the DIA Association Factor (DIAAF) [Sebastiani, 2002], that can be applied to select significant text-features for TC problems. The calculation of the DIAAF, and reported in [Sebastiani, 2002], is achieved by using:

$$diaaf\_score(u_h, C_i) = P(C_i \mid u_h) = count(u_h \in C_i) / count(u_h \in Đ)\ .$$

This weighting score expresses the proportion of the feature's occurrence in the given class divided by the feature's documentbase occurrence. DIAAF will be further used to develop two new statistical keyword identification approaches (see sections 4.3.3 and 4.3.4).

- **Odds Ratio:** The Odds Ratio (OR) [Van Rijsbergen, 1979] was originally developed "*for selecting terms for relevance feedback*" [Zheng and Srihari, 2003]. The basic idea of this approach as argued by Zheng and Srihari [2003] is that "*the distribution of features on the relevant documents is different from the distribution of features on the non-relevant documents*". Mladenic [1998] utilises this statistical measure to identify significant text-features for the TC task. The calculation of the OR score can be specified in probabilistic form using:

$$or\_score(u_h, C_i) = (P(u_h \mid C_i) \times (1 - P(u_h \mid \neg C_i))) \\ / \, ((1 - P(u_h \mid C_i)) \times P(u_h \mid \neg C_i)) \, ,$$

where $\neg C_i$ (equal to $Đ - C_i$) represents the set of documents labelled with the complement of the pre-defined class C_i. This formula can also be written in the following form:

$$or\_score(u_h, C_i) = ((count(u_h \in C_i) / |C_i|) \times (1 - count(u_h \in (Đ - C_i)) / |Đ - C_i|)) \\ / \, ((1 - count(u_h \in C_i) / |C_i|) \times (count(u_h \in (Đ - C_i)) / |Đ - C_i|)) \, .$$

The OR score expresses the ratio of two multiplications. The numerator shows the probability with which the feature occurs in documents of the given class multiplied by the complement of the probability with which the feature occurs in documents that are not labelled with the given class; whilst the denominator shows the complement of the probability with which the feature occurs in documents of the given class multiplied by the probability with which the feature occurs in documents that are not labelled with the given class.

- **Relevancy Score:** The initial concept of Relevancy Score (RS) was introduced by Salton and Buckley [1988] as relevancy weight. It aims to measure how "unbalanced" a text-feature (term) u_h is across documents in $Đ$ with and without a particular text-class C_i. They define a term's relevancy weight as: "*the proportion of relevant documents in which a term occurs divided by the proportion of nonrelevant items in which the term occurs*". In [Wiener *et al.*, 1995] the idea of RS was proposed based on relevancy weight with the objective of selecting significant text-features in $Đ$ for the TC application. A term's relevancy score can be defined as: the number of relevant (the target text-class associated) documents in which a term occurs divided by the number of non-relevant documents in which a term occurs. Fragoudis *et al.* [2005] and Sebastiani [2002] show that RS can be calculated using:

$$relevancy\_score(u_h, C_i) = log((P(u_h \mid C_i) + d) / (P(u_h \mid \neg C_i) + d))$$
$$= log((count(u_h \in C_i) / |C_i| + d) / (count(u_h \in (Đ - C_i)) / |Đ - C_i| + d)) \text{ ,}$$

where d is a constant damping factor. In [Wiener *et al.*, 1995] the value of d was initialised as 1/6. RS will be further used to develop the DIAAF-based-RS keyword identification technique (see section 4.3.3).

- **Mutual Information:** Early work on Mutual Information (MI) can be found in [Church and Hanks, 1989] and [Fano, 1961]. This statistical model is used to determine whether a genuine association exists between two text-features or not. In TC investigation, MI has been broadly employed in a variety of approaches to select the most significant text-features that serve to classify documents. The calculation of the MI score between a text-feature u_h and a pre-defined text-class C_i is achieved using:

$$mi\_score(u_h, C_i) = log(P(u_h, C_i) / (P(u_h) \times P(C_i))) = log(P(u_h \mid C_i) / P(u_h))$$
$$= log((count(u_h \in C_i) / |C_i|) / (count(u_h \in Đ) / |Đ|)) \text{ .}$$

This score expresses the proportion (in a logarithmic term) of the probability with which the feature occurs in documents of the given class divided by the probability with which the feature occurs in the documentbase.

- **Chi-square Statistics:** The Chi-squared (χ^2) statistic has previously been described in section 2.6.3, as a rule weighting/ordering mechanism in CARM. The Chi-squared statistic can also be applied to measure the lack of independence between a term u_h and a pre-defined class C_i and "*can be compared to the χ^2 distribution with one degree of freedom to judge extremeness*" [Yang and Pedersen, 1997] [Zheng and Srihari, 2003]. The χ^2 score between u_h and C_i can be calculated (see [Fragoudis *et al.*, 2005] [Schütze *et al.*, 1995] [Yang and Pedersen, 1997] [Zheng and Srihari, 2003]) using:

$$\chi^2\_score(u_h, C_i) = (|Đ| \times (P(u_h, C_i) \times P(\neg u_h, \neg C_i) - P(u_h, \neg C_i) \times P(\neg u_h, C_i))^2) \\ / (P(u_h) \times P(\neg u_h) \times P(C_i) \times P(\neg C_i)) ,$$

where $\neg u_h$ represents a document that does not involve the feature u_h, and the probabilistic components of the formula $P(u_h, C_i)$, $P(\neg u_h, \neg C_i)$, $P(u_h, \neg C_i)$, $P(\neg u_h, C_i)$, $P(u_h)$, $P(\neg u_h)$, $P(C_i)$ and $P(\neg C_i)$ can also be defined as $count(u_h \in C_i) / |Đ|$, $count(\neg u_h \in (Đ - C_i)) / |Đ|$, $count(u_h \in (Đ - C_i)) / |Đ|$, $count(\neg u_h \in C_i) / |Đ|$, $count(u_h \in Đ) / |Đ|$, $count(\neg u_h \in Đ) / |Đ|$, $|C_i| / |Đ|$ and $|Đ - C_i| / |Đ|$. If the feature/term and the class are independent, the calculated χ^2 score has a natural value 0 [Yang and Pedersen, 1997] [Zheng and Srihari, 2003]. Yang and Pedersen [1997] also indicate a major difference between χ^2 and MI scores — "*χ^2 values are comparable across terms for the same category*".

- **Correlation Coefficient:** The Correlation Coefficient (CC) approach described in [Zheng and Srihari, 2003], and also referred to as the NGL (Ng·Goh·Low) coefficient approach in [Fragoudis *et al.*, 2005] and [Sebastiani, 2002], is a variant of the χ^2 feature selection metric. It was originally introduced in [Ng *et al.*, 1997] to generate a better set of key/significant features and improve the performance of the χ^2 metric. CC, as Ng *et al.* [1997] argue, "*can be viewed as a 'one-sided' χ^2 metric*" where $CC^2 = \chi^2$. Thus the calculation of CC is achieved using:

$$cc\_score(u_h, C_i) = (\sqrt{|Đ|} \times (P(u_h, C_i) \times P(\neg u_h, \neg C_i) - P(u_h, \neg C_i) \times P(\neg u_h, C_i))) \\ / (\sqrt{P(u_h) \times P(\neg u_h) \times P(C_i) \times P(\neg C_i)}) .$$

Ng *et al.* [1997] argue that "*words that come from the irrelevant texts or are highly indicative of non-membership in*" a class C_i are not as useful; and indicate that CC "*selects exactly those words that are highly indicative of membership in a category, whereas the χ^2 metric will not only pick out this set of words but also those words that are indicative of non-membership in the category*".

- **GSS Coefficient:** The GSS (Galavotti·Sebastiani·Simi) coefficient defined in [Galavotti *et al.*, 2000] is a further simplified variant of the χ^2 metric. The GSS coefficient is defined as follows:

$$gss\_score(u_h, C_i) = P(u_h, C_i) \times P(\neg u_h, \neg C_i) - P(u_h, \neg C_i) \times P(\neg u_h, C_i) \;.$$

Galavotti *et al.* [2000] provide an explanation of the rationale of replacing the factors $\sqrt{|Đ|}$, $\sqrt{P(u_h) \times P(\neg u_h)}$, and $\sqrt{P(C_i) \times P(\neg C_i)}$ in the CC calculation, and demonstrate that this very simple approach (GSS) can produce a comparable performance to the χ^2 metric. GSS will be further utilised to produce the DIAAF-based-GSS keyword identification mechanism (see section 4.3.4).

A text-feature may appear more than once in a natural language document. Hence in some cases, instead of simply representing a documentbase in the binary format of VSM, it makes sense to make use of the absolute number of occurrences (or frequency of appearance) of a text-feature in a document. Given a *Đ* that is represented in the frequency format of VSM, a weighting score can be calculated for each text-feature in a document $D_j \in Đ$, based on a selected statistical model. There are two typical models, TFIDF (Term Frequency Inverse Document Frequency) and the categorical term descriptor approach.

- **TFIDF:** The most commonly applied weighting model in feature selection/reduction is the well-known TFIDF approach [Salton and Buckley, 1988]. The standard version of TFIDF is calculated by looking at TF (Term Frequency) — the frequency of appearance of a term (text-feature) u_h in a document $D_j \in Đ$, and then multiplying it by the IDF (Inverse Document Frequency) [Spärck Jones, 1972] — the log function of the division of the total number of documents in *Đ* by the number of documents within *Đ* in which u_h occurs. The formula of the standard TFIDF can be written as:

$$tfidf\_score(u_h, D_j) = freq(u_h, D_j) \times log(|Đ| \,/\, count(u_h \in Đ)) \;.$$

In order to fit a TFIDF score into the [0, 1] interval, Sebastiani [2002] normalises the standard TFIDF score by using a cosine normalisation. Thus:

$$tfidf^N\_score(u_h, D_j) = tfidf\_score(u_h, D_j) \,/\, \sqrt{\rho} \;,$$

$$\text{where } \rho = \textstyle\sum_{\{l\,=\,1\ldots|D_j|\}} (tfidf\_score(u_l, D_j))^2 \;.$$

Rajman and Besancon [1998] present a variant of the standard TFIDF approach, namely the SMART weighting scheme of TFIDF. It aims to produce a more precise weighting score for u_h, as follows:

$$\begin{cases} tfidf^{S}\_score(u_h, D_j) = 0.5 \times (1 + freq(u_h, D_j) / \gamma) \\ \qquad \times\, log(|Đ| / count(u_h \in Đ)) \text{ if } freq(u_h, D_j) \geq 1 \text{ , and} \\ tfidf^{S}\_score(u_h, D_j) = 0 \text{ otherwise ,} \end{cases}$$

where $\gamma = \max_{\{m = 1 \ldots |D_j|\}} tfidf\_score(u_m, D_j)$.

Soucy and Mineau [2005] note that "*there are many variants of TFIDF*", and indicate that the simple TFIDF variant, commonly used in TC, can be formulated as:

$$\begin{cases} tfidf^{C}\_score(u_h, D_j) = log(1 + freq(u_h, D_j)) \\ \qquad \times\, log(|Đ| / count(u_h \in Đ)) \text{ if } freq(u_h, D_j) \geq 1 \text{ , and} \\ tfidf^{C}\_score(u_h, D_j) = 0 \text{ otherwise .} \end{cases}$$

As the TF part appears in all versions of the TFIDF weighting model, it requires the documentbase to be represented in the frequency format of VSM. From the calculation of IDF, it can be observed that significant text-features (terms), selected by TFIDF, perform well to distinguish between documents, but are not suitable for TC application. This occurs for the reason that TFIDF tends to extract text-features that only appear in one or few documents in *Đ*, and ignores terms that repetitively occur in a large amount of documents even if these documents are associated with a particular text-class in *Đ*. This is the opposite of what TC expects.

- **Categorical Term Descriptor:** The Categorical Term Descriptor (CTD), devised by Bong and Narayanan [2004], was proposed to improve the performance of TFIDF for TC, based on additionally involving a consideration of the text-class. A CTD score for a term u_h to a text-class C_i, $ctd\_score(u_h, C_i)$, can be calculated by multiplying three components:

1. **TF^{C} (Class based Term Frequency)** — the frequency of appearance of u_h across the documents in C_i;

2. **IDF^C (Class based Inverse Document Frequency)** — the log function of the division of the total number of documents in C_i to the number of documents within C_i in which u_h occurs; and
3. **ICF (Inverse Category Frequency)** — the log function of the division of the total number of pre-defined classes to the number of classes within *Đ* in which u_h occurs.

Again, CTD requires *Đ* to be represented in the frequency format of VSM.

3.4 Summary

In this chapter, existing approaches to documentbase pre-processing for the TC have been reviewed. The discussion concentrated on the "bag of *" model or VSM and both the "bag of words" and the "bag of phrases" approaches were described in detail. From the discussion a pure language-independent "bag of phrases" strategy that avoids the drawback of the $\mathcal{N}$-gram model is clearly desirable. With respect to both the "bag of words" and the "bag of phrases" approaches it is desirable to reduce the number of identified words/phrases using some kind of filter, so that only the most key/significant features remain to be used in the classification stage. In the discussion of feature selection/reduction, both linguistics based techniques and statistics based methods were analysed. From the discussion it is clear that some methods, especially the linguistics based approaches, will require more computation and/or more language-specific knowledge. The aim of this thesis is investigating documentbase pre-processing techniques that are (i) efficient in dealing with large documentbases, and (ii) language-independent. Therefore statistical feature selection/reduction techniques will be further focused upon in this thesis.

Chapter 4

Language-independent Documentbase Pre-processing

4.1 Introduction

In this chapter, a number of language-independent documentbase pre-processing techniques, to support single-label multi-class TC, are introduced. The discussion focuses on both the "bag of words" and the "bag of phrases" approaches. Four statistically based keyword selection mechanisms are proposed, where each can be used to identify those key words that are expected to significantly contribute to the distinction between classes in a documentbase:

1. Local-To-Global Support Ratio (LTGSR);
2. Local-To-Global Frequency Ratio (LTGFR);
3. DIA Association Factor based Relevancy Score (DIAAF-based-RS); and
4. DIA Association Factor based Galavotti·Sebastiani·Simi (DIAAF-based-GSS).

A further number of *significant phrase* identification strategies are also proposed. The emphasis in all cases is on language-independence so that the techniques described here have general applicability regardless of the language(s) in which the documentbase to be processed are presented.

The general approach employed, for each document in the training set, is as follows:

1. Remove *common* words (defined as a type of *noise* words), i.e. words that are unlikely to contribute to a characterisation of the document.
2. Remove *rare* words (defined as another type of *noise* words), i.e. words that are unlikely to lead to generally applicable classification rules.

3. From the remaining words select those *significant* words that serve to differentiate between classes.

In the case of the "bag of phrases" approach the fourth step is to generate significant phrases from the significant words and associated words identified.

The organisation of this chapter is as follows. The following section presents the proposed statistical identification of noise words in a documentbase. Section 4.3 describes the four language-independent keyword selection approaches in detail, coupled with two potential significant word list construction approaches and two final significant word selection approaches. In section 4.4, four language-independent significant phrase identification strategies are proposed, with an appropriate example provided to illustrate each case. Finally a summary is given in section 4.5.

4.2 Language-independent Noise Word Identification

Common and rare words are collectively considered to be the *noise* words in a documentbase. They can be identified by their *support* value, i.e. the percentage of documents in the training set in which the word appears. Common words are words with a support value above a user-defined Upper Noise Threshold (UNT), and are referred to as Upper Noise Words (UNW). Rare words are those with a support value below a user-defined Lower Noise Threshold (LNT), and are referred to as Lower Noise Words (LNW).

Given any standard documentbase the majority of words will have a low support value, and only a small number of common words will appear in more than 50% of the documents. Examples using common textual datasets are further provided in chapter 5 (see section 5.2.5). If the LNT is set too high then very few non-noise words will be identified and a minimal classifier will be generated which is unlikely to perform well. Alternatively if the LNT is set too low a large number of rare words will be included resulting in a classifier that is likely to "over-fit". The UNT must of course exceed the LNT value, and the "gap" (distance) between the two values determines the number of identified non-noise words from which significant words will be drawn and consequently, if indirectly, the number of identified phrases. If the "gap" is large, a large number of non-noise words will be identified which in turn will impose computational constraints on the classification process.

4.3 Language-independent Significant Word Identification

The desired set of significant words is drawn from an ordered list of potential significant words. A potential significant word also referred to as a *key* word is a non-noise word whose *contribution* value exceeds some user-specified threshold *G*. The contribution value of a word is a measure of the extent to which the word serves to differentiate between classes and can be calculated in a number of ways. For the present study four methods are considered while at the same time acknowledging the fact that alternative approaches exist (see section 3.3.2).

Those words whose contribution exceeds the threshold *G* are placed into a potential significant word list, ordered according to contribution value in a descending manner. This list may include words that are significant for more than one class (noted as "*all words*"), or we may decide to include only those words that are significant with respect to one class only (i.e. "*unique*"). From the potential significant word list we choose the final list of significant words. Two strategies can be proposed for achieving this. The first method is simply to choose the first *K* (user-specified parameter) words from the ordered list (referred to as "*top K*"). It may, however, result in an unequal distribution between classes. In the second approach we choose the top "K / |Є|" words for each class (referred to as "*dist*" — i.e. short for "distribution"), so as to include an equal number of significant words for each class, where Є = $\{C_1, C_2, \ldots, C_{|Є|-1}, C_{|Є|}\}$ is the set of pre-defined text-classes within *Đ*; |Є| is the size function (cardinality) of the set Є; and "K / |Є|" is a user-supplied integer.

Example 4.1: Significant Word List Generation

An example that illustrates the process of generating a (final) list of significant words via the construction of a potential significant word list is provided here. Given a documentbase *Đ* that consists of three classes ("business", "sports" and "entertainments") assume that for each class there are four non-noise words (see Table 4.1). Let the significance threshold (*G*) be 0.5. Hence words with a contribution value ≥ 0.5 are considered to be the potential-significant (key) words in *Đ* (see Table 4.2). There are two strategies available to construct a potential significant word list: "all words" and "unique" (as introduced above). From Table 4.2, it can be seen that the word "international" (highlighted) is found to be potentially significant in both classes "business" and "sports". Thus the constructed potential significant word list, based on the "unique" strategy, does not include this word. Table 4.3 shows the potential significant word lists that are constructed based on "all words" and "unique" criteria.

Class	**Word 1**	**Word 2**	**Word 3**	**Word 4**
Business	financial (0.9)	international (0.5)	restaurant (0.3)	tax (0.9)
Sports	championship (0.5)	club (0.2)	game (0.9)	international (0.6)
Entertainments	gallery (0.4)	jazz (0.7)	music (0.8)	show (0.8)

Table 4.1: List of four non-noise words for each given class (contribution values are given in parentheses)

Class	**Potential Significant Word 1**	**Potential Significant Word 2**	**Potential Significant Word 3**
Business	financial (0.9)	tax (0.9)	international (0.5)
Sports	game (0.9)	international (0.6)	championship (0.5)
Entertainments	music (0.8)	show (0.8)	jazz (0.7)

Table 4.2: List of the potential significant (key) words for the given classes (*G* = 0.5)

Strategy	Potential Significant Word List
All words	{ financial (0.9), game (0.9), tax (0.9), music (0.8), show (0.8), jazz (0.7), international (0.6), championship (0.5), international (0.5) }
Unique	{ financial (0.9), game (0.9), tax (0.9), music (0.8), show (0.8), jazz (0.7), championship (0.5) }

Table 4.3: Two constructed potential significant word lists

Assume that the value of *K* is 6, which means that only the top (most significant) 6 words from the potential significant word list are chosen in building the final list of significant words. There are two strategies to select the final significant words, "top *K*" and "dist" (as introduced above). Table 4.4 shows four final significant word lists, where each is generated by combining one potential significant word list construction strategy with one final significant word selection strategy. Note that both "top *K*" based final significant word lists (one is generated with "all words" and the other generated with "unique") are identical in this example.

	Top *K*	Dist
All words	{ financial (0.9), game (0.9), tax (0.9), music (0.8), show (0.8), jazz (0.7) }	{ financial (0.9), game (0.9), tax (0.9), music (0.8), show (0.8), international (0.6) }
Unique	{ financial (0.9), game (0.9), tax (0.9), music (0.8), show (0.8), jazz (0.7) }	{ financial (0.9), game (0.9), tax (0.9), music (0.8), show (0.8), championship (0.5) }

Table 4.4: Four generated final significant word lists

4.3.1 Method 1: Local-To-Global Support Ratio (LTGSR)

In this and the following three subsections the four keyword identification methods listed in the introduction to this chapter are presented in detail. The first approach to be considered for identifying keywords (potential significant words), namely

Local-To-Global Support Ratio (LTGSR), does so on the basis of the ratio of a word's *local* support to its *global* support. Here, the local support of a word u_h represents the document frequency of u_h within a given class C_i; while the word's global support is the document frequency of u_h within the documentbase $Đ$. The LTGSR score for u_h with respect to C_i can be calculated using:

$$ltgsr\_score(u_h, C_i) = support(u_h \in C_i) \,/\, support(u_h \in Đ) \;.$$

The intuition of this approach is that a significant word in relation to a particular class must demonstrate that the (local) proportion of documents for a particular class that contain the word is greater than the (global) proportion of all documents containing it.

Based on the computation of support (see section 2.4), the calculation of a LTGSR score can be expressed as:

$$ltgsr\_score(u_h, C_i) = (count(u_h \in C_i) \,/\, |C_i|) \,/\, (count(u_h \in Đ) \,/\, |Đ|) \;.$$

This can be seen as a variant of the MI approach (see section 3.3.2), but without the logarithmic operation.

Example 4.2: LTGSR Score Calculation

Given a documentbase $Đ$ containing 100 documents equally divided into 4 classes (i.e. 25 per class) and assuming that word u_h appears in 30 of the documents (i.e. $support(u_h \in Đ)$ = 30%), then the LTGSR score per class can be calculated as shown in Table 4.5.

Class	# docs per class	# docs with u_h per class	Support ($u_h \in C_i$)	Support ($u_h \in Đ$)	LTGSR Score
1	25	15	60%	30%	2.00
2	25	10	40%	30%	1.33
3	25	5	20%	30%	0.67
4	25	0	0%	30%	0

Table 4.5: LTGSR score calculation

The algorithm for identifying keywords in *Đ*, based on LTGSR, is given as follows:

Algorithm 4.1: Keyword Identification — LTGSR
Input: (a) A documentbase *Đ* (the training part, where the noise words have been removed);
(b) A user-defined significance threshold *G*;
Output: A set of identified keywords S_{KW};
Begin Algorithm:

```
$S_{KW}$ ← an empty set for holding the identified keywords in Đ;
Є ← catch the set of pre-defined text-classes within Đ;
$W_{GLO}$ ← read Đ to create a global word set, where the word documentbase
        (global) support $supp_{GLO}$ is associated with each word $u_h$ in $W_{GLO}$;
for each $C_i \in$ Є do
    $W_{LOC}$ ← read documents that reference $C_i$ to create a local word set,
        where the local support $supp_{LOC}$ is associated with each word
        $u_h$ in $W_{LOC}$;
    for each word $u_h \in W_{LOC}$ do
        contribution ← $u_h.supp_{LOC} / u_h.supp_{GLO}$;
        if (contribution $\geq G$) then
            add $u_h$ into $S_{KW}$;
    end for
end for
return ($S_{KW}$);
```

End Algorithm

4.3.2 Method 2: Local-To-Global Frequency Ratio (LTGFR)

A similar approach to LTGSR is the Local-To-Global Frequency Ratio (LTGFR), where the given documentbase *Đ* is represented in terms of vectors containing the frequency count of each word in a document (see section 3.2). The motivation of this approach is that in automatic text retrieval, it appears that frequency based keyword selection mechanisms (i.e. TFIDF) outperform binary (support) based methods [Salton and Buckley, 1988]. The LTGFR score can be calculated by using:

$$ltgfr\_score(u_h, C_i) = (TF(u_h, C_i) / N_i) / (TF(u_h) / N) ,$$

where $TF(u_h, C_i)$ is the number of actual occurrences of word u_h in documents in class C_i, $TF(u_h)$ is the number of actual occurrences of word u_h in documentbase *Đ*, N is the total number of words in *Đ*, and N_i is the total number of words contained in documents labelled as class C_i. The ratio $TF(u_h) / N$ defines the overall term frequency of u_h in *Đ*; if the corresponding ratio $TF(u_h, C_i) / N_i$ is significantly greater than this, then a contribution value G greater than 1 will indicate a potential significant (key) word.

Example 4.3: LTGFR Score Calculation

Given a documentbase *Đ* containing 100 documents (each document consists of 100 words) equally divided into 4 classes (i.e. 2,500 words per class) and assuming that word u_h occurs 800 times in the documentbase (i.e. $TF(u_h) / N$ = 800 / 10,000 = 8%), then the LTGFR score per class can be calculated as shown in Table 4.6.

Class	**# words per class (N_i)**	**Frequency of word u_h per class**	**$TF(u_h, C_i) / N_i$**	**$TF(u_h) / N$**	**LTGFR Score**
1	2,500	400	16%	8%	2.00
2	2,500	300	12%	8%	1.50
3	2,500	100	4%	8%	0.50
4	2,500	0	0%	8%	0

Table 4.6: LTGFR score calculation

The algorithm for identifying keywords in *Đ*, based on LTGFR, is given as follows:

Algorithm 4.2: Keyword Identification — LTGFR

```
Input: (a) A documentbase Đ (the training part, where the noise words have been
           removed);
       (b) A user-defined significance threshold $G$;
Output: A set of identified keywords $S_{KW}$;
Begin Algorithm:
(1)  $S_{KW}$ ← an empty set for holding the identified keywords in Đ;
(2)  Є ← catch the set of pre-defined text-classes within Đ;
(3)  $W_{GLO}$ ← read Đ to create a global word set, where the word documentbase
           frequency $freq_{GLO}$ is associated with each word $u_h$ in $W_{GLO}$;
(4)  for each $C_i \in$ Є do
(5)      $W_{LOC}$ ← read documents that reference $C_i$ to create a local word set,
               where the local frequency $freq_{LOC}$ is associated with each
               word $u_h$ in $W_{LOC}$;
(6)      for each word $u_h \in W_{LOC}$ do
(7)          contribution ← $u_h.freq_{LOC} / u_h.freq_{GLO}$;
(8)          if (contribution $\geq G$) then
(9)              add $u_h$ into $S_{KW}$;
(10)     end for
(11) end for
(12) return ($S_{KW}$);
End Algorithm
```

4.3.3 Method 3: DIA Association Factor based Relevancy Score (DIAAF-based-RS)

In section 3.3.2 two statistics based feature selection mechanisms, the Darmstadt Indexing Approach Association Factor (DIAAF) and Relevancy Score (RS), were described. In this subsection, an alternative statistics based feature selection method is proposed — the DIAAF-based-RS (DIA Association Factor based Relevancy Score) — that utilises the binary vector representation of a documentbase. DIAAF-based-RS is a variant of the original RS approach that makes use of the well-established DIAAF approach.

Recall that the formula for calculating the RS score is given by:

$$rs\_score(u_h, C_i) = log((P(u_h \mid C_i) + d) / (P(u_h \mid \neg C_i) + d)) .$$

It can also be considered as an extension of the Class based Document Frequency (DF^C) approach (see section 3.3.2), where

$$df^C\_score(u_h, C_i) = P(u_h \mid C_i) .$$

The DIAAF score is calculated using:

$$diaaf\_score(u_h, C_i) = P(C_i \mid u_h) .$$

Substituting for the DF^C related parts into the RS score formula using the DIAAF formula, a new RS style formula (DIAAF-based-RS) is defined:

$$diaaf\text{-}based\text{-}rs\_score(u_h, C_i) = log((P(C_i \mid u_h) + d) / (P(C_i \mid \neg u_h) + d)) ,$$

where $\neg u_h$ represents a document that does not involve the feature u_h, and d is a constant damping factor (as mentioned in the original RS). The formula can be further expanded as:

$$\begin{gathered} diaaf\text{-}based\text{-}rs\_score(u_h, C_i) = log((count(u_h \in C_i) / count(u_h \in Đ) + d) \\ / (count(\neg u_h \in C_i) / count(\neg u_h \in Đ) + d)) . \end{gathered}$$

The rationale of this approach is that a significant text-feature (term) with respect to a particular class should: (i) have a high ratio of the class based term support (document frequency) to the documentbase term support, and/or (ii) a low ratio of the class based term support of non-appearance to the documentbase term support of non-appearance.

Example 4.4: DIAAF-based-RS Score Calculation

Given a documentbase *Đ* containing 100 documents equally divided into 4 classes (i.e. 25 per class), and assuming that word u_h appears in 30 of the documents and that the value of *d* (constant damping factor) is 0, then the DIAAF-based-RS score per class can be calculated as shown in Table 4.7.

Class	# docs per class	# docs with u_h per class	# docs without u_h per class	# docs with u_h in *Đ*	# docs without u_h in *Đ*	Prob. $(C_i\|u_h) + d$	Prob. $(C_i\|\neg u_h) + d$	DIAAF -based-RS Score
1	25	15	10	30	70	0.500	0.143	0.544
2	25	10	15	30	70	0.333	0.214	0.192
3	25	5	20	30	70	0.167	0.286	-0.234
4	25	0	25	30	70	0	0.357	-∞

Table 4.7: DIAAF-based-RS score calculation

The algorithm for identifying keywords in *Đ*, based on DIAAF-based-RS, is given as follows:

```
Algorithm 4.3: Keyword Identification — DIAAF-based-RS
Input: (a) A documentbase Đ (the training part, where the noise words have been
           removed);
       (b) A user-defined significance threshold $G$;
       (c) A constant damping factor $d$;
Output: A set of identified keywords $S_{KW}$;
Begin Algorithm:
    $S_{KW}$ ← an empty set for holding the identified keywords in Đ;
    Є ← catch the set of pre-defined text-classes within Đ;
    $W_{GLO}$ ← read Đ to create a global word set, where the word documentbase
            support $supp_{GLO}$ is associated with each word $u_h$ in $W_{GLO}$;
    for each $C_i \in$ Є do
        $W_{LOC}$ ← read documents that reference $C_i$ to create a local word set, where
                the local support $supp_{LOC}$ is associated with each word $u_h$ in $W_{LOC}$;
        for each word $u_h \in W_{LOC}$ do
            contribution ← $log(((u_h.supp_{LOC} / u_h.supp_{GLO}) + d) / ((|C_i| -$
                $u_h.supp_{LOC}) / (|Đ| - u_h.supp_{GLO}) + d))$;
            if (contribution $\geq G$) then
                add $u_h$ into $S_{KW}$;
        end for
    end for
    return ($S_{KW}$);
End Algorithm
```

4.3.4 Method 4: DIA Association Factor based Galavotti·Sebastiani·Simi (DIAAF-based-GSS)

Another proposed statistics based feature selection technique is DIAAF-based-GSS (Darmstadt Indexing Approach Association Factor based Galavotti·Sebastiani·Simi). DIAAF-based-GSS is based on the GSS approach described in section 3.3.2. This new technique incorporates DIAAF into the original GSS approach and utilises the binary vector representation of a documentbase.

Recall that the formula for calculating GSS (see section 3.3.2) is given by:

$$gss\_score(u_h, C_i) = P(u_h, C_i) \times P(\neg u_h, \neg C_i) - P(u_h, \neg C_i) \times P(\neg u_h, C_i)\ .$$

Substituting each probabilistic component in GSS by its DIAAF related function, a DIAAF based formula is derived in a GSS fashion:

$$\begin{aligned} diaaf\text{-}based\text{-}gss\_score(u_h, C_i) &= P(C_i \mid u_h) \times P(\neg C_i \mid \neg u_h) \\ &- P(\neg C_i \mid u_h) \times P(C_i \mid \neg u_h)\ . \end{aligned}$$

It can be further expanded as:

$$\begin{aligned} diaaf\text{-}based\text{-}gss\_score(u_h, C_i) &= (count(u_h \in C_i) / count(u_h \in Đ)) \\ &\times (count(\neg u_h \in (Đ - C_i)) / count(\neg u_h \in Đ)) \\ &- (count(u_h \in (Đ - C_i)) / count(u_h \in Đ)) \\ &\times (count(\neg u_h \in C_i) / count(\neg u_h \in Đ))\ . \end{aligned}$$

The intuition behind the DIAAF-based-GSS approach is:

1. The contribution of term u_h for class C_i tends to be high if the ratio of the class based term support to the documentbase term support is high.
2. The contribution of term u_h for class C_i tends to be high if the ratio of the class-complement based term support of non-appearance to the documentbase term support of non-appearance is high.
3. The contribution of term u_h for class C_i tends to be high if the ratio of the class-complement based term support to the documentbase term support is low.
4. The contribution of term u_h for class C_i tends to be high if the ratio of the class based term support of non-appearance to the documentbase term support of non-appearance is low.

Example 4.5: DIAAF-based-GSS Score Calculation

Given a documentbase *Đ* containing 100 documents equally divided into 4 classes (i.e. 25 per class), and assuming that word u_h appears in 30 of the documents, then the DIAAF-based-GSS score per class can be calculated as shown in Table 4.8.

Class	**# docs per class**	**# docs with u_h per class**	**# docs without u_h per class**	**# docs with u_h in other classes**	**# docs without u_h in other classes**	**# docs with u_h in Đ**	**# docs without u_h in Đ**	**DIAAF -based-GSS Score**
1	25	15	10	15	60	30	70	0.357
2	25	10	15	20	55	30	70	0.119
3	25	5	20	25	50	30	70	-0.119
4	25	0	25	30	45	30	70	-0.357

Table 4.8: DIAAF-based-GSS score calculation

The algorithm for identifying keywords in *Đ*, based on DIAAF-based-GSS, is given as follows:

Algorithm 4.4: Keyword Identification — DIAAF-based-GSS

```
Input: (a) A documentbase Đ (the training part, where the noise words have been
               removed);
       (b) A user-defined significance threshold $G$;
Output: A set of identified keywords $S_{KW}$;
Begin Algorithm:
(1)   $S_{KW}$ ← an empty set for holding the identified keywords in Đ;
(2)   Є ← catch the set of pre-defined text-classes within Đ;
(3)   $W_{GLO}$ ← read Đ to create a global word set, where the word documentbase
               support $supp_{GLO}$ is associated with each word $u_h$ in $W_{GLO}$;
(4)   for each $C_i \in$ Є do
(5)       $W_{LOC}$ ← read documents that reference $C_i$ to create a local word set,
               where the local support $supp_{LOC}$ is associated with each word
               $u_h$ in $W_{LOC}$;
(6)       for each word $u_h \in W_{LOC}$ do
(7)           contribution ← $(u_h.supp_{LOC} / u_h.supp_{GLO}) \times (((|Đ| - |C_i|) -$
                  $(u_h.supp_{GLO} - u_h.supp_{LOC})) / (|Đ| - u_h.supp_{GLO})) -$
                  $((u_h.supp_{GLO} - u_h.supp_{LOC}) / u_h.supp_{GLO}) \times$
                  $((|C_i| - u_h.supp_{LOC}) / (|Đ| - u_h.supp_{GLO}))$;
(8)           if (contribution $\geq G$) then
(9)               add $u_h$ into $S_{KW}$;
(10)      end for
(11)  end for
(12)  return ($S_{KW}$);
End Algorithm
```

4.3.5 Significant Word based Documentbase Pre-processing

A given documentbase can be language-independently pre-processed by combining one proposed keyword (potential significant word) selection method (LTGSR, LTGFR, DIAAF-based-RS or DIAAF-based-GSS), and one proposed potential significant word list construction strategy (either "all words" or "unique"), with one proposed final significant word selection strategy (either "top *K*" or "dist"). This significant word based documentbase pre-processing approach thus comprises two phases: (i) processing of the training set to identify significant words, and (ii) processing of the test set to recast it in terms of the identified significant words. The generation of a text classifier (once significant words have been identified) can be undertaken using any classification algorithm although the TFPC CARM technique is chosen for this thesis. Each phase in this documentbase pre-processing approach is described as follows.

- **Processing of the Training Set:** Phase one (Algorithm 4.5) takes the training set of a documentbase $Đ_R$ as the input, and applies one of the Algorithms 4.1, 4.2, 4.3 and 4.4 to generate a set of potential significant words from $Đ_R$. The generated potential significant words are listed based on either the "all words" or the "unique" strategy. A final list of significant words is chosen, based on either the "top *K*" or the "dist" strategy, from the constructed potential significant word list. Next $Đ_R$ is processed again, with reference to the final list of significant words, to build a vector of vectors data structure, where each primary vector represents a document in $Đ_R$, and each vector element represents an individual significant word that appears in this document. By convention the class attribute of each document is stored as the last vector element.

- **Processing of the Test Set:** Phase two takes the test set of a documentbase $Đ_E$ as the input. The general algorithm is presented in Algorithm 4.6. With reference to the identified final list of significant words (from the previous phase), $Đ_E$ is read document by document: for each significant word in the final list, if it appears in the current document, this significant word is added to a created vector that represents this test document. Finally all of the created document vectors are accumulated to build a vector of vectors data structure that represents the given test set as a "bag of (key) words".

Algorithm 4.5: Significant Word based Documentbase Pre-processing — Phase 1: Processing the Training Set

```
Input: The training set of a documentbase $Đ_R$;
Output: (a) A created training data vector (vector of vectors) $DV_R$;
        (b) A set of identified significant words $S_{GW}$;
Begin Algorithm:
    $S_{GW}$ ← an empty set for holding the identified significant words in $Đ_R$;
    $S_{KW}$ ← an empty set for holding the identified potential significant
            words in $Đ_R$;
    $DV_R$ ← a training data vector of vectors (size equals to $|Đ_R|$, and each
            primary vector represents a document in $Đ_R$);
    $S_{KW}$ ← generate a set of potential significant words from $Đ_R$; // an
            algorithm from the Algorithms 4.1, 4.2, 4.3 and 4.4 is employed
    $S_{KW}$ ← refine this set; // either "all words" or "unique" is employed
    $S_{GW}$ ← select the final significant words from $S_{KW}$; // either "top K" or
            "dist" is employed
    for each document $D_j \in Đ_R$ do
        $V_R$ ← an empty significant word based document vector;
        for each element $e_k \in S_{GW}$ do
            if ($e_k$ appears in $D_j$) and (the class contributed by $e_k$ equals to
                    the class attribute of $D_j$) then
                add $e_k$ into $V_R$;
        end for
        add (the class attribute of $D_j$) into $V_R$;
        add $V_R$ into $DV_R$;
    end for
    return ($DV_R$, $S_{GW}$);
End Algorithm
```

Algorithm 4.6: Significant Word based Documentbase Pre-processing — Phase 2: Processing the Test Set

```
Input: (a) The test set of a documentbase $Đ_E$;
       (b) A set of identified significant words $S_{GW}$ (from phase 1);
Output: A created test data vector (vector of vectors) $DV_E$;
Begin Algorithm:
    $DV_E$ ← a test data vector of vectors (size equals to $|Đ_E|$, and each primary
            vector represents a document in $Đ_E$);
    for each document $D_j \in Đ_E$ do
        $V_E$ ← an empty significant word based document vector;
        for each element $e_k \in S_{GW}$ do
            if ($e_k$ appears in $D_j$) then
                add $e_k$ into $V_E$;
        end for
        add $V_E$ into $DV_E$;
    end for
    return ($DV_E$);
End Algorithm
```

4.4 Language-independent Significant Phrase Identification

Whichever of the significant word identification methods described above (in a combination of four keyword selection techniques, two potential significant word list construction strategies and two final significant word selection strategies) is selected, we define five different categories of word:

- **Upper Noise Words (UNW):** Words whose support is above a user-defined UNT (Upper Noise Threshold).
- **Lower Noise Words (LNW):** Words whose support is below a user-defined LNT (Lower Noise Threshold).
- **Significant Words (G):** Selected key words that are expected to serve to distinguish between classes.
- **Ordinary Words (O):** Other non-noise words that have not been selected as significant words.
- **Stop Marks (S):** Not actual words but six (key punctuation marks) non-alphabetic characters (*, . : ; !* and *?*) referred to as *delimiters*, and used in phrase identification. All other non-alphabetic characters are ignored.

It also identifies two groups of categories of words:

1. **Non-noise Words:** The union of significant and ordinary words.
2. **Noise Words (N):** The union of upper and lower noise words.

4.4.1 Significant Phrase Identification Strategies

Significant phrases are defined as sequences of words that include at least one significant word. Four different schemes for defining phrases are distinguished below, depending on: (i) what are used as *delimiters* and (ii) what the *contents* of the phrase should be made up of:

- **DelSNcontGO:** Phrases are delimited by stop marks (S) and/or noise words (N), and made up of sequences of one or more significant words (G) and ordinary words (O). Sequences of ordinary words delimited by stop marks and/or noise words that do not include at least one significant word are ignored. The rationale for DelSNcontGO is that if a word is significant for a class, then words which are directly adjacent to those words and are not so common/rare as to be noise words are perhaps also related to the class. For example “data” is not likely to be significant in distinguishing machine learning documents from a dataset of computer science articles, however “mining” may be and “data mining” should certainly be.

- **DelSNcontGW:** As DelSNcontGO but replacing ordinary words in phrases by wild card symbols (W) that can be matched to any single word. The idea here is that much more generic phrases are generated.
- **DelSOcontGN:** Phrases are delimited by stop marks (S) and/or ordinary words (O), and made up of sequences of one or more significant words (G) and noise words (N). Sequences of noise words delimited by stop marks and/or ordinary words that do not include at least one significant word are ignored. The rationale for DelSOcontGN is that there are many noise words that are used to link important words into a short, significant phrase. For example, “King” and “Spain” may be significant separately, but “King of Spain” as a phrase may be a better indicator even though “of” as an individual word is considered to be a noise word.
- **DelSOcontGW:** As DelSOcontGN but replacing noise words in phrases by wild card characters (W). Again the idea here is to produce generic phrases.

Example 4.6: Significant Phrase Identification Process

An example will help to illustrate the consequences of each method. Table 4.9 shows a document taken from the well-known Usenet (20 Newsgroup) collection [Lang, 1995] (with some proper names changed for ethical reasons). Note that the first line is the class label and plays no part in the phrase generation process. The first stage in pre-processing replaces all stop marks by a ‘‖’ character and removes all other non-alphabetic characters (Table 4.10). In Table 4.11 the document is shown “marked up” after the significant word identification has been completed.

Significant words are highlighted in the document (*abc…*), upper noise words are over-lined (*abc…*), and lower noise words are underlined (*abc…*), all other words are ordinary words. The "*@Class rec.motorcycles*" is a header indicating the class of this document.

@Class rec.motorcycles
paint jobs in the uk
can anyone recommend a good place for reasonably
priced bike paint jobs, preferably but not
essentially in the london area.
thanks
john somename.
–
acme technologies ltd xy house,
147 somewherex road

Table 4.9: Example document in its unprocessed form

paint jobs in the uk can anyone recommend a good place
for reasonably priced bike paint jobs ‖ preferably but not
essentially in the london area ‖ thanks john somename ‖
acme technologies ltd xy house ‖ somewherex road

Table 4.10: Example document with stop marks indicated by a '‖'
and non-alphabetic characters removed

paint jobs in the uk can anyone recommend a good place
for reasonably priced bike paint jobs ‖ preferably but not
essentially in the london area ‖ thanks john somename ‖
acme technologies ltd xy house ‖ somewherex road

Table 4.11: Example document with lower, upper and significant
words marked (all other words are ordinary words)

Phrase Identification Algorithms	**Example of Phrase Representation (Attributes)**
DelSNcontGO	{{*road*}, {*preferably*}, {*reasonably priced bike paint jobs*}, {*acme technologies ltd*}}
DelSNcontGW	{{*road*}, {*preferably*}, {Δ Δ *bike* Δ Δ}, {Δ *technologies* Δ}}
DelSOcontGN	{{*somewherex road*}, {*preferably but not*}, {*bike*}, {*technologies*}}
DelSOcontGW	{{Δ *road*}, {*preferably* Δ Δ}, {*bike*}, {*technologies*}}

Table 4.12: Example phrases (attributes) generated for the example document given in Table 4.9 using the four advocated phrase identification strategies

Table 4.12 shows the significant phrases used to represent the example document shown in Table 4.9 using each of the four different phrase identification algorithms, where appropriate "wild card" words are indicated by a "Δ" symbol. Note that a phrase can comprise any number of words, unlike (word based) 𝒩-gram approaches where phrases are a fixed length. Note also that in the context described here, the phrases identified in a document become the attributes used to describe it with respect to the classification process.

4.4.2 Significant Phrase based Documentbase Pre-processing

The generic approach of significant phrase based documentbase pre-processing is similar to the significant word based documentbase pre-processing approach (see section 4.3.5), which comprises two phases: (i) processing of the training set to identify significant phrases, and (ii) processing of the test set to recast it in terms of the identified phrases. Again, the generation of a text classifier (once significant phrases have been identified) can be undertaken using any classification algorithm, although the TFPC CARM technique is used in this thesis. Each phase in this documentbase pre-processing approach is described as follows.

- **Processing of the Training Set:** Phase one is compatible with any of the phrase mining schemes identified in section 4.4.1. The general algorithm is presented in Algorithm 4.7. This algorithm takes the training set of a documentbase $Đ_R$ as the input, and builds a binary tree data structure to store all single words (that appear in $Đ_R$) with their documentbase (global) support count. Hence the lower and upper noise words can be marked in the word binary tree, based on their support value. One of the proposed Algorithms 4.1, 4.2, 4.3 and 4.4, combined with a potential significant word list construction method (either "all words" or "unique") and a final significant word selection strategy (either "top *K*" or "dist") can then be used to identify the significant words (a word that significantly contributes to exactly or at least one given category) from non-noise words (noted as ordinary words) in the word binary tree. Here, each significant word is marked with its associated class(es). $Đ_R$ is processed again, with reference to the word binary tree, to build another binary tree data structure, in which all significant phrases that are mined from $Đ_R$ (based on one of the four proposed phrase mining schemes), are stored. Here, each phrase is recorded with respect to the documents in which they are contained. Finally a vector of vectors data structure is generated regarding the phrase binary tree, where each primary vector represents a document in $Đ_R$, and each vector element represents an individual significant phrase that appears in this document. By convention the class attribute of each document is stored as the last vector element.

- **Processing of the Test Set:** Once the phrases have been identified in the training set (i.e. the phrase binary tree has been built), the recasting of the test set is straightforward (Algorithm 4.8). This procedure takes the test set and the constructed phrase binary tree as input. It reads the test set document by document: for each node in the phrase binary tree, if it appears in the current document, this phrase will be added, as an element, to the created document vector. Finally all document vectors are accumulated to generate a vector of vectors data structure that represents the given test set as a "bag of phrases".

Algorithm 4.7: Significant Phrase based Documentbase Pre-processing — Phase 1: Processing the Training Set

```
Input: The training set of a documentbase $Đ_R$;
Output: (a) A created training data vector (vector of vectors) $DV_R$;
        (b) A constructed phrase binary tree $TP$;
Begin Algorithm:
(1)   $TW \leftarrow$ an empty word binary tree;
(2)   $TP \leftarrow$ an empty phrase binary tree;
(3)   $DV_R \leftarrow$ a training data vector of vectors (size equals to $|Đ_R|$, and each
          primary vector represents a document in $Đ_R$);
(4)   for each document $D_j \in Đ_R$ do
(5)       read $D_j$ and for each word $u_h \in D_j$ do
(6)           if $u_h$ is not already in $TW$ then
(7)               create a new $TW$ node and add it to $TW$ (with
                      support = 1);
(8)           else
(9)               if $u_h$ first-time occurs in current document then
(10)                  increment support by 1;
(11)      end for
(12)  end for
(13)  identify words in $TW$ in different types (below/above lower/upper noise
          threshold, significant to exactly one or more class(es), and words
          that do not serve to distinguish between classes);
(14)  process $Đ_R$ with reference to $TW$ (based on a proposed phrase mining
          scheme) to identify significant phrases in $Đ_R$, and place identified
          phrases into $TP$;
(15)  for each document $D_j \in Đ_R$ do
(16)      $V_R \leftarrow$ an empty significant phrase based document vector;
(17)      for each element $e_k \in TP$ do
(18)          if ($e_k$ appears in $D_j$) and (the class contributed by $e_k$ equals to
                  the class attribute of $D_j$) then
(19)              add $e_k$ into $V_R$;
(20)      end for
(21)      add (the class attribute of $D_j$) into $V_R$;
(22)      add $V_R$ into $DV_R$;
(23)  end for
(24)  return ($DV_R$, $TP$);
End Algorithm
```

Note: In line (13), an algorithm from the Algorithms 4.1, 4.2, 4.3 and 4.4 is employed; one of the potential significant word list construction strategies (either "all words" or "unique") is employed; and one of the final significant word selection strategies (either "top K" or "dist") is employed.

Algorithm 4.8: Significant Phrase based Documentbase Pre-processing — Phase 2: Processing the Test Set

Input: (a) The test set of a documentbase $Đ_E$;
(b) A constructed phrase binary tree TP (from Phase 1);

Output: A created test data vector (vector of vectors) DV_E;

```
Begin Algorithm:
    $DV_E \leftarrow$ a test data vector of vectors (size equals to $|Đ_E|$, and each primary
        vector represents a document in $Đ_E$);
    for each document $D_j \in Đ_E$ do
        $V_E \leftarrow$ an empty significant phrase based document vector;
        for each element $e_k \in TP$ do
            if ($e_k$ appears in $D_j$) then
                add $e_k$ into $V_E$;
        end for
        add $V_E$ into $DV_E$;
    end for
    return ($DV_E$);
End Algorithm
```

4.5 Summary

In this chapter, a number of documentbase pre-processing approaches were presented for single-label multi-class TC that operate in a language-independent manner. The language-independent identification of noise words was described in section 4.2. In section 4.3 four language-independent keyword selection methods were introduced; coupled with two potential significant word list construction strategies ("all words" with appropriate level of contribution, or "unique" words only), and two final significant word selection strategies ("top *K*" versus "dist — top *K* / |Є| for each class"). Section 4.4 proposed four language-independent significant phrase identification mechanisms, where each represents a particular pattern of combining upper/lower noise words, significant words, non-significant and non-noise (ordinary) words, and key punctuation (stop) marks. In summary 16 different significant word identification schemes (4 keyword selection methods × 2 potential significant word list construction strategies × 2 final significant word selection strategies), and 64 distinct significant phrase identification schemes (16 significant word identification schemes × 4 phrase identification mechanisms) were postulated in this chapter. In the next chapter, an experimental evaluation of these schemes will be described.

Chapter 5

Experiments and Results

5.1 Introduction

This chapter presents an evaluation of the proposed language-independent documentbase pre-processing approaches, using three well-known English text collections (Usenet Articles[4], Reuters-21578[5] and MedLine-OHSUMED[6]) and a popular Chinese textual data set (Chinese Text Classification Corpus[7]). The aim of this evaluation is to assess the approaches with respect to both the accuracy of classification and the efficiency of computation. All evaluations described in this chapter were conducted using the TFPC CARM algorithm (as previously described in section 2.6.4), coupled with the Best First Rule CSRS (Case Satisfaction and Rule Selection) approach (see section 2.6.2) and the CSA rule ordering strategy (see section 2.6.3); although any other classifier generator could equally well have been used. All algorithms were implemented using the standard Java programming language. Experiments were run on a 1.86 GHz Intel(R) Core(TM)2 CPU with 1.00 GB of RAM running under Windows Command Processor. Note that some of the experimental results presented in this chapter have been previously reported in [Wang *et al.*, 2006] and [Coenen *et al.*, 2007].

The organisation of this chapter is as follows. The following section describes the textual data sets (documentbases) that were used in the experiments. Sections 5.3, 5.4, 5.5 and 5.6 describe four groups of experiments:

- **Group 1: Threshold Testing** (Table 5.1): Three sub-groups (sets) of experiments which demonstrate the effects (with respect to classification accuracy) of changing the values for the thresholds of significance, support and

[4] http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes/20_newsgroups.tar.gz
[5] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[6] http://trec.nist.gov/data/filtering/
[7] http://www.nlp.org.cn/

confidence, in order to identify appropriate values for different parameters/thresholds of the proposed approaches.

- **Group 2: Documentbase Pre-processing Strategy Evaluation** (Table 5.2): To evaluate: (i) the performance of the phrase generation mechanisms in respect of both the accuracy of classification and the efficiency of computation, and (ii) the classification accuracy of the strategies for the potential significant word list construction and the final significant word selection.
- **Group 3: Keyword Selection Method Evaluation** (Table 5.3): Evaluation of the performance of each proposed keyword selection method (in terms of classification accuracy) when directly applied using the language-independent "bag of words" approach or when used in a language-independent "bag of phrases" setting. These experiments also compare the performance of the "bag of phrases" approach versus the "bag of words" approach.
- **Group 4: Chinese Data Set Experiments** (Table 5.4): To examine both the classification accuracy and the process efficiency for the proposed language-independent documentbase pre-processing approaches when dealing with a non-English language textual data set.

A summary is presented in section 5.7.

No.	Experiment Title	Documentbase	Objective
1.1	Effect of Changing the Significance Threshold	NGA.D10000.C10	To demonstrate that the significance threshold (based on the LTGSR keyword selection method) should not be valued too high.
1.2		NGB.D9997.C10	
1.3		Reuters.D6643.C8	
1.4	Effect of Changing the Support Threshold	NGA.D10000.C10	To demonstrate that the support threshold should be valued quite low.
1.5		NGB.D9997.C10	
1.6		Reuters.D6643.C8	
1.7	Effect of Changing the Confidence Threshold	NGA.D10000.C10	To demonstrate that the confidence threshold value should be relatively low, say at most at 50%.
1.8		NGB.D9997.C10	
1.9		Reuters.D6643.C8	
1.10	Effect of Changing UNT with Low Support Threshold and 150 selected Final Significant Words per Class	NGA.D10000.C10	To demonstrate that (i) the UNT should be set higher than the default value of 3.5%; (ii) the support threshold should be valued even lower (i.e. approaching 0); and (iii) the maximum number of selected final significant words should be set at 150 per class.

Table 5.1: List of experiments described in experiment group 1: Threshold testing

No.	Experiment Title	Documentbase	Objective
2.1	Number of Attributes	NGA.D10000.C10	To show (i) the number of attributes, (ii) the accuracy of classification (in percentage), (iii) the number of empty training documents, and (iv) the execution times (in seconds); when processing alternative language-independent documentbase pre-processing techniques based on the determined most appropriate (from experiment group 1) threshold/parameter values.
2.2	Classification Accuracy in Percentage		
2.3	Number of Empty Documents in the Training Data Set		
2.4	Execution Times in Seconds		
2.5	Relationship between the Significance Threshold and the Number of identified Significant Words		To demonstrate the relationship between the significance threshold and (i) the number of identified significant words or (ii) the number of generated empty training documents, for both proposed keyword selection techniques of LTGFR and LTGSR.
2.6	Relationship between the Significance Threshold and the Number of generated Empty Documents in the Training Data Set		

Table 5.2: List of experiments described in experiment group 2: Documentbase pre-processing strategy evaluation

No.	Experiment Title	Documentbase	Objective
3.1	Classification Accuracy obtained when varying both the Support and Confidence Thresholds	NGA.D10000.C10	For each English language documentbase, finding the appropriate values for both the support and the confidence thresholds, which produce the best classification accuracy when directly applying DIAAF as a "bag of words".
3.2		NGB.D9997.C10	
3.3		Reuters.D6643.C8	
3.4		OHSUMED.D6855.C10	
3.5	Comparison of Keyword Selection Techniques in "Bag of Words"	NGA.D10000.C10	For each English language documentbase, finding the best performing keyword selection technique (with respect to classification accuracy) when directly applying alternative techniques as a language-independent "bag of words".
3.6		NGB.D9997.C10	
3.7		Reuters.D6643.C8	
3.8		OHSUMED.D6855.C10	
3.9	Comparison of Keyword Selection Techniques in "Bag of Phrases"	NGA.D10000.C10	For each English language documentbase, finding the best performing keyword selection technique (with respect to classification accuracy) when using alternative techniques in DelSNcontGO (suggested by experiment group 2) based language-independent "bag of phrases".
3.10		NGB.D9997.C10	
3.11		Reuters.D6643.C8	
3.12		OHSUMED.D6855.C10	
3.13	Comparison of the "Bag of Phrases" Approach and the "Bag of Words" Approach	All English language documentbases	To find the best performing (with respect to classification accuracy) approach in between of "bag of phrases" and "bag of words". Note that results of experiments no. 3.5 ~ 3.12 are utilised.

Table 5.3: List of experiments described in experiment group 3: Keyword selection method evaluation

No.	Experiment Title	Documentbase	Objective
4.1	Classification Accuracy obtained by Varying both the Support and Confidence Thresholds	Chinese.D2816.C10	To find the appropriate values for both the support and the confidence thresholds, which produce the best classification accuracy when directly applying DIAAF as a "bag of words".
4.2	Comparison of Keyword Selection Techniques in "Bag of Words"		To find the best performing keyword selection technique (with respect to classification accuracy) when directly applying alternative techniques as a language-independent "bag of words".
4.3	Comparison of Keyword Selection Techniques in "Bag of Phrases"		To find the best performing keyword selection technique (with respect to classification accuracy) when using alternative techniques in DelSNcontGO (suggested by experiment group 2) based language-independent "bag of phrases".
4.4	Comparison of the "Bag of Phrases" Approach and the "Bag of Words" Approach		To find the best performing (with respect to classification accuracy) approach between "bag of phrases" and "bag of words". Note that results in experiments no. 4.2 and 4.3 are utilised.
4.5	Execution Times in Seconds		To show the execution times (in seconds) when processing alternative language-independent documentbase pre-processing approaches.

Table 5.4: List of experiments described in experiment group 4: Chinese data set experiments

5.2 Experimental Data Description

For the experiments outlined in the following sections, five individual documentbases (textual data sets) were used. Each was extracted (as a subset) from one of the following text collections:

1. Usenet Articles,

2. Reuters-21578,

3. MedLine-OHSUMED, and

4. Chinese Text Classification Corpus.

In this section these documentbases are introduced in detail.

5.2.1 Usenet Articles

The "Usenet Articles" collection is a well-known English language text collection. It was compiled by Lang [1995] from 20 different newsgroups and is sometimes referred to as the "20 Newsgroups" collection. Each newsgroup represents a pre-defined class. There are exactly 1,000 documents per class with one exception — the class "soc.religion.christian" that contains 997 documents only. In comparison with other common English text collections, the structure of the "20 Newsgroups" collection is relatively *"neat"* — every document within this collection is labelled with one class only and almost all documents have a "proper" text-content. In the context of this thesis a proper text-content document is one that contains at least q recognised words (see section 3.2). The value of q is usually small (q is set to be 20 in our study). Previous TC studies have used this text collection in various ways. For example:

- In [Deng *et al.*, 2002] the entire "20 Newsgroups" was randomly divided into two non-overlapping and (almost) equally sized documentbases covering 10 classes each: *20NG.D10000.C10* and *20NG.D9997.C10*.
- In [Wu *et al.*, 2002] four smaller documentbases were extracted from the collection and used in evaluations.

In this thesis we adopted the approach of Deng *et al.* [2002]. The entire collection was randomly split into two documentbases covering 10 classes each: *NGA.D10000.C10* and *NGB.D9997.C10*. Table 5.5 shows the detail of each documentbase.

NGA.D10000.C10		**NGB.D9997.C10**	
Class	**# of docs**	**Class**	**# of docs**
comp.windows.x	1,000	comp.graphics	1,000
rec.motorcycles	1,000	comp.sys.mac.hardware	1,000
talk.religion.misc	1,000	rec.sport.hockey	1,000
sci.electronics	1,000	sci.crypt	1,000
alt.atheism	1,000	sci.space	1,000
misc.forsale	1,000	talk.politics.guns	1,000
sci.med	1,000	comp.os.ms-windows.misc	1,000
talk.politics.mideast	1,000	rec.autos	1,000
comp.sys.ibm.pc.hardware	1,000	talk.politics.misc	1,000
rec.sport.baseball	1,000	soc.religion.christian	997

Table 5.5: Documentbase description (NGA.D10000.C10 & NGB.D9997.C10)

5.2.2 Reuters-21578

Reuters-21578 is another popular English language text collection widely applied in text mining investigations. It comprises 21,578 documents collected from the Reuters newswire service with 135 pre-defined classes. Within the entire collection, 13,476 documents are labelled with at least one class, 7,059 are not marked with any class and 1,043 have their class-label as "*bypass*" (which, at least in this study, is not considered to be a proper class-label). Within the 13,476 classified documents, 2,308 appear to have a class but on further investigation that class turns out to be spurious. This leaves 11,168 documents, of which 9,338 are single-labelled and 1,830 are multi-labelled.

There are in total 135 classes. However, many TC studies (see for example [Li and Liu, 2003], [Zaïane and Antonie, 2002]) have used only the 10 most populous classes for their experiments and evaluations. There are 68 classes that consist of fewer than 10 documents, and many others consist of fewer than 100 documents. The extracted documentbase, suggested in [Li and Liu, 2003] and [Zaïane and Antonie, 2002], can be referred to as *Reuters.D10247.C10* and comprises 10,247 documents with 10 classes. However *Reuters.D10247.C10* includes multi-labelled documents that are inappropriate for a single-label TC investigation (the approach adopted here).

Deng *et al.* [2002] introduce the *Reuters_100* documentbase that comprises 8,786 documents with 10 classes. Deng *et al.* assign "*one document (to) one category and adopt categories that contain training documents (of) more than 100*". Unfortunately which 10 of the 135 classes were chosen was not specified, but it can be assumed that they are close to or identical with the classes included in *Reuters.D10247.C10* where many documents were in fact found without a proper text-content. Filtering away such non-text documents from the extracted Reuters-21578 based documentbase is suggested. It ensures that documentbase quality is maintained and means that the application of various feature selection approaches described here are more reliable.

In this thesis, the preparation of a Reuters-21578 based documentbase consisted of two stages: (i) identification of the top-10 populous classes, as in [Li and Liu, 2003] and [Zaïane and Antonie, 2002]; and (ii) removal of multi-labelled and/or non-text documents from each class. As a consequence class "wheat" had only one "qualified" document, and no document was contained in class "corn". Hence, the final documentbase, namely *Reuters.D6643.C8*, omitted these classes of "wheat" and "corn", leaving a total of 6,643 documents in 8 classes. A description of this documentbase is given in Table 5.6.

Class	# of docs	Class	# of docs
acq	2,108	interest	216
crude	444	money	432
earn	2,736	ship	174
grain	108	trade	425

Table 5.6: Documentbase description (Reuters.D6643.C8)

5.2.3 MedLine-OHSUMED

The MedLine-OHSUMED (English language) text collection, collected by Hersh *et al.* [1994], consists of 348,566 records relating to 14,631 pre-defined MeSH (Medical Subject Headings) categories. The OHSUMED collection accounts for a subset of the MedLine text collection[8] for 1987 to 1991. Characteristics of OHSUMED include: (1) many multi-labelled documents; (2) the 14,631 classes are arranged in a hierarchy (e.g. classes "male" and "female" are subclasses of the class "human"; classes "adult" and "child" are subclasses of "male" and/or "female", etc.); and (3) the text-content of each document comprises either a title on its own (without a text-content), or a "title-plus-abstract" (with a text-content) from various medical journals.

With the goal of investigating the multi-label TC problem, Joachims [1998] uses the first 10,000 "title-plus-abstract" texts of the 50,216 documents for 1991 as a training set, and the second 10,000 such documents as a test set. This defines the *OHSUMED.D20000.C23* documentbase, in which the classes are 23 MeSH "diseases" categories. Since each record within this documentbase may be labelled with more than one class, it does not satisfy our requirements. This is also the case for the *OHSUMED.maximal* [Zaïane and Antonie, 2002], which consists of all OHSUMED classes incorporating all 233,445 "title-plus-abstract" documents.

[8] http://medline.cos.com/

The process of extracting a documentbase from MedLine-OHSUMED in our study can be detailed as follows. First, the top-100 most populous classes were identified in the collection. These included many super-and-sub class-relationships. Due to the difficulty of obtaining a precise description of all the possible taxonomy-like class-relationships, we simply selected 10 target-classes from these classes by hand, so as to exclude obvious super-and-sub class-relationships. Documents that are either multi-labelled or without a proper text-content (containing < q recognised words) were then removed from each class. Finally a documentbase, namely *OHSUMED.D6855.C10*, was created that includes 6,855 documents in total. In Table 5.7, the description of this documentbase is provided.

Class	**# of docs**	**Class**	**# of docs**
amino_acid_sequence	333	kidney	871
blood_pressure	635	rats	1,596
body_weight	192	smoking	222
brain	667	tomography,_x-ray_computed	657
dna	944	united_states	738

Table 5.7: Documentbase description (OHSUMED.D6685.C10)

5.2.4 Chinese Text Classification Corpus

The Chinese Text Classification Corpus is a popular Chinese text collection, compiled by Ronglu Li that is published on the website http://www.nlp.org.cn. The usage of this text collection has been considered in many TC and/or TC related studies, e.g. [Wang and Wang, 2005]. It comprises 2,816 documents in 10 pre-defined classes (see Table 5.8).

Class	**# of docs**	**Class**	**# of docs**
交通 (transports)	214	教育 (education)	220
体育 (sports)	450	环境 (environments)	201
军事 (military)	249	经济 (economics)	325
医药 (medicine)	204	艺术 (arts)	248
政治 (politics)	505	计算机 (computers)	200

Table 5.8: Documentbase description (Chinese.D2816.C10)

The structure of this collection is considered *neat* — like the "20 Newsgroups" collection is. Each document is associated with one text-category only, and all documents have a proper text-content. Hence the entire collection was used as a documentbase (*Chinese.D2816.C10*) in our experiments. It is worth noting here that *Chinese.D2816.C10* is a relatively small data set compared to the other documentbases considered in this study.

5.2.5 Documentbase Analysis

Some statistics concerning the characteristics of the documentbases used in our experiments are presented in Table 5.9. In each case the figures were obtained from the first 9/10th of the documentbase (as the training set of data) with LNT (Lower Noise Threshold) = 1% and UNT (Upper Noise Threshold) = 50%.

Documentbase	**# words**	**# Lower Noise Words**	**# Upper Noise Words**	**# Potential Key Words**	**% Lower Noise Words**	**% Upper Noise Words**	**% Potential Key Words**
NGA.D10000.C10	49,605	47,981	21	1,603	96.73	0.04	3.23
NGB.D9997.C10	47,973	46,223	22	1,728	96.35	0.05	3.60
Reuters.D6643.C8	19,839	18,749	11	1,079	94.51	0.06	5.43
OHSUMED.D6855.C10	27,140	25,620	13	1,507	94.40	0.05	5.55
Chinese.D2816.C10	4,886	2,909	63	1,914	59.54	1.23	39.23

Table 5.9: Statistics for the five documentbases
(LNT = 1% & UNT = 50%)

From Table 5.9 it can be seen that the majority of words occur in less than 1% of documents, although there is an apparent difference between *Chinese.D2816.C10* and the other (English language) documentbases. Hence LNT must be set at a low value so as not to miss any potential significant words. Relatively few words are common, appearing in over 50% of the documents. The implication is that UNT should be set at a (relatively) low value to avoid finding a large number of non-noise words that might lead to computational constraints on the classification process.

It was decided, for reasons of computational efficiency, to limit the total number of identified attributes[9] (significant words/phrases) to 2^{15}. The LNT and UNT values must therefore be set so that the number of resulting attributes is less than 2^{15}. Figure 5.1 shows the relationship between LNT and UNT with regard to the limit on the number of identified attributes.

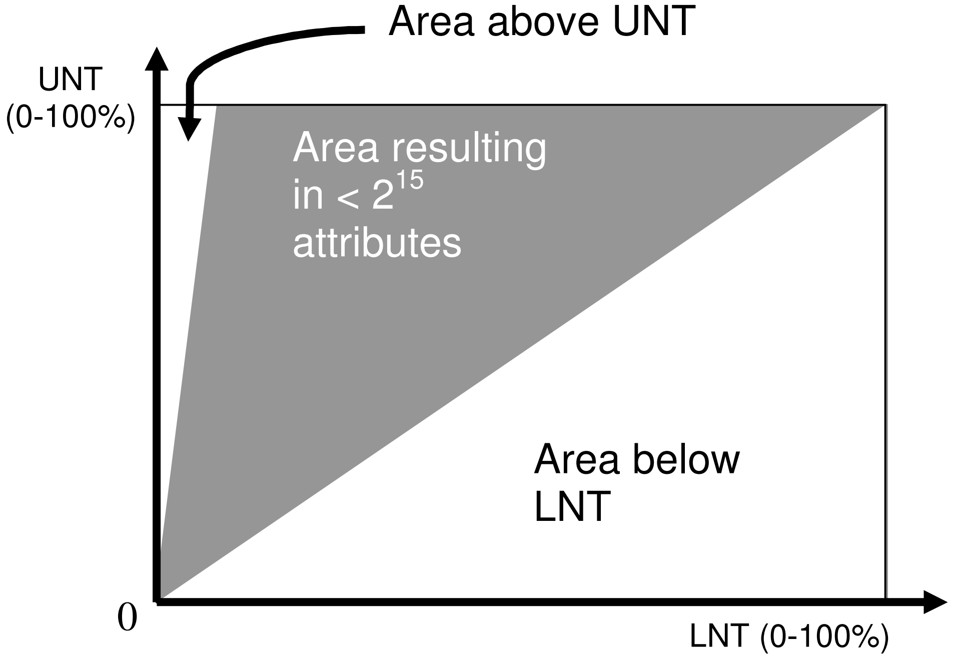


Figure 5.1: Relationship between LNT and UNT with regard to the limit of attribute generation

Tables 5.10, 5.11, 5.12, 5.13, and 5.14 show the most common (upper noise) words found in the documentbases using an UNT of 50%. In each case figures in parentheses indicate the number of documents where the word appears; recall that there are 10,000, 9,997, 6,643, 6,855 and 2,816 documents in these documentbases respectively, and that the first 9/10th of each documentbase was used as the training set. The *NGB.D9997.C10* documentbase contains just one additional common word "but", which is less common in *NGA.D10000.C10*. It is interesting to note that "mln" (abbreviation for million) and "reuter" appear as two very common words in *Reuters.D6643.C8*. In *Chinese.D2816.C10* 63 UNWs (Upper Noise Words) were found. A coarse translation (Chinese to English) for each word is provided; note that some UNWs translate to more than one English meaning.

[9] The TFPC algorithm stores attributes as a signed short integer.

a (7,666)	and (7,330)	are (4,519)	be (4,741)	for (6,367)	have (5,135)
i (6,803)	in (7,369)	is (6,677)	it (5,861)	not (4,565)	of (7,234)
on (5,075)	re (5,848)	that (6,012)	the (8,203)	this (5,045)	to (7,682)
with (4,911)	writes (4,581)	you (5,015)			

Table 5.10: Very common words (UNT = 50%) in NGA.D10000.C10

a (7,834)	and (7,406)	are (4,805)	be (5,257)	but (4,632)	for (6,399)
have (5,364)	i (6,852)	in (7,576)	is (6,858)	it (6,166)	not (4,846)
of (7,543)	on (5,506)	re (6,264)	that (6,513)	the (8,424)	this (5,331)
to (7,902)	with (4,872)	writes (4,701)	you (5,012)		

Table 5.11: Very common words (UNT = 50%) in NGB.D9997.C10

a (4,000)	and (4,326)	for (3,439)	in (4,089)	it (3,252)	mln (3,291)
of (4,953)	reuter (5,911)	said (4,295)	the (4,425)	to (4,247)	

Table 5.12: Very common words (UNT = 50%) in Reuters.D6643.C8

a (5,513)	and (6,053)	by (3,971)	for (4,041)	in (5,942)	is (3,387)
of (6,131)	that (4,246)	the (6,112)	to (5,582)	was (4,102)	were (3,815)
with (4,980)					

Table 5.13: Very common words (UNT = 50%) in OHSUMED.D6855.C10

一 (2,270) a(n); one	上 (1,889) on; up; upon; above	不 (1,534) not; no	加 (1,475) to add; plus	动 (1,287) to use; to act; to move	华 (2,184) china
发 (1,747) to send out; to issue	同 (1,418) like; same; with	后 (1,364) after; back; later	和 (2,175) and; with	到 (1,534) to; until; up to; to go	前 (1,447) before; front; ago
国 (2,094) country; nation	在 (2,317) at; in	地 (1,499) -ly; earth; place; land	多 (1,440) many; much; a lot	大 (1,915) big; large; major; great	天 (1,464) day; sky
完 (1,944) to finish; to be over; whole; complete	家 (1,489) -ist; ier; -ian; home; family	对 (1,600) for; to; pair; couple; right	将 (1,397) will; shall	年 (1,790) year	开 (1,471) start; open
成 (1,633) turn into; finish; complete; become	是 (1,925) is; are; am; yes; to be	时 (1,615) when; time; hour	月 (2,325) month; moon	有 (1,932) have; there is; there are; to be	本 (1,275) this; the current; origin; root
来 (1,634) to come	新 (2,215) new	方 (1,403) just; square	日 (2,420) day; date; sun	现 (1,273) now; current; appear; present	生 (1,333) life; to give birth; to grow
于 (1,584) in; at; to; from; by; than; out of	人 (1,940) man; person; people	今 (1,433) this; now; current; today	以 (1,822) by; with; because	了 (1,973) *complete action marker*	会 (1,726) be able to; be likely to; to meet; group
作 (1,522) to do; to make	全 (1,436) all; every; whole	内 (1,310) within; inside; internal	出 (1,703) out; to happen	分 (1,509) part; minute; to divide	中 (2,101) in; within; while; during
个 (1,612) *a measure word*; individual	为 (1,940) for; to; because of; to be; to do	主 (1,407) primary; to own	等 (1,345) etc.; and so on; same as; wait for	社 (1,931) society; group	经 (1,280) through; to undergo
者 (1,569) -ist; -er; person	行 (1,840) okay; all right; to go; to do	要 (1,416) must; to be going to; demand; request	这 (1,810) this; these	进 (1,633) to come in; enter	部 (1,464) *a measure word*; part; division; section
长 (1,355) always; constantly; forever; long; head; chief; to grow	的 (2,503) *possessive, modifying, or descriptive particle*; of; truly	电 (2,030) e-; electric; electricity; electrical			

Table 5.14: Very common words (UNT = 50%) in Chinese.D2816.C10

5.3 Experiment Group 1: Threshold Testing

In this section, we present three sets of experiments to examine the selection of appropriate values for the thresholds of significance (*G*), support (σ) and confidence (α). The objective of the experiments was to study the following expectancies:

1. **Effect of changing *G*:** The value of *G* influences the number of discriminating words discovered and the degree of the discrimination. The higher the value of *G* the greater the discrimination and the less the number of significant words identified.

2. **Effect of changing σ:** The lower the support threshold value the greater the number of frequent sets of text-attributes that will be identified.

3. **Effect of changing α:** The higher the confidence threshold value the greater the degree of evidence for the identified rules and the less the number of rules discovered.

In each case all other parameters (thresholds) remained constant while a set of values for the corresponding parameter were assigned and tested. The LNT and UNT values used were 0.5% (low) and 3.5% (relatively low) respectively. From general data mining theory and experience using the "support-confidence" framework [Delgado *et al.*, 2002] it is well-known that, in general, low support thresholds tend to produce the best results. Many published evaluations of CARM, (e.g. [Coenen and Leng, 2004], [Li *et al.*, 2001], [Yin and Han, 2003]) use a threshold of 1%.

In related work to that described here, a relationship between the selected value of support threshold (σ) and the accuracy of classification (*Accy*), when dealing with a large database (i.e. a database comprising a large number of data records), was identified: $\downarrow\sigma \Rightarrow \uparrow Accy$. This relationship was observed for values of support ranging from 1% down to 0.03%. This work has been subsequently published in [Wang *et al.*, 2007a].

All documentbases dealt with in this thesis, especially the English language documentbases, may be considered to be large databases, hence an appropriate value of σ should be lower than 1%. For a two-class problem (binary classification) it is generally acknowledged that α should be at least 50% (i.e. better than a random guess) but preferably higher. In the documentbases dealt with here we have either 8 or 10 classes therefore reasonable results may be expected with confidence thresholds set at 50% or less.

Experimental results presented in this section are based on the previously introduced LTGSR keyword selection mechanism (see section 4.3.1), where significant words are identified by the *G* value only, irrespective of the different potential significant word list construction and/or final significant word strategies chosen. All experiments here make use of three documentbases: *NGA.D10000.C10*, *NGB.D9997.C10* and *Reuters.D6643.C8*. Accuracy figures, describing the proportion of correctly classified "unseen" documents, were obtained using the Ten-fold Cross Validation (TCV) approach.

5.3.1 Effect of Changing the Significance Threshold

In this set of experiments (noted as experiments 1.1 to 1.3 in Table 5.1) the value of *G* was steadily increased from 1 up to 3 in increments of 0.5; and σ, α, LNT and UNT were kept constant at 0.5%, 35%, 0.5% and 3.5% respectively. In general, better results were obtained from lower values of *G*. A possible explanation of this is that as the value of *G* increased, the degree of discrimination attached to significant words increased, and the overall number of significant words decreased. Consequently the number of identified phrases decreased and the representation became increasingly sparse to the extent that some documents in the test data set were represented by an empty vector. On the other hand, in a number of cases, where *G* was very small, too many phrases were generated and consequently the 2^{15} attribute limit was reached.

Table 5.15 gives the best results obtained from this set of experiments. In some cases the best accuracy was obtained equally across a range of values for *G*. Overall using stop marks and noise words (DelSN strategies) as discriminators/delimiters produced better results than using stop marks and ordinary words (DelSO). This follows from the fact the latter approach generated many more phrases and the resulting representation had a tendency to result in overfitting.

Documentbase Pre-processing Strategy	**NGA.D10000.C10**		**NGB.D9997.C10**		**Reuters.D6643.C8**	
	Accy (%)	***G***	***Accy (%)***	***G***	***Accy (%)***	***G***
DelSNcontGO	64.18	1 ~ 2.5	69.35	1 ~ 2.5	73.64	1
DelSNcontGW	64.18	1 ~ 2	69.42	2.5	73.64	1
DelSOcontGN		1 ~ 3		1 ~ 3		1 ~ 3
DelSOcontGW	11.26	3	14.12	2	38.87	1.5
Keywords	67.61	1 ~ 2.5	71.70	1 ~ 2.5	74.94	1

Table 5.15: The best classification accuracy when changing values of *G* with $\sigma = 0.5\%$, $\alpha = 35\%$, LNT = 0.5%, and UNT = 3.5% constant

Using stop marks and ordinary words as discriminators without wild cards (DelSOcontGN) did not produce any results because the 2^{15} limit was reached. It was also interesting to note, in this experiment, that when using significant words on their own (referred to as *Keywords*) a slightly better accuracy was produced, suggesting that this experimental organisation and parameterisation was relatively unsuccessful in generating phrases effectively.

5.3.2 Effect of Changing the Support Threshold

The second set of experiments (noted as experiments 1.4 to 1.6 in Table 5.1) investigated the effect of keeping *G*, α, LNT and UNT constant (at 1.5, 35%, 0.5% and 3.5% respectively) while steadily decreasing σ (support threshold) by 0.05%, from 1% to 0.25%. The experiments, consistent with the general experience of Association Rule Mining, confirm that a low support threshold value produces the best results although if the threshold is too small (i.e. approaches 0) either: (i) too many phrases may result; or (ii) very specific phrases, resulting in overfitting, may be produced. Again, better results were produced using stop marks and noise words as discriminators rather than using stop marks and ordinary words. Table 5.16 gives the best results. Again there are no results for DelSOcontGN since in every case the 2^{15} limit was reached.

Documentbase Pre-processing Strategy	NGA.D10000.C10		NGB.D9997.C10		Reuters.D6643.C8	
	Accy (%)	σ (%)	*Accy* (%)	σ (%)	*Accy* (%)	σ (%)
DelSNcontGO	69.14	0.25	73.66	0.25	76.10	0.25
DelSNcontGW	69.14	0.25	73.66	0.25	76.05	0.25
DelSOcontGN		0.25 ~ 1		0.25 ~ 1		0.25 ~ 1
DelSOcontGW	15.38	0.25		0.25 ~ 1	44.56	0.25
Keywords	69.14	0.25	73.46	0.25	76.26	0.25

Table 5.16: The best classification accuracy when changing values of σ with G = 1.5, α= 35%, LNT = 0.5%, and UNT = 3.5% constant

5.3.3 Effect of Changing the Confidence Threshold

The third set of experiments (noted as experiments 1.7 to 1.9 in Table 5.1) investigated the effect on classification accuracy when G, σ, LNT and UNT are kept constant (at 1.5, 0.5%, 0.5% and 3.5% respectively) and α (confidence threshold) is varied (from 15 up to 65%). The result obtained here shows that accuracy tends to fall as confidence increases because fewer rules are generated. The best results are presented in Table 5.17. Note that again in the case of DelSOcontGN and also in the case of DelSOcontGW, with respect to the *NGB.D9997.C10* documentbase, that the 2^{15} limit is reached.

Documentbase Pre-processing Strategy	NGA.D10000.C10		NGB.D9997.C10		Reuters.D6643.C8	
	Accy (%)	α (%)	*Accy* (%)	α (%)	*Accy* (%)	α (%)
DelSNcontGO	64.95	15	70.33	15	75.91	25
DelSNcontGW	64.95	15	70.33	15	75.93	25
DelSOcontGN		15 ~ 65		15 ~ 65		15 ~ 65
DelSOcontGW	12.82	65		15 ~ 65	40.52	55
Keywords	68.06	15	72.15	15	74.87	15

Table 5.17: The best classification accuracy when changing values of α with G = 1.5, σ= 0.5%, LNT = 0.5%, and UNT = 3.5% constant

5.3.4 Discussion

The foregoing experiments examined the proposed LTGSR keyword selection method and four mechanisms based on this keyword selection technique for defining significant phrases; and studied variations in the significance, support and confidence thresholds used in the application of the TFPC CARM algorithm. In these experiments, it can be seen that the best accuracy was obtained using DelSNcontGO, DelSNcontGW and Keywords approaches; with the other two (phrase based) approaches performing poorly. In general, and consistent with general ARM experience, it was found that a low support threshold value (0.25% or maybe even lower) worked best, and also a relatively low confidence threshold value around 25% (±10%). A significance threshold between 1 and 3 is suggested. It was also found that a low LNT was beneficial to ensure that potentially significant words were not omitted.

These preliminary results were used to select suitable parameters for use in a further set of experiments that refined the approach for identifying significant words. These experiments, focussed on the DelSNcontGO algorithm, began by identifying significant words for each class, and placing these in order of their contribution to that class. The final selection of significant words was then made so

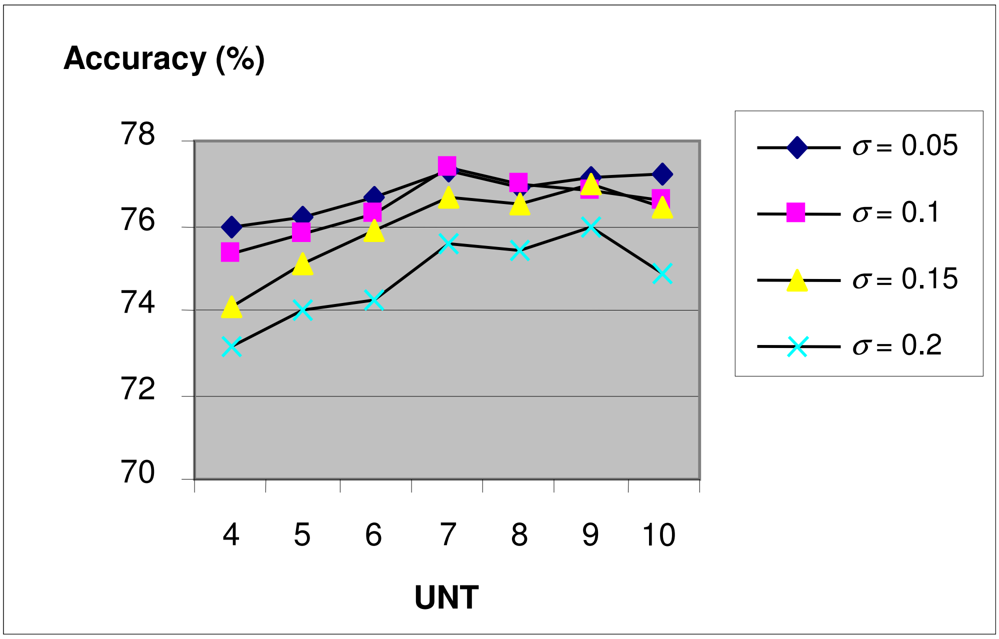


Figure 5.2: Accuracy obtained for a range of support and UNT values with $G = 3$, $\alpha = 35\%$, LNT = 0.2%, $K = 150 \times 10$ classes, and documentbase = NGA.D10000.C10

that each class has an equal number *k* (introduced as "*K* / |*C*|" in section 4.3), i.e. *k* words with the highest contribution to the class. In [Apte *et al.*, 1994] a value of 150 was suggested to be the appropriate value of *k*, so that the value of *K* (the maximum number of selected final significant words) is equal to 1,500 (150 × 10 classes). Some results (experiment 1.10 in Table 5.1) using the *NGA.D10000.C10* documentbase are shown in Figure 5.2. Best accuracy is obtained with an UNT of 7% and a support of 0.1% or 0.05%.

5.4 Experiment Group 2: Documentbase Pre-processing Strategy Evaluation

Experiments in this section were conducted using the *NGA.D10000.C10* documentbase; made use of both the proposed LTGSR and LTGFR keyword selection methods (see section 4.3.1) and investigated all combinations of the eight different significant word identification schemes (2 keyword selection methods × 2 potential significant word list construction strategies × 2 final significant word selection strategies) under the four proposed different phrase generation mechanisms (DelSNcontGO, DelSNcontGW, DelSOcontGN, and DelSOcontGW). This group of experiments also investigated the effect of using the identified significant words on their own as a "bag of words" representation.

The suite of experiments described here used the first 9/10th documentbase (9,000 documents) as the training set, and the last 1/10th (1,000 documents) as the test set (noted as the "9/10th & 1/10th" setting). For all the results presented here, the following thresholds were used: *G* (significance) = 3, σ (support) = 0.1%, α (confidence) = 35%, LNT = 0.2%, UNT = 7%, and *K* (maximum number of selected final significant words) = 1,500. These parameters produced a word distribution that is shown in Table 5.18. As would be expected the number of potential significant words is less when only unique words (unique to a single class) are selected. Note also that using LTGFR to calculate the contribution of words leads to fewer significant words being generated than is the case when using LTGSR which considers only the number of documents in which a word is encountered.

Number of Noise Words above UNT	208			
Number of Noise Words below LNT	43,681			
Number of Ordinary Words	4,207			
Number of Significant Words	1,500			
Number of Words	49,596			
	LTGFR		**LTGSR**	
	Unique	**All**	**Unique**	**All**
Number of Potential Significant Words	2,911	3,609	3,188	3,687

Table 5.18: Number of potential significant (key) words calculated per strategy for NGA.D10000.C10

Tables 5.19 and 5.20 illustrate the application of the LTGSR keyword selection mechanism to the identification of potential significant words. Table 5.19 gives the distribution of potential significant words per class for *NGA.D10000.C10* (using *G* = 3, LNT = 0.2% and UNT = 7%), and demonstrates the rationale of choosing the value of *K* as 1,500 (*k* = 150 per class). Note that the number of potential significant words per class is not balanced, with the general "forsale" class having the least number of potential significant words and the more specific "mideast" class the most.

Class Label	**# Sig. Words**	**Class Label**	**# Sig. Words**
comp.windows.x	384	rec.motorcycles	247
talk.religion.misc	357	sci.electronics	219
alt.atheism	346	misc.forsale	127
sci.med	381	talk.politics.mideast	1,091
comp.sys.ibm.pc.hardware	175	rec.sport.baseball	360

Table 5.19: Number of potential significant words in NGA.D10000.C10 using LTGSR, "all words" and "top *K*", with *G* = 3, LNT = 0.2%, and UNT = 7%

Table 5.20 shows the 10 most significant words for each class using the same keyword selection strategy and parameters/thresholds. The value shown in parentheses is the contribution of the word to the class in each case. Recall that using the LTGSR (support count based) strategy, the highest possible contribution value of the *NGA.D10000.C10* documentbase is 10, obtained when the word is unique to a certain class. In the "forsale" category, quite poor contribution values are found, while the "mideast" category has many high contribution words.

windows.x	**motorcycles**	**Religion**	**Electronics**	**Atheism**
colormap(10)	behanna(10)	ceccarelli(10)	circuits(9.8)	inimitable(10)
contrib(10)	biker(10)	kendig(10)	detectors(9.6)	mozumder(10)
imake(10)	bikers(10)	rosicrucian(10)	surges(9.5)	tammy(10)
makefile(10)	bikes(10)	atf(9.5)	ic(9.3)	wingate(10)
mehl(10)	cages(10)	mormons(9.5)	volt(9.3)	rushdie(9.8)
mwm(10)	countersteering(10)	batf(9.3)	volts(9.2)	beauchaine(9.7)
olwn(10)	ducati(10)	davidians(9.2)	ir(9.2)	benedikt(9.4)
openlook(10)	fxwg(10)	abortions(9.0)	voltage(9.2)	queens(9.4)
openwindows(10)	glide(10)	feds(8.9)	circuit(8.9)	atheists(9.3)
pixmap(10)	harley(10)	fbi(8.8)	detector(8.9)	sank(9.1)
forsale	**med**	**mideast**	**hardware**	**Baseball**
cod(10)	albicans(10)	aggression(10)	nanao(10)	alomar(10)
forsale(9.8)	antibiotic(10)	anatolia(10)	dma(9.4)	astros(10)
comics(9.5)	antibiotics(10)	andi(10)	vlb(9.4)	baerga(10)
obo(9.0)	candida(10)	ankara(10)	irq(9.3)	baseman(10)
sale(8.8)	diagnosed(10)	apartheid(10)	soundblaster(9.0)	batter(10)
postage(8.6)	dyer(10)	appressian(10)	eisa(8.8)	batters(10)
shipping(8.6)	fda(10)	arabs(10)	isa(8.8)	batting(10)
mint(8.4)	homeopathy(10)	argic(10)	bios(8.7)	bullpen(10)
cassette(8.2)	infections(10)	armenia(10)	jumpers(8.7)	cardinals(10)
panasonic(7.6)	inflammation(10)	armenian(10)	adaptec(8.7)	catcher(10)

Table 5.20: Top 10 significant words per class for NGA.D10000.C10 using LTGSR, "all words" and "top *K*", with *G* = 3, LNT = 0.2%, and UNT = 7%

5.4.1 Number of Attributes

Table 5.21 shows the number of attributes generated using alternative combinations of the presented significant word generation and phrase generation strategies (experiment 2.1 in Table 5.2), including the case where the significant words alone were used as attributes (the strategy of Keywords). In all cases, the algorithms use as attributes the selected words or phrases, and the ten target classes. Thus, for the strategy of Keywords the number of attributes is the maximum number of significant words (1,500) plus the number of classes (10). In other experiments, reported in Appendix A, we examine the effect on Keywords of removing the upper limit, allowing up to 4,000 significant words to be used as attributes. We find that this leads to reduced accuracy, suggesting that a limit on the number of words used is necessary to avoid including words whose contribution may be spurious.

In the DelSNcontGO and DelSNcontGW algorithms, stop marks and noise words are used as delimiters. As the results demonstrate, this leads to many fewer phrases being identified than is the case for the other two phrase generation strategies, which use stop marks and ordinary words as delimiters. For DelSOcontGN (and to a lesser extent DelSOcontGW) the number of attributes generated usually exceeded the TFPC maximum of 2^{15} (32,767) attributes. This follows from the fact that these algorithms allow the inclusion of noise words in phrases. Since there are many more noise words (43,889) than ordinary words (4,207), the number of possible combinations for phrases far exceeds the number obtained using the two DelSN strategies. Further experiments that attempt to reduce the number of phrases produced by adjusting the *G* and *K* thresholds are reported in Appendix B. However, they did not lead to good results, and led to the abandonment of use of the DelSOcontGN and DelSOcontGW strategies.

Variations within the DelSN strategies were less extreme. DelSNcontGW produces fewer attributes than DelSNcontGO because phrases that are distinct in DelSNcontGO are collapsed into a single phrase in DelSNcontGW. Intuitively it appears that identifying more attributes (phrases) can improve the quality of representation and lead to better classification accuracy. In other experiments, reported in Appendix C, we increase the number of attributes produced by the DelSNcontGO and DelSNcontGW strategies by increasing the limit on the number of significant words generated. However, as was the case with the stragety of Keywords, this did not lead to any better accuracy, and was presumably caused by the fact that the additional significant words identified included many that were unhelpful or spurious.

Documentbase Pre-processing Strategy	**LTGFR**				**LTGSR**			
	Unique		**All**		**Unique**		**All**	
	Dist	**Top *K***	**Dist**	**Top *K***	**Dist**	**Top *K***	**Dist**	**Top *K***
DelSNcontGO	27,551	27,903	26,973	27,020	26,658	25,834	26,335	25,507
DelSNcontGW	11,888	12,474	12,118	13,657	11,970	11,876	11,819	11,591
DelSOcontGN	64,474	63,134	60,561	61,162	59,453	58,083	59,017	57,224
DelSOcontGW	32,913	34,079	32,549	35,090	32,000	32,360	31,542	31,629
Keywords	1,510	1,510	1,510	1,510	1,510	1,510	1,510	1,510

Table 5.21: Number of attributes (words or phrases) generated in NGA.D10000.C10

5.4.2 Classification Accuracy

Table 5.22 shows the percentage classification accuracy results obtained using the different strategies (experiment 2.2 in Table 5.2). Due to the fact that too many phrases were generated using DelSOcontGN and, in some cases, DelSOcontGW for the TFPC algorithm to operate, the results are incomplete for these algorithms. As can be seen, results obtained for DelSOcontGW are invariably poorer than for the other strategies. In the other cases, it is apparent that better results are always obtained when significant words are distributed equally between classes (columns headed "Dist") rather than by selecting only the *K* (1,500) most significant words. The best results were obtained with this policy using a potential significant word list made up of all words with a contribution above the *G* threshold (columns headed "All"), rather than when using only those that were unique to one class. Overall, DelSNcontGO performed slightly better than DelSNcontGW, and both phrase-generation strategies outperformed the algorithm of Keywords. The contribution calculation mechanism used did not appear to make a significant difference to these results.

Documentbase Pre-processing Strategy	**LTGFR**				**LTGSR**			
	Unique		**All**		**Unique**		**All**	
	Dist	**Top *K***	**Dist**	**Top *K***	**Dist**	**Top *K***	**Dist**	**Top *K***
DelSNcontGO	75.9	73.6	**77.3**	72.4	76.4	73.2	**77.4**	74.5
DelSNcontGW	75.1	71.6	**76.2**	68.5	74.9	71.3	**75.8**	72.3
DelSOcontGN								
DelSOcontGW			**70.9**		70.4	66.0	**71.2**	68.9
Keywords	75.1	73.9	**75.8**	71.2	74.4	72.2	**75.6**	73.7

Table 5.22: Classification accuracy in percentage for NGA.D10000.C10

5.4.3 Number of Empty Documents

Table 5.23 shows the number of "empty" training set documents found in the different cases: that is, documents in which no significant attributes were identified (experiment 2.3 in Table 5.2). These represent between 2% and 5% of the total training set. Perhaps more importantly, any such documents in the test set will necessarily be assigned to the default classification. Although no obvious relationship between the frequency of empty documents and classification accuracy is apparent from these results, further investigation of this group of documents may provide further insight into the operation of the proposed strategies.

Documentbase Pre-processing Strategy	LTGFR				LTGSR			
	Unique		All		Unique		All	
	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*
DelSNcontGO	190	258	251	299	229	238	224	370
DelSNcontGW	190	226	251	299	229	147	224	370
DelSOcontGN								
DelSOcontGW			251		229	411	224	370
Keywords	190	226	251	299	229	411	224	370

Table 5.23: Number of empty documents in the training set of NGA.D10000.C10

5.4.4 Execution Times

Table 5.24 shows execution times in seconds for the various algorithms, including both the time to generate rules and the time to classify the test set (experiment 2.4 in Table 5.2). The Keywords approach is faster than DelSNcontGO because a smaller number of attributes are considered and in this case TFPC generates fewer frequent sets and rules. However DelSNcontGW is the fastest as the use of the wild card leads to faster phrase matching.

Documentbase Pre-processing Strategy	LTGFR				LTGSR			
	Unique		All		Unique		All	
	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*
DelSNcontGO	244	250	253	242	250	248	328	235
DelSNcontGW	155	148	145	158	157	194	145	224
DelSOcontGN								
DelSOcontGW			370		326	281	278	314
Keywords	183	176	282	287	261	262	235	220

Table 5.24: Execution times in seconds for NGA.D10000.C10

5.4.5 Additional Experiments on the Significance Threshold

A further set of experiments were conducted to investigate the effect of adjusting the value of the significance threshold (*G*). The *G* value defines the minimum contribution that a potential significant word must have. The size of the potential significant word list thus increases with a corresponding decrease in *G*; conversely, we expect the quality of the words in the list to increase with *G*.

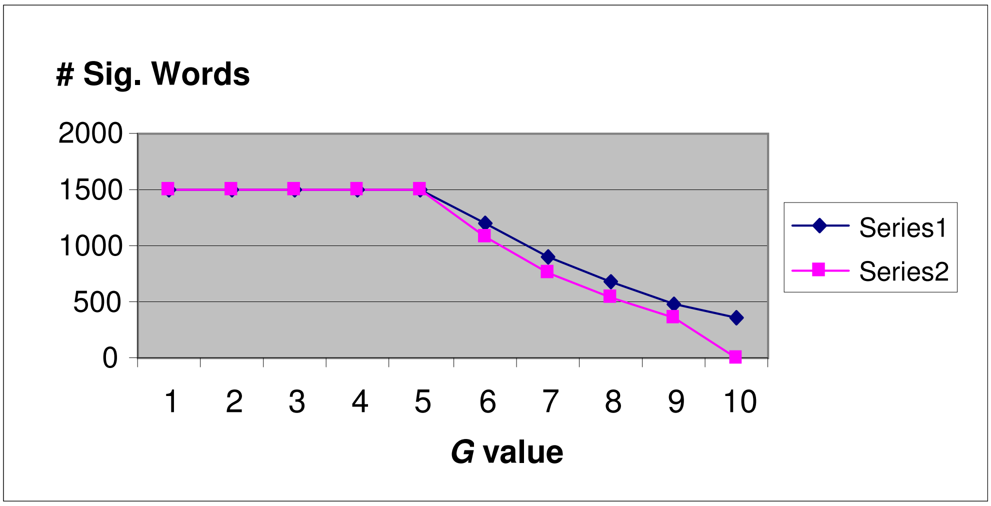


Figure 5.3: Relationship between *G* and number of final significant words selected for NGA.D10000.C10 with LNT = 0.2%, UNT = 7%, and *K* = 1,500 (Series 1 = LTGFR contribution calculation) & (Series 2 = LTGSR contribution calculation)

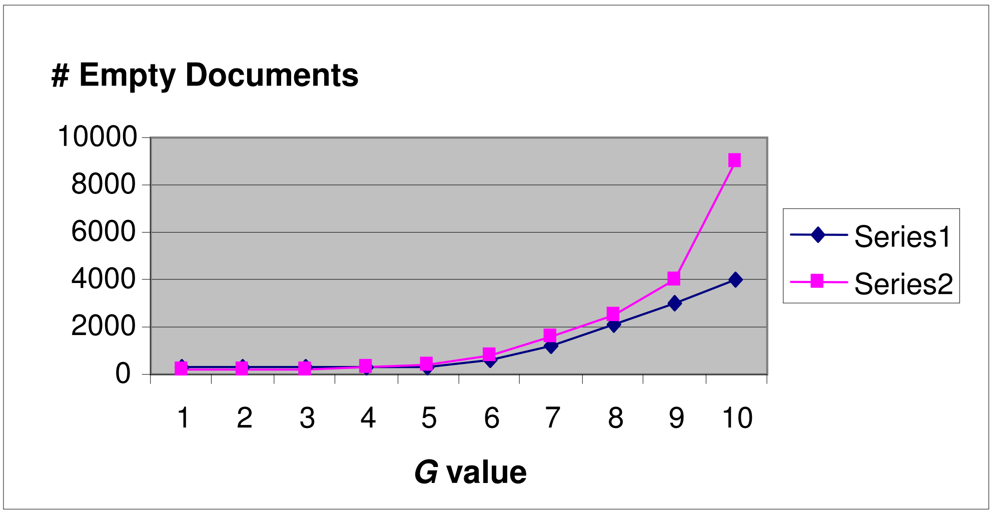


Figure 5.4: Relationship between *G* and number of empty documents generated for NGA.D10000.C10 with LNT = 0.2%, UNT = 7%, and *K* = 1,500 (Series 1 = LTGFR contribution calculation) & (Series 2 = LTGSR contribution calculation)

Figure 5.3 shows the effect on the number of selected final significant words with changes in *G*, when LNT = 0.2%, UNT = 7%, and *K* = 1,500 (experiment 2.5 in Table 5.2). The figure shows that there is little effect until the value of *G* reaches a point at which the size of the potential significant words list drops below *K*, and at which the number of chosen significant words falls rapidly and a corresponding fall in accuracy is also experienced. The drop is less severe using the LTGFR contribution calculation compared to the LTGSR contribution calculation.

Figure 5.4 shows the effect on the number of generated empty documents with changes in *G*, when LNT = 0.2%, UNT = 7%, and *K* = 1,500 (experiment 2.6 in Table 5.2). The figure demonstrates that the higher the value of *G* the higher the number of empty documents generated. The increase in the number of empty documents is less severe using the LTGFR contribution calculation compared to the LTGSR contribution calculation.

A further set of experiments is reported in Appendix D, which demonstrates the relationship between the significance threshold and the classification accuracy under the LTGFR and LTGSR approaches.

5.4.6 Discussion

The main findings of the above experiments were:

1. Best results were obtained from a strategy that made use of words that were significant in one or more classes, rather than only those that were unique to one class, coupled with a selection strategy that produced an equal distribution between classes.

2. The most successful phrase based strategy outperformed classification using only keywords: the most accurate approach is DelSNcontGO; and the fastest approach (with acceptable classification accuracy) is DelSNcontGW.

5.5 Experiment Group 3: Keyword Selection Method Evaluation

A further group of experiments are reported in this section that evaluate the four proposed keyword selection methods (see section 4.3) when (i) applied in a “bag of words” representation directly, and (ii) are employed to carry out a corresponding “bag of phrases” technique. They also examine whether the proposed “bag of phrases” approach outperforms the “bag of words” approach with respect to the accuracy of classification. Four prepared English language documentbases (*NGA.D10000.C10*, *NGB.D9997.C10*, *Reuters.D6643.C8*, and *OHSUMED.D6855.*

C10) were used in this experiment group. The accuracy of classification was determined using the TCV approach. The three sets of evaluations can be described as follows.

1. **Comparison of keyword selection methods in the "bag of words" setting:** A variety of statistics based feature selection techniques were described in section 3.3.2. Four of these previously developed techniques were selected for comparison against the proposed techniques. Each feature selection method concerned in this set of evaluations was directly applied as a "bag of words" documentbase pre-processing technique.

2. **Comparison of keyword selection methods in the "bag of phrases" setting:** All the above statistics based feature selection methods were employed to select potential significant words that were in turn used to generate significant phrases using the DelSNcontGO strategy. This set of evaluations aims to determine the most successful keyword selection method in the "bag of phrases" setting.

3. **The "bag of phrases" approach versus the "bag of words" approach:** Results obtained from the above experiments were used in the third set of evaluations, which investigated whether the proposed "bag of phrases" approach (DelSNcontGO) outperforms (is more accurate than) the "bag of words" approach (Keywords).

Following the main findings of the second group of experiments (see subsection 5.4.6), evaluations presented here were conducted using (i) the "all words" rather than "unique" (contribute to one class only) strategy in the construction of a potential significant word list, and (ii) the "dist or top K / $|\mathcal{C}|$" (equal distribution between classes) rather than "top K" strategy for choosing the final significant words. The parameter K (maximum number of selected final significant words) was chosen to be 150 × $|\mathcal{C}|$ (as described in subsection 5.3.4). To ensure the "top K / $|\mathcal{C}|$" (final) significant words were selected properly for each category, the G parameter was given a minimal value (*almost zero*) so that the G parameter could be ignored.

The value of LNT was chosen to be 0.2%. UNT was chosen to be 10% for both "20 Newsgroups" documentbases and 20% for both the Reuters-21578 and the

MedLine-OHSUMED documentbases. Although Figure 5.2 shows the best classification accuracy was achieved with a value of UNT of 7% for *NGA.D10000.C10*, a higher UNT value was applied in this group of experiments. The reason for this was that, as Table 5.21 shows, the number of attributes (phrases) generated using DelSNcontGO (with $G = 3$, $\sigma = 0.1\%$, $\alpha = 35\%$, LNT = 0.2%, and $K = 1{,}500$) was less than 28,000 for *NGA.D10000.C10*. It suggested that there was room to generate at least $2^{15} - 28{,}000 = 4{,}767$ more phrases and in so doing could improve the accuracy of classification. The value of 10% was set for *NGA.D10000.C10* and *NGB.D9997.C10* as the number of documents between classes is balanced and there are 10 pre-defined classes in each documentbase. For *Reuters.D6643.C8* and *OHSUMED.D6855.C10* where the number of documents between classes is not perfectly balanced, a higher UNT value (20%) was set.

5.5.1 Appropriate Values for Support and Confidence Thresholds

In the comparison of different keyword selection techniques, simply applying a support threshold value of 0.1% and confidence threshold value of 35% as used in experiment group 2 might not always be appropriate. In this case, the DIA Association Factor method (see section 3.3.2) was chosen as a benchmark — where a pair of appropriate support and confidence values were determined for each individual documentbase, based on directly applying DIAAF as a "bag of words" (Keywords) documentbase pre-processing technique.

$\sigma\backslash\alpha$	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**	**45%**	**50%**
0.1%	**77.59**	**77.59**	**77.59**	77.58	77.56	77.56	77.34	77.26	77.12	76.82
0.2%	77.23	77.23	77.23	77.23	77.23	77.24	77.08	76.97	76.65	76.19
0.3%	75.32	75.32	75.32	75.32	75.32	75.30	75.11	74.99	74.44	73.63
0.4%	74.00	74.00	74.00	74.00	74.00	73.98	73.77	73.57	72.88	73.57
0.5%	71.99	71.99	71.99	71.99	71.99	71.97	71.76	71.54	70.67	69.52
0.6%	69.25	69.25	69.25	69.25	69.25	69.23	68.99	68.75	67.72	66.50
0.7%	67.19	67.19	67.19	67.19	67.19	67.17	66.89	66.70	65.63	64.28
0.8%	65.12	65.12	65.12	65.12	65.11	65.06	64.83	64.63	63.56	62.19
0.9%	61.97	61.97	61.97	61.94	61.92	61.92	61.68	61.51	60.36	59.09
1%	60.18	60.18	60.18	60.13	60.09	60.09	59.88	59.72	58.33	57.14

Table 5.25: Classification accuracy obtained with varying thresholds of support and confidence with LNT = 0.2%, UNT = 10%, $K = 150 \times 10$, documentbase = NGA.D10000.C10, keyword selection = DIA Association Factor, and pre-processing approach = "bag of words"

$\sigma \backslash \alpha$	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**	**45%**	**50%**
0.1%	**80.69**	**80.69**	**80.69**	80.68	**80.69**	80.63	70.57	80.54	80.44	80.31
0.2%	80.02	80.02	80.02	80.02	80.02	79.98	79.91	79.86	79.73	79.64
0.3%	77.87	77.87	77.87	77.87	77.87	77.84	77.78	77.71	77.46	77.30
0.4%	76.24	76.24	76.24	76.24	76.24	76.19	76.10	75.95	75.66	75.39
0.5%	74.54	74.54	74.54	74.54	74.54	74.50	74.40	74.23	73.84	73.57
0.6%	72.84	72.84	72.84	72.84	72.84	72.79	72.66	72.50	72.06	71.83
0.7%	71.13	71.13	71.13	71.13	71.13	71.08	70.94	70.67	70.28	70.09
0.8%	68.94	68.94	68.94	68.94	68.94	68.88	68.74	68.47	67.87	67.82
0.9%	67.78	67.78	67.78	67.78	67.78	67.72	67.52	67.16	66.64	66.56
1%	66.93	66.93	66.93	66.93	66.93	66.87	66.65	66.25	65.92	65.92

Table 5.26: Classification accuracy obtained with varying thresholds of support and confidence with LNT = 0.2%, UNT = 10%, *K* = 150 × 10, documentbase = NGB.D9997.C10, keyword selection = DIA Association Factor, and pre-processing approach = "bag of words"

Tables 5.25 and 5.26 show classification accuracies for the *NGA.D10000.C10* and *NGB.D9997.C10* documentbases when reducing the value of support threshold in steps of 0.1% from 1% to 0.1% and the value of confidence threshold in steps of 5% from 50% to 5% (experiments 3.1 and 3.2 in Table 5.3). It can be seen that with a 0.1% support threshold value and a 5%, 10% or 15% confidence threshold value, the best classification accuracies are generated. Hence for all evaluations in this group of experiments using these documentbases, 0.1% and 5% were chosen to be the values for both support and confidence thresholds.

$\sigma \backslash \alpha$	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**	**45%**	**50%**
0.1%	84.67	84.66	84.66	84.66	84.66	84.89	85.46	85.55	85.77	**86.09**
0.2%	84.77	84.77	84.77	84.77	84.77	84.78	85.10	85.26	85.37	85.62
0.3%	84.70	84.70	84.70	84.70	84.70	84.70	85.01	85.07	85.23	85.38
0.4%	83.83	83.83	83.83	83.83	83.83	83.83	84.04	84.15	84.30	84.34
0.5%	83.35	83.35	83.35	83.35	83.35	83.35	83.49	83.55	83.71	83.79
0.6%	82.83	82.83	82.83	82.83	82.83	82.83	82.92	82.98	83.20	83.25
0.7%	82.62	82.62	82.62	82.62	82.62	82.62	82.63	82.73	82.96	82.97
0.8%	81.71	81.71	81.71	81.71	81.71	81.71	81.63	81.78	81.87	81.87
0.9%	80.94	80.94	80.94	80.94	80.94	80.94	80.73	80.70	80.69	80.69
1.0%	80.75	80.75	80.75	80.75	80.75	80.75	80.48	80.40	80.40	80.28

Table 5.27: Classification accuracy obtained with varying thresholds of support and confidence with LNT = 0.2%, UNT = 20%, *K* = 150 × 8, documentbase = Reuters.D6643.C8, keyword selection = DIA Association Factor, and pre-processing approach = "bag of words"

$\sigma \backslash \alpha$	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**	**45%**	**50%**
0.1%	77.18	77.18	77.18	77.18	77.17	77.17	77.26	77.37	77.61	**78.53**
0.2%	77.49	77.49	77.49	77.49	77.48	77.48	77.52	77.46	77.57	78.31
0.3%	77.69	77.69	77.69	77.69	77.68	77.68	77.61	77.48	77.46	78.05
0.4%	76.98	76.98	76.98	76.98	76.97	76.97	76.97	76.54	76.50	76.82
0.5%	76.54	76.54	76.54	76.54	76.54	76.51	76.18	75.78	75.74	75.66
0.6%	76.13	76.13	76.13	76.13	76.13	76.09	75.71	75.32	75.25	74.97
0.7%	75.87	75.87	75.87	75.87	75.87	75.83	75.39	74.88	74.83	74.56
0.8%	75.61	75.61	75.61	75.61	75.61	75.57	75.10	74.53	74.43	74.14
0.9%	75.58	75.58	75.58	75.58	75.58	75.51	75.00	74.41	74.31	74.03
1.0%	75.45	75.45	75.45	75.45	75.45	75.35	74.79	74.18	74.09	73.81

Table 5.28: Classificataion accuracy obtained with varying thresholds of support and confidence with LNT = 0.2%, UNT = 20%, *K* = 150 × 10, documentbase = OHSUMED.D6855.C10, keyword selection = DIA Association Factor, and pre-processing approach = "bag of words"

Table 5.27 shows classification accuracies for *Reuters.D6643.C8,* and Table 5.28 for *OHSUMED.D6855.C10*, for the same range of thresholds (experiments 3.3 and 3.4 in Table 5.3). In both cases the highest accuracies were obtained with a 0.1% support threshold value and a 50.0% confidence threshold value. Thus for all evaluations in this experiment group using these documentbases, 0.1% and 50% were applied as the values of support and confidence thresholds.

5.5.2 Keyword Selection Method Comparison in "Bag of Words"

Four new significant word selection methods were introduced in section 4.3: LTGSR (Local-To-Global Support Ratio), LTGFR (Local-To-Global Frequency Ratio), DIAAF-based-RS (Darmstadt Indexing Approach Association Factor based Relevancy Score), and DIAAF-based-GSS (Darmstadt Indexing Approach Association Factor based Galavotti·Sebastiani·Simi). This subsection aims to evaluate the performance of these methods with respect to the accuracy of classification. Each method was directly applied as a "Keywords" documentbase pre-processing strategy. Four established keyword selection techniques were used for comparison: DIAAF (Darmstadt Indexing Approach Association Factor), RS (Relevancy Score), OR (Odds Ratio), and MI (Mutual Information). Note that since the *G* parameter was set to *almost zero*, operation of both the LTGSR and the MI methods was very similar (see section 4.3.1). Hence there were seven distinct keyword selection techniques used in this set of evaluations: DIAAF, OR, RS, LTGSR/MI, LTGFR, DIAAF-based-RS, and DIAAF-based-GSS. In both RS and DIAAF-based-RS, 0 was used as the constant damping factor value (see section 3.3.2 & section 4.4.3).

Table 5.29 shows that with a 0.1% support threshold, a 5% confidence threshold, 0.2% LNT, 10% UNT and *K* of 1,500, the most accurate keyword selection method for both *NGA.D10000.C10* and *NGB.D9997.C10* was LTGFR (experiments 3.5 and 3.6 in Table 5.3). With the *Reuters.D6643.C8* documentbase (with $\sigma = 0.1\%$, $\alpha = 50\%$, LNT = 0.2%, UNT = 20%, and K = 1,200), the best keyword selection approach was LTGSR/MI (experiment 3.7 in Table 5.3). Finally, the DIAAF-based-RS technique outperformed other alternative methods with the *OHSUMED.D6855.C10* documentbase (with $\sigma = 0.1\%$, $\alpha = 50\%$, LNT = 0.2%, UNT = 20%, and K = 1,500) (experiment 3.8 in Table 5.3). Overall LTGSR/MI was the most accurate keyword selection technique with an 81.16% average accuracy of classification for the four English language documentbases considered here, while LTGFR was found to have the highest number of best classification accuracies (2 out of 4 cases).

	DIAAF	OR	RS	LTGSR / MI	LTGFR	DIAAF-based-RS	DIAAF-based-GSS
NGA.D10000.C10	77.59	77.47	77.59	77.59	**77.61**	77.38	77.54
NGB.D9997.C10	80.69	80.71	80.69	80.69	**80.89**	80.69	80.72
Reuters.D6643.C8	86.09	86.65	86.78	**87.04**	86.48	86.90	85.05
OHSUMED.D6855.C10	78.53	79.32	79.27	79.32	79.17	**79.37**	78.75
Average Accuracy	80.73	81.04	81.08	**81.16**	81.04	81.09	80.52
# of Best Accuracies	0	0	0	1	**2**	1	0

Table 5.29: Classification accuracy — comparison of the seven keyword selection techniques in the "bag of words" setting

5.5.3 Keyword Selection Method Comparison in "Bag of Phrases"

Table 5.30 shows classification accuracies, obtained when the different keyword selection techniques were employed to generate significant phrases. It can be seen that the DIAAF, RS and LTGSR/MI techniques performed the best for the

NGA.D10000.C10 documentbase (with $\sigma = 0.1\%$, $\alpha = 5\%$, LNT = 0.2%, UNT = 10%, and K = 1,500) (experiment 3.9 in Table 5.3). With the parameters set identically, LTGFR was found to be the most accurate approach for the *NGB.D9997.C10* documentbase (experiment 3.10 in Table 5.3). DIAAF-based-RS and DIAAF-based-GSS were the best techniques for the *Reuters.D6643.C8* documentbase with $\sigma = 0.1\%$, $\alpha = 50\%$, LNT = 0.2%, UNT = 20%, and K = 1,000 (experiment 3.11 in Table 5.3). Here the value of K was chosen to be 1,000 instead of 1,200 (as used in the previous set of evaluations) because 1,200 (choosing the top 150 significant words per class) resulted in more than 2^{15} significant phrases being generated. Keeping all other parameter values unchanged and changing the K value from 1,000 to 900, the highest classification accuracy for *OHSUMED.D6855.C10* was generated by LTGFR (experiment 3.12 in Table 5.3). Again, the reason for decreasing the value of K from 1,500 (as used in the previous set of evaluations) to 900 was that handling 1,500 significant words would breach the 2^{15} limit of significant phrase generation. Overall, the DIAAF-based-GSS technique gave the highest average accuracy of classification (82.14%) for the four English language documentbases considered here, and the LTGFR method gave the highest number of best classification accuracies (2 out of 4 cases).

	DIAAF	OR	RS	LTGSR / MI	LTGFR	DIAAF-based-RS	DIAAF-based-GSS
NGA.D10000.C10	**78.35**	78.21	**78.35**	**78.35**	77.98	78.21	78.31
NGB.D9997.C10	82.22	82.31	82.22	82.22	**82.52**	82.39	82.28
Reuters.D6643.C8	87.57	87.84	87.79	87.79	87.51	**88.23**	**88.23**
OHSUMED.D6855.C10	78.83	79.68	79.64	79.53	**79.80**	79.62	79.74
Average Accuracy	81.74	82.01	82.00	81.97	81.95	82.11	**82.14**
# of Best Accuracies	1	0	1	1	**2**	1	1

Table 5.30: Classification accuracy — comparison of the seven keyword selection techniques in the "bag of phrases" setting

5.5.4 "Bag of Phrases" Versus "Bag of Words"

From Tables 5.29 and 5.30, it can be seen that in all cases (all considered keyword selection methods with all used documentbases) the accuracy of classification was improved by employing the proposed “bag of phrases” rather than the alternative “bag of words” approach. Table 5.31 further demonstrates the comparison between both documentbase pre-processing approaches — in general the “bag of phrases” outperformed the “bag of words” approach (experiment 3.13 in Table 5.3) and the classification accuracy was improved by a factor of around 1%.

	DIAAF	**OR**	**RS**	**LTGSR /MI**	**LTGFR**	**DIAAF-based-RS**	**DIAAF-based-GSS**
Average accuracy in “bag of words”	80.73	81.04	81.08	81.16	81.04	81.09	80.52
Average accuracy in “bag of phrases”	81.74	82.01	82.00	81.97	81.95	82.11	82.14
Accuracy difference	1.01	0.97	0.92	0.81	0.91	1.02	1.62
Average difference in accuracy	**1.04**						

Table 5.31: Classification accuracy — “bag of phrases” vs. “bag of words”

A further evaluation between the “bag of phrases” and the “bag of words” approaches is provided, based on the *standard deviation*. In the “bag of words” setting, the average (mean) accuracy of classification for the seven different keyword selection techniques is 80.95%, and the standard deviation is 0.23%. In comparison, the average (mean) accuracy of classification for these seven keyword selection methods in the “bag of phrases” setting is 81.99%, and the standard deviation is 0.13%. It can be concluded on the basis of the four English documentbases that the classification accuracies generated by using the “bag of phrases” approach are more concentrated than the “bag of words” approach. In other words, the “bag of phrases” approach outperforms the “bag of words” approach not only because the former one has a higher average accuracy of classification but also the stability of the performance.

5.5.5 Discussion

In Table 5.32, the average classification accuracies for all seven keyword selection techniques are presented in rank order. The best keyword selection method in the “bag of words” setting is the proposed LTGSR/MI approach, and the best approach in the "bag of phrases" setting is the proposed DIAAF-based-GSS technique. Table 5.33 shows the number of instances of best classification accuracies for all seven keyword selection techniques considered here: the best results here in both the "bag of words" and the "bag of phrases" settings coming from LTGFR. Overall, the newly introduced DIAAF-based-RS mechanism may be identified as the most consistently successful method. In all cases the proposed "bag of phrases" approach outperformed the "bag of words" approach.

Rank No.	**In "bag of words" Setting**		**Rank No.**	**In "bag of phrases" Setting**	
	Technique	Accuracy		Technique	Accuracy
1	LTGSR/MI	81.16	1	DIAAF-based-GSS	82.14
2	DIAAF-based-RS	81.09	2	DIAAF-based-RS	82.11
3	RS	81.08	3	OR	82.01
4	LTGFR	81.04	4	RS	82.00
4	OR	81.04	5	LTGSR/MI	81.97
6	DIAAF	80.73	6	LTGFR	81.95
7	DIAAF-based-GSS	80.52	7	DIAAF	81.74

Table 5.32: Ranked order of classification accuracies for the seven keyword selection techniques

Rank No.	**In "bag of words" Setting**		**Rank No.**	**In "bag of phrases" Setting**	
	Technique	No. of Bests		Technique	No. of Bests
1	LTGFR	2	1	LTGFR	2
2	DIAAF-based-RS	1	2	DIAAF-based-RS	1
2	LTGSR/MI	1	2	DIAAF-based-GSS	1
4	DIAAF-based-GSS	0	2	LTGSR/MI	1
4	RS	0	2	RS	1
4	OR	0	2	DIAAF	1
4	DIAAF	0	7	OR	0

Table 5.33: Ranked order of the number of best accuracies for the seven keyword selection techniques

5.6 Experiment Group 4: Chinese Data Set Experiments

In this section, a set of experiments are reported that examine the performance of the proposed language-independent documentbase pre-processing approaches when applied to a non-English (i.e. Chinese) language text collection. The *Chinese.D2816.C10* documentbase (see subsection 5.2.4) was used in the experiments reported here. The suite of experiments used the first 9/10th (2,534 documents) as the training set, and the last 1/10th (282 documents) as the test set. For the experimental results shown here, the following thresholds/parameters were used: LNT = 0.2%, UNT = 10%, K = 150 × 10 classes, potential significant word list construction = "all words", and final significant word selection = "dist or top K / |€|" (as previously used when dealing with "20 Newsgroups" documentbases in experiment group 3). The G parameter was also set to *almost zero* (as discussed in section 5.5).

Again, the DIAAF method was chosen as the benchmark "bag of words" technique to determine the most appropriate support and confidence threshold values (experiment 4.1 in Table 5.4) to be used in this experiment group. Table 5.34 shows that the highest classification accuracy (70.92%) was produced by a 1% support threshold and a 50% confidence threshold. Note that with a low support threshold value (i.e. 0.1% or 0.2%), too many attributes (more than 2^{15}) were generated, thus no classifier was produced and consequently no classification accuracy could be obtained. Thus a 1% support threshold and a 50% confidence threshold were applied in this group of evaluations.

$\sigma \backslash \alpha$	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**	**45%**	**50%**
0.1%										
0.2%	66.67		66.67	62.06						
0.3%	66.67	66.67	66.67	64.89	62.06	61.35	63.83			
0.4%	66.67	66.67	66.67	66.31	66.67	66.67	67.38	59.57	59.57	69.50
0.5%	65.60	65.60	65.60	65.60	65.25	64.89	65.25	59.57	57.45	68.09
0.6%	64.89	64.89	64.89	64.89	64.54	64.89	67.38	62.06	58.87	64.89
0.7%	65.25	65.25	65.25	65.25	64.54	65.25	66.31	62.41	58.87	64.89
0.8%	68.44	68.44	68.44	68.44	68.44	68.79	68.79	64.89	62.77	67.02
0.9%	69.50	69.50	69.50	69.50	69.50	69.86	69.86	67.02	65.96	68.44
1.0%	69.50	69.50	69.50	69.50	69.50	69.50	69.86	67.02	67.38	**70.92**

Table 5.34: Classification accuracy obtained with varying thresholds of support and confidence with LNT = 0.2%, UNT = 10%, K = 150 × 10, documentbase = Chinese.D2816.C10, keyword selection = DIA Association Factor, and pre-processing approach = "bag of words"

Rank No.	Technique	Accuracy	Rank No.	Technique	Accuracy
1	LTGFR (phrases)	72.34	7	DIAAF (words)	70.92
2	LTGSR/MI (phrases)	71.99	9	DIAAF-based-GSS (words)	70.57
3	DIAAF-based-RS (phrases)	71.63	10	LTGSR/MI (words)	69.86
3	OR (phrases)	71.63	11	LTGFR (words)	68.44
5	DIAAF-based-GSS (phrases)	71.28	12	OR (words)	68.09
5	RS (phrases)	71.28	13	DIAAF-based-RS (words)	65.96
7	DIAAF (phrases)	70.92	14	RS (words)	63.83

Table 5.35: Ranked order of classification accuracies for all fourteen pre-processing methods with LNT = 0.2%, UNT = 10%, K = 150 × 10, σ= 1%, α= 50%, and documentbase = Chinese.D2816.C10

The evaluations presented in this section show that the proposed language-independent "bag of phrases" approach performed with an acceptable accuracy of classification when a non-English language documentbase (*Chinese.D2816.C10*) was used. Table 5.35 shows that in this case the newly proposed LTGFR keyword selection technique was the most accurate approach with 72.34% classification accuracy; and again in all cases the "bag of phrases" approach outperformed the "bag of words" approach (experiments 4.2 ~ 4.4 in Table 5.4).

A further evaluation between the "bag of phrases" and the "bag of words" approaches is provided, based on the *standard deviation*. In the "bag of words" setting, the average (mean) accuracy of classification for the seven different keyword selection techniques is 68.24%, and the standard deviation is 2.58%. In comparison, the average (mean) accuracy of classification for these seven keyword selection methods in the "bag of phrases" setting is 71.58%, and the standard deviation is 0.48%. These show that for the *Chinese.D2816.C10* documentbase the classification accuracies generated by using the "bag of phrases" approach are more concentrated than the "bag of words" approach. In other words, the "bag of phrases" approach outperforms the "bag of words" approach not only because the former one has a higher average accuracy of classification but also the stability of the performance.

Technique	Time	Technique	Time
LTGFR (phrases)	192.03	DIAAF (words)	47.45
LTGSR/MI (phrases)	180.39	DIAAF-based-GSS (words)	51.06
DIAAF-based-RS (phrases)	161.88	LTGSR/MI (words)	47.09
OR (phrases)	160.46	LTGFR (words)	46.06
DIAAF-based-GSS (phrases)	160.57	OR (words)	47.31
RS (phrases)	183.56	DIAAF-based-RS (words)	45.89
DIAAF (phrases)	173.25	RS (words)	48.00

Table 5.36: Execution times in seconds for all fourteen pre-processing methods with LNT = 0.2%, UNT = 10%, K = 150 × 10, σ = 1%, α = 50%, and documentbase = Chinese.D2816.C10

Table 5.36 further shows the execution time when each language-independent documentbase pre-processing approach is applied to the *Chinese.D2816.C10* documentbase (experiment 4.5 in Table 5.4). This set of evaluations demonstrate that when directly applying each keyword selection technique as a "bag of words" the training data and the test data (for this Chinese language documentbase) can be efficiently learned and classified within 45 to 60 seconds. On the other hand, the overall execution time has an upper limit of 200 seconds when substituting the "bag of words" approach by the proposed language-independent "bag of phrases" (in particular the DelSNcontGO) approach.

Two additional experiments that were run and are presented at the end of this section show that acceptable classification accuracies using TCV can be produced when employing both the LTGFR and the DIAAF-based-RS keyword selection techniques to generate significant phrases (DelSNcontGO) that are in turn used to mine rules and classify "unseen" Chinese documents (i.e. the test data of *Chinese.D2816.C10*). With the following parameter settings, σ = 1%, α = 50%, G = *almost zero* (as 10^{-4}), LNT = 0.2%, UNT = 20% (as previously used for *unbalanced* documentbases), and K = 100 × 10, the LTGFR approach resulted in a 71.23% classification accuracy, while the accuracy of classification for the DIAAF-based-RS was found to be 70.46%.

5.7 Summary

At the beginning of this chapter, four tables (Tables 5.1, 5.2, 5.3 and 5.4) were provided that listed all the individual experiments described in this chapter, where the experiment title, documentbase used and the objective of each experiment was detailed. Tables 5.37, 5.38, 5.39 and 5.40 recall Tables 5.1, 5.2, 5.3 and 5.4 but provide a summary of main findings in place of objectives.

No.	Experiment Title	Documentbase	Main Findings
1.1	Effect of Changing the Significance Threshold	NGA.D10000.C10	The significance threshold (based on the LTGSR keyword selection method) should have a maximum value of 2.5. Note that for Reuters. D6643.C8, the value was best set to 1.
1.2		NGB.D9997.C10	
1.3		Reuters.D6643.C8	
1.4	Effect of Changing the Support Threshold	NGA.D10000.C10	The support threshold value should be 0.25% or maybe even lower.
1.5		NGB.D9997.C10	
1.6		Reuters.D6643.C8	
1.7	Effect of Changing the Confidence Threshold	NGA.D10000.C10	The confidence threshold value should be less than or equal to 50%. Note that for both “20 Newsgroups” documentbases, this value should be 15% or lower.
1.8		NGB.D9997.C10	
1.9		Reuters.D6643.C8	
1.10	Effect of Changing UNT with Low Support Threshold and 150 selected Final Significant Words per Class	NGA.D10000.C10	(i) The UNT should be set at 7% or maybe even higher; (ii) The support threshold value should be ≤ 0.1%; and (iii) The maximum number of selected final significant words should be set at 150 per class.

Table 5.37: Main findings of experiments described in experiment group 1: Threshold testing

No.	Experiment Title	Documentbase	Main Findings
2.1	Number of Attributes	NGA.D10000.C10	Both DelSO strategies perform badly.
2.2	Classification Accuracy in Percentage		(i) Best accuracies are always obtained from the “all words” potential significant word list construction strategy; (ii) Best accuracies are always obtained from the “dist” final significant word selection strategy; (iii) The DelSNcontGO approach outperforms the approach of Keywords (with respect to classification accuracy); and (iv) There is no significant difference in accuracy between LTGSR and LTGFR.
2.3	Number of Empty Documents in the Training Data Set		Between 2% and 5% of the total training set were found as empty documents (with no significant attributes identified). There remain possibilities to improve the proposed techniques with regard to the avoidance of empty documents.
2.4	Execution Times in Seconds		The most efficient technique is DelSNcontGW, which also produces comparable classification accuracy to the Keywords approach.
2.5	Relationship between the Significance Threshold and the Number of identified Significant Words		The higher the significance threshold value (from 3 to 10) the lower the number of significant words generated leading to a decrease in classification accuracy.
2.6	Relationship between the Significance Threshold and the Number of generated Empty Documents in the Training Data Set		The higher the significance threshold value (from 3 to 10) the more empty documents were generated leading to a decrease in classification accuracy.

Table 5.38: Main findings of experiments described in experiment group 2: Documentbase pre-processing strategy evaluation

No.	Experiment Title	Documentbase	Main Findings
3.1	Classification Accuracy obtained when varying both the Support and Confidence Thresholds	NGA.D10000.C10	0.1% and 5% should be used as the support and the confidence threshold values respectively.
3.2		NGB.D9997.C10	
3.3		Reuters.D6643.C8	0.1% and 50% should be used as the support and confidence threshold values respectively.
3.4		OHSUMED.D6855.C10	
3.5	Comparison of the Keyword Selection Techniques in "Bag of Words"	NGA.D10000.C10	Best performing technique is LTGFR with 77.61% accuracy.
3.6		NGB.D9997.C10	Best performing technique is LTGFR with 80.89% accuracy.
3.7		Reuters.D6643.C8	Best performing technique is LTGSR/MI with 87.04% accuracy.
3.8		OHSUMED.D6855.C10	Best performing technique is DIAAF-based-RS with 79.37% accuracy.
3.9	Comparison of the Keyword Selection Techniques in "Bag of Phrases"	NGA.D10000.C10	Best performing techniques are DIAAF, RS and LTGSR/MI with 78.35% accuracy.
3.10		NGB.D9997.C10	Best performing technique is LTGFR with 82.52% accuracy.
3.11		Reuters.D6643.C8	Best performing techniques are DIAAF-based-RS and DIAAF-based-GSS with 88.23% accuracy.
3.12		OHSUMED.D6855.C10	Best performing technique is LTGFR with 79.80% accuracy.
3.13	Comparison of the "Bag of Phrases" Approach and the "Bag of Words" Approach	All English language documentbases	With respect to both the average accuracy of classification and the standard deviation for the seven different keyword selection techniques, the "bag of phrases" approach (DelSNcontGO) outperforms the "bag of words" approach.

Table 5.39: Main findings of experiments described in experiment group 3: Keyword selection method evaluation

No.	Experiment Title	Documentbase	Main Findings
4.1	Classification Accuracy obtained with Varying both the Support and Confidence Thresholds	Chinese.D2816.C10	1% and 5% should be used as the support and the confidence threshold values respectively.
4.2	Comparison of the Keyword Selection Techniques in "Bag of Words"		Best performing technique is DIAAF with 70.92% accuracy.
4.3	Comparison of the Keyword Selection Techniques in "Bag of Phrases"		Best performing technique is LTGFR with 72.34% accuracy.
4.4	Comparison of the "Bag of Phrases" Approach and the "Bag of Words" Approach		With respect to both the average accuracy of classification and the standard deviation for the seven different keyword selection techniques, the "bag of phrases" approach (DelSNcontGO) outperforms the "bag of words" approach.
4.5	Execution Times in Seconds		Alternative language-independent documentbase pre-processing approaches coupled with the TFPC CARM approach can be processed efficiently.

Table 5.40: Main findings of experiments described in experiment group 4: Chinese data set experiments

In this chapter, the overall performance of the proposed language-independent documentbase pre-processing strategies for the single-label multi-class TC task was evaluated. The overall main findings can be summarised as follows:

1. The four newly proposed statistics based feature selection mechanisms appear to produce better results (in classification accuracy) than the previous mechanisms when (i) directly utilising each mechanism in a "bag of words" approach, and (ii) employing each mechanism in a corresponding (proposed) "bag of phrases" approach.
2. Both proposed "bag of phrases" documentbase pre-processing strategies (DelSNcontGO and DelSNcontGW) outperform the "bag of words" approach with regard to both the accuracy of classification and the efficiency of computation.
3. The proposed documentbase pre-processing strategies were successful in giving acceptable classification accuracy and fast processing efficiency for different documentbases presented in two distinct languages. It is conjectured that this result would extend to other languages.

Chapter 6

Overall Conclusion

6.1 Introduction

Text Classification/Categorisation (TC) has become a popular topic in the research areas of both Knowledge Discovery in Databases (KDD) and machine learning. In general, the TC process comprises stages of textual data pre-processing and Classification Rule Mining (CRM). The work described in this thesis has investigated a number of ways of pre-processing textual data in a language-independent fashion so as to enable:

1. TC to be carried out efficiently without any deep linguistic analysis or the use of language-specific techniques; and
2. Text classifiers to be built that are globally applicable to cross-lingual, multi-lingual and/or unknown-lingual textual data collections.

In the CRM stage, the TFPC CARM algorithm was chosen to be used. Experimental results based on TFPC have shown that the language-independent documentbase pre-processing methods that were considered performed well.

In this chapter an overall summary of the proposed methods examined is given, coupled with an evaluation of them. The organisation of this chapter is as follows. The following section reviews the language-independent "bag of words" approach, where four statistics based feature selection mechanisms were examined. Section 6.3 reviews the proposed language-independent "bag of phrases" approach, and concludes that the "bag of phrases" representation seems to outperform the "bag of words" approach (contrary to [Lewis, 1992] and [Scott and Matwin, 1999]). An overall summary is provided in section 6.4 that concludes that it is indeed possible to perform TC effectively in a language-independent and domain-independent manner. Finally a number of issues for further research are discussed in section 6.5.

6.2 Language-independent "Bag of Words"

The framework of the "bag of words" approach is common and simple, consisting of four phases:

1. All non-alphabetic textual components are removed from the given documentbase;
2. All LNWs (Lower Noise Words) and UNWs (Upper Noise Words) are removed from consideration;
3. Selecting the key text-features (single words) from the documentbase that significantly serve to differentiate between classes, based on a selected statistics based feature selection mechanism; and
4. Recasting the original documentbase in terms of the selected significant words only.

The third phase of this framework defines the specific "bag of words" approach. Note that by employing different feature selection methods will lead to different keyword sets and consequently result in different classification accuracies.

In this thesis, four new feature selection mechanisms have been investigated, namely: LTGSR, LTGFR, DIAAF-based-RS, and DIAAF-based-GSS. For each mechanism a number of parameters that might influence the classification accuracy had to be calibrated. These included: G (significance threshold), σ (support threshold), α (confidence threshold), LNT (Lower Noise Threshold), UNT (Upper Noise Threshold), K (maximum number of selected final significant words), the strategy for selecting the list of potential significant words (either "all words" or "unique"), and the final significant word selection strategy (either "top K" or "dist").

Taking the LTGSR approach as an example and setting the parameters at their default values ($G = 1.5$; $\sigma = 0.25\%$; $\alpha = 35\%$; LNT = 0.5%; and UNT = 3.5%) resulted in the best possible classification accuracy for the *NGA.D10000.C10*, *NGB.D9997.C10* and *Reuters.D6643.C8* documentbases at 69.14%, 73.46% and 76.26% respectively. Setting the parameters at levels designed to improve accuracy (G = *almost zero*; σ = 0.1%; α = 5% for both "20 Newsgroups" documentbases and 50% for *Reuters.D6643.C8*; LNT = 0.2%; UNT = 10% for both "20 Newsgroups" documentbases and 20% for *Reuters.D6643.C8*; K = 1,500; potential significant word list strategy = "all words"; and final significant word selection strategy = "dist"), classification accuracies obtained for *NGA.D10000.C10*, *NGB.D9997.C10* and *Reuters.D6643.C8* improved to 77.59%, 80.69% and 87.04% respectively.

The accuracy of the four new feature selection methods was compared with that obtained using established methods, (i.e. DIAAF, OR and RS). In general, improved accuracy was obtained using the new methods. The highest average accuracy of classification for the four English language documentbases considered was 81.16%, produced by the LTGSR approach; with the second highest generated by DIAAF-based-RS. The highest number of best classification accuracies was obtained using LTGFR, and the second highest number was obtained equally using DIAAF-based-RS and LTGSR methods. In general DIAAF, RS and OR gave lower average accuracies of classification, and in no case led to the best classification accuracy. However, when applying the methods to a Chinese textual dataset (*Chinese.D2816.C10*) the best performing method was DIAAF with an accuracy of 70.92%, although the DIAAF-based-GSS produced comparable results (70.57%). Note (see section 5.6) that when considering the "bag of phrases" approach the best results, using the Chinese dataset, were obtained using LTGFR.

6.3 Language-independent "Bag of Phrases"

A number of strategies were described for language-independently identifying significant phrases in documentbases to be used in a "bag of phrases" representation for TC. Phrases are generated using four different schemes to combine noise, ordinary and significant words, and stop marks: DelSNcontGO, DelSNcontGW, DelSOcontGN, and DelSOcontGW. The essential characteristic of these schemes is that they allow phrases to be constructed simply and automatically, using only simple statistical properties of words, without the need for any semantic analysis. Initial experiments conducted were based on three documentbases

(*NGA.D10000.C10*, *NGB.D9997.C10*, and *Reuters.D6643.C8*) and used the LTGSR keyword selection method to define significant words for use in the construction of phrases. Early results indicated that the DelSN strategies perform better than the DelSO schemes. Again, a series of experiments was carried out with the aim of finding better parameter settings (as detailed in section 6.2). These led to better classification accuracies: (i) for *NGA.D10000.C10* accuracy increased from 69.14% to 78.35%; (ii) for *NGB.D9997.C10* accuracy increased from 73.66% to 82.22%; and (iii) for *Reuters.D6643.C8*, accuracy increased from 76.10% to 87.79%. The *NGA.D10000.C10* documentbase was also used in another set of experiments using the LTGSR and LTGFR keyword selection mechanisms. The main findings of this set of experiments were:

1. For both keyword selection mechanisms, the DelSN strategies outperformed the other pre-processing approaches (including the keyword approach) with regards to both accuracy of classification (DelSNcontGO produced the best accuracy) and the efficiency of computation (DelSNcontGW was the most efficient); and
2. For both keyword selection methods the best classification accuracy was obtained from a strategy that made use of words that were significant in one or more classes, rather than only those that were unique to one class, coupled with a selection strategy that produced an equal distribution between classes ("dist or top *K* / |€|" is better than "top *K*").

In a further set of evaluations, the four new keyword selection mechanisms (LTGSR, LTGFR, DIAAF-based-RS, and DIAAF-based-GSS) were compared with previous methods (i.e. DIAAF, OR, RS) in the "bag of phrases" setting. Results of this set of experiments corroborated the results produced when comparing these mechanisms in the "bag of words" setting — the proposed methods in general seem to outperform the established methods. The highest average accuracy of classification throughout the four English language documentbases was 82.14%, produced by DIAAF-based-GSS, and the second highest average accuracy of classification was generated by the DIAAF-based-RS mechanism. The LTGFR approach gave the highest accuracy in most cases (2 out of 4). When using the *Chinese.D2816.C10* documentbase the best performance was obtained using LTGFR with an accuracy of 72.34%.

6.4 Overall Summary

In the work described in this thesis, the possibility of carrying out Text Classification effectively in a language-independent and/or domain-independent fashion has been investigated. A number of new strategies for accomplishing this have been described and evaluated experimentally.

- A language-independent noise word identification method was introduced, where (upper and lower) noise words are determined by their documentbase support value. The upper noise words for each documentbase used in this study were identified (see Tables 5.10 ~ 5.14) that demonstrate the effectiveness of this approach.

- In language-independent significant word identification, four statistical keyword (potential significant word) selection methods were proposed. The experimental results demonstrate that these newly proposed methods outperform existing approaches.

- Two strategies were introduced for the potential significant word list construction. It was shown that the "all words" strategy outperformed the "unique" strategy.

- Two strategies were proposed for the final significant word selection. The "dist" approach outperformed the "top *K*" approach.

- In language-independent significant phrase identification, four schemes were proposed to generate significant phrases. The experimental results demonstrate that the DelSNcontGO "bag of phrases" approach outperforms all other alternatives (including the *Keywords* "bag of words" approach) in terms of both the average accuracy of classification and the standard deviation for a variety of keyword selection techniques; whilst the DelSNcontGW "bag of phrases" approach was found to be the most efficient.

The results obtained in this study demonstrate that, using simple and efficient algorithms that do not make use of language-specific characteristics or deep linguistic analysis, relatively high classification accuracy can be achieved. The results demonstrate that with English texts, classification can be performed language-independently with greater than 82% average accuracy of classification; and greater than 71% classification accuracy when dealing with the Chinese textual data (using TCV).

6.5 Future Work

The experimental work described in this thesis gave rise to a number of observations that have not been fully investigated in the present work. A number of rule ordering approaches were discussed, in section 2.6.3, as alternatives to the CSA (Confidence Support & size-of-rule-Antecedent) mechanism that was used in the experiments presented in this thesis. In an examination of these, reported in [Wang *et al.*, 2007b], experiments with various rule ordering approaches, based on a range of database-like datasets rather than textual datasets, found that CSA based hybrid rule ordering schemes appear to significantly increase the classification accuracy produced by CSA. This suggests an investigation to further find out if CSA based hybrid rule ordering schemes might also perform well for the classification of textual data.

Further research is also suggested to examine the effectiveness of the methods described in this thesis for a wider range of textual data and range of languages. Other CRM approaches might also be investigated to examine whether these pre-processing approaches work equally well using different classifiers.

Finally, other directions for further research include:

- The present work has focussed on single-label multi-class TC. Further work is suggested to examine the application of the proposed language-independent documentbase pre-processing methods to binary, multi-label and/or hierarchical TC tasks.

- A range of text mining applications were described in section 2.7. The possibility of solving the traditional TC problem in a language-independent fashion has been demonstrated in this thesis. It will be interesting to apply this approach to other text mining problems, i.e. language-independent document clustering, language-independent topic detection and tracking, language-independent text summarisation, etc.

Bibliography

[Agrawal *et al.*, 1993] Agrawal, R., Imielinski, T., and Swami, A. (1993) Mining association rules between sets of items in large databases. In: Buneman, P., and Jajdia, S. (Eds.): *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data* (*SIGMOD-93, ACM Press*), Washington, DC, USA, May 1993, pages 207 – 216.

[Agrawal and Srikant, 1994] Agrawal, R., and Srikant, R. (1994) Fast algorithm for mining association rules. In: Bocca, J.B., Jarke, M., and Zaniolo, C. (Eds.): *Proceedings of the 20th International Conference on Very Large Data Bases* (*VLDB-94, Morgan Kaufmann Publishers*), Santiago de Chile, Chile, September 1994, pages 487 – 499. (ISBN 1-55860-153-8)

[Agarwal *et al.*, 2007] Agarwal, S., Godbole, S., Punjani, D., and Roy, S. (2007) How much noise is too much: A study in automatic text classification. In: Ramakrishnan, N., Zaïane, O.R., Shi, Y., Clifton, C.W., and Wu, X. (Eds.): *Proceedings of the 7th IEEE International Conference on Data Mining* (*ICDM-07, IEEE Computer Society*), Omaha, NE, USA, October 2007, pages 3 – 12. (ISBN 0-7695-3018-4)

[Ahmed *et al.*, 2003] Ahmed, S., Coenen, F., and Leng, P. (2003) Strategies for partitioning data in association rule mining. In: Coenen, F., Preece, A., and Macintosh, A.L. (Eds.): *Research and Development in Intelligent Systems XX – Proceedings of AI-2003, the Twenty-third SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence* (*AI-2003, Springer-Verlag*), Cambridge, UK, December 2003, pages 127 – 139. (ISBN 1-85233-780-X)

[Ahmed, 2004] Ahmed, S. (2004) *Strategies for partitioning data in association rule mining*. Ph.D. Thesis, The University of Liverpool, UK.

[Ahonen-Myka *et al.*, 1999] Ahonen-Myka, H., Heinonen, O., Klemettinen, M., and Verkamo, A.I. (1999) Finding co-occurring text phrases by combining sequence and frequent set discovery. In: *Proceedings of the 1999 IJCAI Workshop on Text Mining: Foundations, Techniques and Applications*, Stockholm, Sweden, July-August 1999, pages 1 – 9.

[Airio *et al.*, 2004] Airio, E., Keskustalo, H., Hedlund, T., and Pirkola, A. (2004) The impact of word normalization methods and merging strategies on multilingual IR. In: Peters, C., Gonzalo, J., Braschler, M., and Kluck, M. (Eds.): *Comparative Evaluation of Multilingual Information Access Systems – Proceedings (Revised Selected Papers) of the 4th Workshop of the Cross-Language Evaluation Forum*

(*CLEF-03, Springer-Verlag*), Trondheim, Norway, August 2003, pages 74 – 84. (LNCS 3237, ISBN 3-540-24017-9)

[Ali *et al.*, 1997] Ali, K., Manganaris, S., and Srikant, R. (1997) Partial classification using association rules. In: Heckerman, D., Mannila, H., Pregibon, D., and Uthurusamy, R. (Eds.): *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (*KDD-97, AAAI Press*), Newport Beach, CA, USA, August 1997, pages 115 – 118. (ISBN 1-57735-027-8)

[Allan, 2002] Allan, J. (2002) *Topic detection and tracking: Event-based information organization.* Kluwer Academic Publishers, Norwell, MA, USA. (ISBN 0792376641)

[Anand *et al.*, 1995] Anand, S.S., Bell, D.A., and Hughes, J.G. (1995) Evidence based discovery of knowledge in databases. *IEE Colloquium on Knowledge Discovery in Databases*, Digest No: 1995/021(A):9/1 – 9/5, London, UK, February 1995.

[Antonie *et al.*, 2001] Antonie, M.-L., Coman, A., and Zaïane, O.R. (2001) Application of data mining techniques for medical image classification. In: *Proceedings of the Second International Workshop on Multimedia Data Mining – a Workshop in Conjunction with the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*MDM/KDD-01, ACM Press*), San Francisco, CA, USA, August 2001, pages 94 – 101.

[Antonie and Zaïane, 2002] Antonie, M.-L., and Zaïane, O.R. (2002) Text document categorization by term association. In: *Proceedings of the 2002 IEEE International Conference on Data Mining* (*ICDM-02, IEEE Computer Society*), Maebashi City, Japan, December 2002, pages 19 – 26. (ISBN 0-7695-1754-4)

[Apte *et al.*, 1994] Apte, C., Damerau, F., and Weiss, S.M. (1994) Towards language independent automated learning of text categorization models. In: Croft, W.B., and Rijsbergen, C. J. van (Eds.): *Annual ACM Conference on Research and Development in Information Retrieval – Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-94, ACM/Springer*), Dublin, Ireland, July 1994, pages 23 – 30. (ISBN 3-540-19889-X)

[Baeza-Yates, 2005] Baeza-Yates, R. (2005) Web usage mining in search engines. In: Scime, A. (Editor): *Web Mining: Applications and Techniques.* Idea Group Inc, pages 307 – 321. (ISBN 1-59140-414-2)

[Baralis and Garza, 2006] Baralis, E., and Garza, P. (2006) Associative text categorization exploiting negated words. In: Haddad, H. (Editor): *Proceedings of the 2006 ACM Symposium on Applied Computing* (*SAC-06, ACM Press*), Dijon, France, April 2006, pages 530 – 535. (ISBN 1-59593-108-2)

[Basili *et al.*, 2004] Basili, R., Serafini, A., and Stellato, A. (2004) Classification of musical genre: A machine learning approach. In: *Proceedings of the 5th*

International Conference on Music Information Retrieval (*ISMIR-04*), Barcelona, Spain, October 2004.

[Baumgarten *et al.*, 1999] Baumgarten, M., Buchner, A.G., Anand, S.S., Mulvenna, M.D., and Hughes, J.G. (1999) User-driven navigating pattern discovery from internet data. In: Masand, B., and Spiliopoulou, M. (Eds.): *Proceedings of the International Workshop on Web Usage Analysis and User Profiling (Revised Papers) – a Workshop in Conjunction of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*WEB/KDD-99, Springer-Verlag*), San Diego, CA, USA, August 1999, pages 74 – 91. (ISBN 3-540-67818-2)

[Bayardo and Agrawal, 1999] Bayardo, R.J., and Agrawal, R. (1999) Mining the most interesting rules. In: *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD-99, ACM Press*), San Diego, CA, USA, August 1999, pages 145 – 154.

[Beeferman *et al.*, 1999] Beeferman, D., Berger, A., and Lafferty, J. (1999) Statistical models for text segmentation. *Machine Learning* 34(1-3): 177 – 210. (ISSN 0885-6125)

[Bel *et al.*, 2003] Bel, N., Koster, C.H.A., and Villegas, M. (2003) Cross-lingual text categorization. In: Koch, T., and Solvberg, I.T. (Eds.): *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries* (*ECDL-03, Springer-Verlag*), Trondheim, Norway, August 2003, pages 126 – 139. (LNCS 2769, ISBN 3-540-40726-X)

[Berger and Merkl, 2004] Berger, H., and Merkl, D. (2004) A comparison of text-categorization methods applied to *n*-gram frequency statistics. In: Webb, G.I., and Yu, X. (Eds.): *Advances in Artificial Intelligence – Proceedings of the 17th Australian Joint Conference on Artificial Intelligence* (*AI-04, Springer-Verlag*), Cairns, Australia, December 2004, pages 998 – 1003. (LNAI 3339, ISBN 3-540-24059-4)

[Berry and Linoff, 1997] Berry, M.J.A., and Linoff, G. (1997) *Data mining techniques for marketing, sales, and customer support*. John Wiley & Sons, Inc., USA. (ISBN 0-471-17980-9)

[Berry, 2004] Berry, M.W. (2004) *Survey of text mining: Clustering, classification, and retrieval*. Springer-Verlag New York, Inc. (ISBN 0-387-95563-1)

[Berry and Castellanos, 2008] Berry, M.W., and Castellanos, M. (2008) *Survey of text mining II: Clustering, classification, and retrieval*. Springer. (ISBN 978-1-84800-045-2)

[Bestgen, 2006] Bestgen, Y. (2006) Improving text segmentation using latent semantic analysis: A reanalysis of Choi, Wiemer-Hastings, and Moore (2001). *Computational Linguistics* 32(1): 5 – 12.

[Bong and Narayanan, 2004] Bong, C.H., and Narayanan, K. (2004) An empirical study of feature selection for text categorisation based on term weightage. In: *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence* (*WI-04, IEEE Computer Society*), Beijing, China, September 2004, pages 599 – 602. (ISBN 0-7695-2100-2)

[Boser *et al.*, 1992] Boser, B.E., Guyon, I.M., and Vapnik, V.N. (1992) A training algorithm for optimal margin classifiers. In: Haussler, D. (Editor): *Proceedings of the 5th ACM Annual Workshop on Computational Learning Theory* (*COLT-92, ACM Press*), Pittsburgh, PA, USA, July 1992, pages 144 – 152. (ISBN 0-89791-497-X)

[Brachman and Anand, 1996] Brachman, R.J., and Anand, T. (1996) The process of knowledge discovery in databases: A human centred approach. In: Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusammy, R. (Eds.): *Advance in Knowledge Discovery and Data Mining*, AAAI/MIT Press, pages 37 – 57. (ISBN 0-262-56097-6)

[Bramer, 2007] Bramer, M. (2007) *Principles of data mining*. Springer-Verlag London Limited. (ISBN 1-84628-765-0)

[Brill, 1992] Brill, E. (1992) A simple rule-based part of speech tagger. In: *Proceedings of the 3rd Conference on Applied Natural Language Processing* (*ANLP-92, Association for Computational Linguistics*), Trento, Italy, April 1992, pages 152 – 155.

[Brin *et al.*, 1997] Brin, S., Motwani, R., Ullman, J.D., and Tsur, S. (1997) Dynamic itemset counting and implication rules for market basket data. In: Peckham, J. (Editor): *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data* (*SIGMOD-97, ACM Press*), Tucson, AZ, USA, May 1997, pages 255 – 264. (SIGMOD Record 26(2))

[Burdick *et al.*, 2001] Burdick, D., Calimlim, M., and Gehrke, J. (2001) MAFIA: A maximal frequent itemset algorithm for transactional databases. In: *Proceedings of the 17th International Conference on Data Engineering* (*ICDE-01, IEEE Computer Society*), Heidelberg, Germany, April 2001, pages 443 – 452. (ISBN 0-7695-1001-9)

[Cardoso-Cachopo and Oliveira, 2006] Cardoso-Cachopo, A., and Oliveira, A.L. (2006) Empirical evaluation of centroid-based models for single-label text categorization. *INESC-ID Technical Report 7/2006*, Instituto Superior Técnico – Universidade Técnica de Lisboa / INESC-ID, Portugal.

[Cardoso-Cachopo, 2007] Cardoso-Cachopo, A. (2007) *Improving methods for single-label text categorization.* Ph.D. Thesis, Instituto Superior Técnico – Universidade Técnica de Lisboa / INESC-ID, Portugal.

[Caropreso *et al.*, 2001] Caropreso, M.F., Matwin, S., and Sebastiani, F. (2001) A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In: Chin, A.G. (Editor): *Text Databases and Document Management: Theory and Practice*, Idea Group Inc., pages 78 – 102. (ISBN 1-878289-93-4)

[Cataltepe *et al.*, 2007] Cataltepe, Z., Yaslan, Y., and Sonmez, A. (2007) Music genre classification using MIDI and audio features. *EURASIP Journal on Advances in Signal Processing* Volume 2007, Article ID 36409, 8 pages.

[Cavnar, 1994] Cavnar, W.B. (1994) Using an N-gram-based document representation with a vector processing retrieval model. In: *Proceedings of the 3rd Text Retrieval Conference* (*TREC-94*), Gaithersburg, MD, USA, 1994, pages 269 – 278.

[Chakrabarti, 2002] Chakrabarti, S. (2002) *Mining the web: Analysis of hypertext and semi structured data*. Morgan Kaufmann Publishers, San Francisco, CA, USA. (ISBN 1558607544)

[Chakrabarti and Faloutsos, 2006] Chakrabarti, D., and Faloutsos, C. (2006) Graph mining: Laws, generators and algorithms. *ACM Computing Surveys* 38(1). (ISSN 0360-0300)

[Chang *et al.*, 2006] Chang, G., Healey, M., Mchugh, J.A.M., Wang, T.L. (2006) *Mining the world wide web – an information search approach*. Kluwer Academic Publishers, Norwell, MA, USA. (ISBN 0-7923-7349-9)

[Church and Hanks, 1989] Church, K.W., and Hanks, P. (1989) Word association norms, mutual information, and lexicography. In: *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics* (*ACL-89, Association for Computational Linguistics*), Vancouver, BC, Canada, 1989, pages 76 – 83.

[Cios *et al.*, 1998] Cios, K.J., Pedrycz, W., and Swiniarski, R.W. (1998) *Data mining methods for knowledge discovery*. Kluwer Academic Publishers, Norwell, MA, USA. (ISBN 0-7923-8252-8)

[Clark and Boswell, 1991] Clark, P., and Boswell, R. (1991) Rule induction with CN2: Some recent improvement. In: *Proceedings of the Fifth European Working Session on Learning* (*EWSL-91, Springer-Verlag*), Porto, Portugal, March 1991, pages 111 – 116.

[Coenen and Leng, 2001] Coenen, F., and Leng, P. (2001) Optimising association rule algorithms using itemset ordering. In: Bramer, M., Coenen, F., and Preece, A. (Eds.): *Research and Development in Intelligent Systems XVIII – Proceedings of the Twenty-first SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence* (*ES-01, Springer-Verlag*), Cambridge, UK, December 2001, pages 53 – 66. (ISBN 1-85233-535-1)

[Coenen *et al.*, 2001] Coenen, F., Goulbourne, G., and Leng, P. (2001) Computing association rules using partial totals, In: Readt, L.D., and Siebes, A. (Eds.): *Principles of Data Mining and Knowledge Discovery – Proceedings of the 5th*

European Conference on Principles and Practice of Knowledge Discovery in Databases (*PKDD-01, Springer-Verlag*), Freiburg, Germany, September 2001, pages 54 – 66. (LNAI 2168, ISBN 3-540-42534-9)

[Coenen and Leng, 2002] Coenen, F., and Leng, P. (2002) Finding association rules with some very frequent attributes. In: Elomaa, T., Mannila, H., and Toivonen, H. (Eds.): *Principles of Data Mining and Knowledge Discovery – Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases* (*PKDD-02, Springer-Verlag*), Helsinki, Finland, August 2002, pages 99 – 111. (LNAI 2431, ISBN 3-540-44037-2)

[Coenen and Leng, 2004] Coenen, F., and Leng, P. (2004) An evaluation of approaches to classification rule selection. In: *Proceedings of the 4th IEEE International Conference on Data Mining* (*ICDM-04, IEEE Computer Society*), Brighton, UK, November 2004, pages 359 – 362. (ISBN 0-7695-2142-8)

[Coenen *et al.*, 2004a] Coenen, F., Leng, P., and Ahmed, S. (2004) Data structure for association rule mining: T-trees and p-trees. *IEEE Transactions on Knowledge and Data Engineering* 16(6): 774 – 778.

[Coenen *et al.*, 2004b] Coenen, F., Leng, P., and Goulbourne, G. (2004) Tree structures for mining association rules. *Journal of Data Mining and Knowledge Discovery* 8(1): 25 – 51.

[Coenen *et al.*, 2005] Coenen, F., Leng, P., and Zhang, L. (2005) Threshold tuning for improved classification association rule mining. In: Ho, T.B., Cheung, D., and Liu, H. (Eds.): *Advances in Knowledge Discovery and Data Mining – Proceedings of the Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining* (*PAKDD-05, Springer-Verlag*), Hanoi, Vietnam, May 2005, pages 216 – 225. (LNAI 3518, ISBN 3-540-26076-5)

[Coenen, 2007] Coenen, F. (2007) Association rule mining in the wider context of text, images and graphs. In: Freitas, A.A. (Editor): *Proceedings of the Third UK Knowledge Discovery and Data Mining Symposium* (*UKKDD-07*), Canterbury, UK, April 2007.

[Coenen and Leng, 2007] Coenen, F. and Leng, P. (2007) The effect of threshold values on association rule based classification accuracy. *Journal of Data and Knowledge Engineering* 60(2): 345 – 360.

[Coenen *et al.*, 2007] Coenen, F., Leng, P., Sanderson, R., and Wang, Y.J. (2007) Statistical identification of key phrases for text classification. In: Perner, P. (Editor): *Proceedings of the 5th International Conference on Machine Learning and Data Mining* (*MLDM-07, Springer-Verlag*), Leipzig, Germany, July 2007, pages 838 – 853. (LNAI 4571, ISBN 978-3-540-73498-7)

[Cohen, 1995] Cohen, W.W. (1995) Fast effective rule induction. In: Prieditis, A., and Russell, S.J. (Eds.): *Machine Learning – Proceedings of the Twelfth International Conference on Machine Learning* (*ICML-95, Morgan Kaufmann*

Publishers), Tahoe City, CA, USA, July 1995, pages 115 – 123. (ISBN 1-55860-377-8)

[Cohen and Singer, 1996] Cohen, W.W., and Singer, Y. (1996) Context-sensitive learning methods for text categorization. In: Frei, H.-P., Harman, D., Schauble, R., and Wilkinson, R. (Eds.): *Proceedings of 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-96, ACM Press*), Zurich, Switzerland, August 1996, pages 307 – 315. (ISBN 0-89791-792-8)

[Combarro *et al.*, 2005] Combarro, E.F., Montanes, E., Diaz, I., Ranilla, J., and Mones, R. (2005) Introducing a family of linear measures for feature selection in text categorization. *IEEE Transactions on Knowledge and Data Engineering* 17(9): 1223 – 1232.

[Cook and Holder, 2006] Cook, D.J., and Holder, L.B. (2006) *Mining graph data*. John Wiley & Sons, Inc., Hoboken, NJ, USA. (ISBN 0-471-73190-0)

[Cooley *et al.*, 1997] Cooley, R., Mobasher, B., and Srivastava, J. (1997) Web mining: information and pattern discovery on the world wide web. In: *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence* (*ICTAI-97, IEEE Computer Society*), New Port Beach, CA, USA, November 1997, pages 558 – 567.

[Cornelis *et al.*, 2006] Cornelis, C., Yan, P., Zhang, X., and Chen, G. (2006) Mining positive and negative association rules from large databases. In: *Proceedings of the 2006 IEEE International Conference on Cybernetics and Intelligent Systems* (*CIS-06, IEEE Computer Society*), Bangkok, Thailand, June 2006, pages 613 – 618. (ISBN 1-4244-0023-6)

[Cutting *et al.*, 1992] Cutting, D., Kupiec, J., Pedersen, J., and Sibun, P. (1992) A practical part-of-speech tagger. In: *Proceedings of the 3rd Conference on Applied Natural Language Processing* (*ANLP-92, Association for Computational Linguistics*), Trento, Italy, April 1992, pages 133 – 140.

[Damashek, 1995] Damashek, M. (1995) Gauging similarity with *n*-grams: Language-independent categorization of text. *Science* Volume 267 (Issue 5199): 843 – 848.

[Daniel and Ding, 2004] Daniel, C., and Ding, Q. (2004) A framework for Bayesian classification on banner images. In: *Proceedings of the 5th International Workshop on Multimedia Data Mining – a Workshop in Conjunction of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*MDM/KDD-04, ACM Press*), Seattle, WA, USA, August 2004, pages 61 – 66.

[De Veaux and Hand, 2005] De Veaux, R.D., and Hand, D.J. (2005) How to lie with bad data? *Statistical Science* 20(3): 231 – 238.

[Delgado *et al.*, 2002] Delgado, M., Martin-Bautista, M.J., Sanchez, D., and Vila, M.A. (2002) Mining text data: Special features and patterns. In: Hand, D.J., Adams, N.M., and Bolton, R.J. (Eds.): *Pattern Detection and Discovery – Proceedings of ESF Exploratory Workshop* (*Springer-Verlag*), London, UK, September 2002, pages 140 – 153. (LNAI 2447, ISBN 3-540-44148-4)

[Deng *et al.*, 2002] Deng, Z.-H., Tang, S.-W., Yang, D.-Q., Zhang, M., Wu, X.-B., and Yang, M. (2002) Two odds-radio-based text classification algorithms. In: Huang, B., Ling, T.W., Mohania, M.K., Ng, W.K., Wen, J.-R., and Gupta, S.K. (Eds.): *Proceedings of the Third International Conference on Web Information Systems Engineering workshop* (*WISEw-02, IEEE Computer Society*), Singapore, December 2002, pages 223 – 231. (ISBN 0-7695-1813-3)

[Dhillon *et al.*, 2004] Dhillon, I., Kogan, J., and Nicholas, C. (2004) Feature selection and document clustering. In: Berry, M.W. (Editor): *Survey of text mining: Clustering, classification, and retrieval*, Springer-Verlag New York, Inc., pages 73 – 100. (ISBN 0-387-95563-1)

[Diestel, 2005] Diestel, R. (2005) *Graph theory* (*Third Edition*). Springer-Verlag Berlin Heidelberg, Germany. (ISBN 3-540-26182-6)

[Ding *et al.*, 2002] Ding, Q., Ding, Q., and Perrizo, W. (2002) Association rule mining on remotely sensed images using P-trees. In: Cheng, M.-S., Yu, P.S., and Liu, B. (Eds.): *Advances in Knowledge Discovery and Data Mining – Proceedings of the 6th Pacific-Asia Conference* (*PAKDD-02, Springer-Verlag*), Taipei, Taiwan, May 2002, pages 66 – 79. (LNCS 2336, ISBN 3-540-43704-5)

[Domingos and Pazzani, 1997] Domingos, P., and Pazzani, M. (1997) On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29(2/3): 103 – 130.

[Dong and Li, 1999] Dong, G., and Li, J. (1999) Efficient mining of emerging patterns: Discovering trends and differences. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD-99, ACM Press*), San Diago, CA, USA, August 1999, pages 43 – 52.

[Dong *et al.*, 1999] Dong, G., Zhang, X., Wong, L., and Li, J. (1999) CAEP: Classification by aggregating emerging patterns. In: Arikawa, S., and Furukawa, K. (Eds.): *Discovery Science – Proceedings of the Second International Conference on Discovery Science* (*DS-99, Springer-Verlag*), Tokyo, Japan, December 1999, pages 30 – 42. (LNAI 1721, ISBN 3-540-66713-X)

[Dunham, 2002] Dunham, M.H. (2002) *Data mining: Introductory and advanced topics*. Prentice Hall. (ISBN 0130888923)

[El-Hajj and Zaïana, 2003] El-Hajj, M., and Zaïana, O.R. (2003) Inverted matrix: Efficient discovery of frequent items in large datasets in the context of interactive mining. In: Getoor, L., Senator, T.E., Domingos, P., and Faloutsos, C. (Eds.): *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge*

Discovery and Data Mining (*KDD-03, ACM Press*), Washington, DC, USA, August 2003, pages 109 – 118. (ISBN 1-58113-737-0)

[Fan *et al.*, 2003] Fan, J., Gao, Y., Luo, H., and Hacid, M.-S. (2003) A novel framework for semantic image classification and benchmark. In: *Proceedings of the 4th International Workshop on Multimedia Data Mining – a Workshop in Conjunction of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*MDM/KDD-03, ACM Press*), Washington, DC, USA, August 2003.

[Fano, 1961] Fano, R.M. (1961) *Transmission of information – A statistical theory of communication*. The MIT Press. (ISBN 0-262-56169-7)

[Fayyad *et al.*, 1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996) Knowledge discovery and data mining: Towards a unifying framework. In: Simoudis, E., Han, J., and Fayyad, U.M. (Eds.): *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (*KDD-96, AAAI Press*), Portland, OR, USA, August 1996, pages 82 – 95. (ISBN 1-57735-004-9)

[Feldman and Sanger, 2006] Feldman, R., and Sanger, J. (2006) *The text mining handbook: Advanced approaches in analyzing unstructured data*. Cambridge University Press. (ISBN 0521836573)

[Feng, *et al.*, 2005] Feng, Y., Wu, Z., and Zhou, Z. (2005) Multi-label text categorization using K-nearest neighbor approach with M-similarity. In: Consens, M.P., and Navarro, G. (Eds.): *Proceedings of the 12th International Conference on String Processing and Information Retrieval* (*SPIRE-05, Springer-Verlag*), Buenos Aires, Argentina, November 2005, pages 155 – 160. (LNCS 3772, ISBN 3-540-29740-5)

[Forman, 2003] Forman, G. (2003) An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research* 3: 1289 – 1305.

[Fragoudis *et al.*, 2005] Fragoudis, D., Meretaskis, D., and Likothanassis, S. (2005) Best terms: An efficient feature-selection algorithm for text categorization. *Knowledge and Information Systems* 8(1): 16 – 33. (ISSN 0219-1377)

[Frawley *et al.*, 1991] Frawley, W.J., Piatetsky-Shapiro, G., and Matheus, C.J. (1991) Knowledge discovery in databases: An overview. In: Piatetsky-Shapiro, G., and Frawley, W.J. (Eds.): *Knowledge Discovery in Databases*, AAAI/MIT Press, pages 1 – 27. (ISBN 0262660709)

[Freitas, 2002] Freitas, A.A. (2002) *Data mining and knowledge discovery with evolutionary algorithms*. Springer-Verlag Berlin Heidelberg New York, Germany. (ISBN 3-540-4331-7)

[Freitas, 2006] Freitas, A.A. (2006) Are we really discovering “interesting” knowledge from data? In: Smith, G.D. (Editor): *Proceedings of the Second UK Knowledge Discovery and Data Mining Symposium* (*UKKDD-06*), Norwich, UK, April 2006.

[Fuhr, 1989] Fuhr, N. (1989) Models for retrieval with probabilistic indexing. *Information Processing and Management* 25(1): 55 – 72.

[Fuhr and Buckley, 1991] Fuhr, N., and Buckley, C. (1991) A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems* 9(3): 223 – 248.

[Fürnkranz, 1998] Fürnkranz, J. (1998) A study using n-gram features for text categorization. *Technical Report OEFAI-TR-98-30*, Austrian Research Institute for Artificial Intelligence, Austria.

[Gagnon and Sylva, 2005] Gagnon, M., and Sylva, L.D. (2005) Text summarization by sentence extraction and syntactic pruning. In: *Proceedings of the 3rd Computational Linguistics in the North-East Workshop* (*CLiNE-05*), Gatineau, Quebec, Canada, August 2005.

[Galavotti *et al.*, 2000] Galavotti, L., Sebastiani, F., and Simi, M. (2000) Experiments on the use of feature selection and negative evidence in automated text categorization. In: Borbinha, J.L., and Baker, T. (Eds.): *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries* (*ECDL-00, Springer-Verlag*), Lisbon, Portugal, September 2000, pages 59 – 68. (LNCS 1923, ISBN 3-540-41023-6)

[Getoor and Diehl, 2005] Getoor, L., and Diehl, C.P. (2005) Link mining: A survey. *ACM SIGKDD Explorations Newsletter* 7(2): 3 – 12.

[Giorgetti and Sebastiani, 2003] Giorgetti, D., and Sebastiani, F. (2003) Multiclass text categorization for automated survey coding. In: *Proceedings of the 2003 ACM Symposium on Applied Computing* (*SAC-03, ACM Press*), Melbourne, FL, USA, March 2003, pages 798 – 802.

[Goncalves and Quaresma, 2004] Goncalves, T., and Quaresma, P. (2004) Using IR techniques to improve automated text classification. In: Meziane, F., and Metais, E. (Eds.): *Natural Language Processing and Information Systems – Proceedings of the 9th International Conference on Applications of Natural Languages to Information Systems* (*NLDB-04, Springer-Verlag*), Salford, UK, June 2004, pages 374 – 379. (LNCS 3136, ISBN 3-540-22564-1)

[Goncalves and Quaresma, 2005] Goncalves, T., and Quaresma, P. (2005) Enhancing a Portuguese text classifier using part-of-speech tags. *Intelligent Information Systems* 2005: 189 – 198.

[Gouda and Zaki, 2001] Gouda, K., and Zaki, M.J. (2001) Efficiently mining maximal frequent itemsets. In: Cercone, N., Lin, T.Y., and Wu, X. (Eds.):

Proceedings of the 2001 IEEE International Conference on Data Mining (*ICDM-01, IEEE Computer Society*), San Jose, CA, USA, November-December 2001, pages 163 – 170. (ISBN 0-7695-1119-8)

[Goulbourne *et al.*, 2000] Goulbourne, G., Coenen, F., and Leng, P. (2000) Algorithms for computing association rules using a partial-support tree. *Journal of Knowledge-Based Systems* 13: 141 – 149.

[Hájek *et al.*, 1966] Hájek, P., Havel, I., and Chytil, M. (1966) The GUHA method of automatic hypotheses determination. *Computing* 1: 293 – 308.

[Halevy, 2001] Halevy, A.Y. (2001) Answering queries using views: A survey. *International Journal on Very Large Data Bases* 10(4): 270 – 294.

[Hamdi, 2005] Hamdi, M.S. (2005) Extracting and customizing information using multi-agents. In: Scime, A. (Editor): *Web Mining: Applications and Techniques*. Idea Group Inc., pages 228 – 252. (ISBN 1-59140-414-2)

[Hammouda and Kamel, 2003] Hammouda, K.M., and Kamel, M.S. (2003) Incremental document clustering using cluster similarity histograms. In: *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence* (*WI-03, IEEE Computer Society*), Halifax, Canada, October 2003, pages 597 – 601. (ISBN 0-7695-1932-6)

[Han *et al.*, 1992] Han, J., Cai, Y., and Cercone, N. (1992) Knowledge discovery in databases: An attribute-oriented approach. In: Yuan, L.-Y. (Editor): *Proceedings of the 18th International Conference on Very Large Data Bases* (*VLDB-92, Morgan Kaufmann Publishers*), Vancouver, British Columbia, Canada, August 1992, pages 547 – 559. (ISBN 1-55860-151-1)

[Han *et al.*, 2000] Han, J., Pei, J., and Yin, Y. (2000) Mining frequent patterns without candidate generation. In: Chen, W., Naughton, J.F., and Bernstein, P.A. (Eds.): *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (*SIGMOD-00, ACM Press*), Dallas, TX, USA, May 2000, pages 1 – 12. (ISBN 1-58113-218-2)

[Han and Kamber, 2001] Han, J., and Kamber, M. (2001) *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers, Dan Francisco, CA, USA. (ISBN 1-55860-489-8)

[Han and Kamber, 2006] Han, J., and Kamber, M. (2006) *Data mining: Concepts and techniques* (*Second Edition*). Morgan Kaufmann Publishers, San Francisco, CA, USA. (ISBN 1-55860-901-6)

[Hand *et al.*, 2001] Hand, D., Mannila, H., and Smyth, R. (2001) *Principles of data mining*. The MIT Press. (ISBN 0-262-08290-X)

[Harrism, 1968] Harrism, Z.S. (1968) *Mathematical structures of language* (*Interscience tracts in pure and applied mathematics*). Wiley-Interscience Publishers, New York, USA. (ISBN 0470353163)

[Havre *et al.*, 2001] Havre, S., Hetzler, E., Perrine, K., Jurrus, E., and Miller, N. (2001) Interactive visualization of multiple query result. In: *Proceedings of the IEEE Symposium on Information Visualization* (*INFOVIS-01, IEEE Computer Society*), San Diego, CA, USA, October 2001, pages 105 – 112.

[Hearst and Karadi, 1997] Hearst, M.A., and Karadi, C. (1997) Cat-a-cone: An interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In: *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-97, ACM Press*), Philadelphia, PA, USA, July 1997, pages 246 – 255.

[Hersh *et al.*, 1994] Hersh, W.R., Buckley, C., Leone, T.J., and Hickman, D.H. (1994) OHSUMED: An interactive retrieval evaluation and new large test collection for research. In: Croft, W.B., and van Rijsbergen, C.J. (Eds.): *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-94, ACM/Springer*), Dublin, Ireland, July 1994, pages 192 – 201. (ISBN 3-540-19889-X)

[Hidber, 1999] Hidber, C. (1999) Online association rule mining. In: Delis, A., Faloutsos, C., and Ghandeharizadeh, S. (Eds.): *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (*SIGMOD-99, ACM Press*), Philadelphia, PA, USA, June 1999, pages 145 – 156. (ISBN 1-58113-084-8)

[Holsheimer *et al.*, 1995] Holsheimer, M., Kersten, M.L., Mannila, H., and Toivonen, H. (1995): A perspective on databases and data mining. In: Fayyad, U.M., and Uthurusamy, R. (Eds.): *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (*KDD-95, AAAI Press*), Montreal, Canada, August 1995, pages 150 – 155. (ISBN 0-929280-82-2)

[Honda and Konoshi, 2000] Honda, R., and Konoshi, O. (2000) Semantic indexing and temporal rule discovery for time-series satellite images. In: *Proceedings of the 2000 Workshop on Multimedia Data Mining – a Workshop in Conjunction with the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*MDM/KDD-00, ACM Press*), Boston, MA, USA, August 2000, pages 82 – 90.

[Hotho *et al.*, 2005] Hotho, A., Nürnberger, A., and Paaß, G. (2005) A brief survey of text mining. *LDV Forum – GLDV Journal for Computational Linguistics and Language Technology* 20(1): 19 – 62. (ISSN 0175-1336)

[Houtsma and Swami, 1995] Houtsma, M., and Swami, A. (1995) Set-oriented mining of association rules in relational databases. In: Yu, P.S., and Chen, A.L. (Eds.): *Proceedings of the Eleventh International Conference on Data Engineering* (*ICDE-95, IEEE Computer Society*), Taipei, Taiwan, March 1995, pages 25 – 33. (ISBN 0-8186-6910-1)

[Hovy, 2003] Hovy, E. (2003) Text summarization. In: Mitkov, R. (Editor): *The Oxford Handbook of Computational Linguistics*, Oxford University Press Inc., New York, USA, pages 583 – 598. (ISBN 0-19-823882-7)

[Hsu *et al.*, 2002] Hsu, W., Lee, M.L., and Zhang, J. (2002) Image mining: Trends and developments. *Journal of Intelligent Information Systems* 19(1): 7 – 23.

[Hull, 1997] Hull, R. (1997) Managing semantic heterogeneity in database: A theoretical perspective. In: *Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (*PODS-97, ACM Press*), Tucson, AZ, USA, May 1997, pages 51 – 61. (ISBN 0-89791-910-6)

[Hulth and Megyesi, 2006] Hulth, A., and Megyesi, B.B. (2006) A study on automatically extracted keywords in text categorization. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (*ACL-06, Association for Computer Linguistics*), Sydney, Australia, July 2006, pages 537 – 544.

[Ide and Veronis, 1998] Ide, N., and Veronis, J. (1998) Word sense disambiguation: The state of the art. *Computational Linguistics* 24(1): 1 – 40.

[James, 1985] James, M. (1985) *Classification algorithms*. Wiley-Interscience, New York, NY, USA. (ISBN 0-471-84799-2)

[Joachims, 1996] Joachims, T. (1996) A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *CMU-CS-96-118 – Technical Report*, School of Computer Science, Carnegie Mellon University, USA.

[Joachims, 1998] Joachims, T. (1998) Text categorization with support vector machines: Learning with many relevant features. *LS-8 Report 23 – Research Reports of the Unit no. VIII (AI)*, Computer Science Department, University of Dortmund, Germany.

[Katrenko, 2004] Katrenko, S. (2004) Textual data categorization: Back to phrase-based representation. In: *Proceedings of the Second IEEE International Conference on Intelligent Systems* (*IS-04, IEEE Computer Society*), June 2004, pages 64 – 66.

[Kelly and Stone, 1975] Kelly, E.F., and Stone, P.J. (1975) *Computer recognition of English word senses*. North-Holland, Amsterdam. (ISBN 0-444-10831-9)

[Klein, 1998] Klein, B.D. (1998) Data quality in the practice of consumer product management: evidence from the field. *Data Quality Journal* 4(1).

[Klemettinen *et al.*, 1994] Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A.I. (1994) Finding interesting rules from large sets of discovered association rules. In: *Proceedings of the Third International Conference on Information and Knowledge Management* (*CIKM-94, ACM Press*), Gaitherburg, MD, USA, November-December 1994, pages 401 – 407.

[Kobayashi and Aono, 2004] Kobayashi, M., and Aono, M. (2004) Vector space models for search and cluster mining. In: Berry, M.W. (Editor): *Survey of Text Mining – Clustering, Classification, and Retrieval*, Springer-Verlag New York, Inc., pages 103 – 122. (ISBN 0-387-95563-1)

[Kosala and Blockeel, 2000] Kosala, R., and Blockeel, H. (2000) Web mining research: A survey. *ACM SIGKDD Explorations* 2(1): 1 – 15.

[Kovalerchun and Vityaev, 2000] Kovalerchun, B., and Vityaev, E. (2000) *Data mining in finance: Advances in relational and hybrid methods*. Kluwer Academic Publisher. (ISBN 0-7923-7804-0)

[Kozima, 1993] Kozima, H. (1993) Text segmentation based on similarity between words. In: *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics* (*ACL-93, Association for Computational Linguistics*), Columbus, OH, USA, June 1993, pages 286 – 288.

[Kozima and Furugori, 1993] Kozima, H., and Furugori, T. (1993) Similarity between words computed by spreading activation on an English dictionary. In: *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics* (*EACL-93*), Utrecht, The Netherlands, April 1993, pages 232 – 239.

[Lam and Ho, 1998] Lam, W., and Ho, C.Y. (1998) Using a generalized instance set for automatic text categorization. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-98, ACM Press*), Melbourne, Australia, August 1998, pages 81 – 89.

[Lang, 1995] Lang, K. (1995) NewsWeeder: Learning to filter netnews. In: Prieditis, A., and Russell, S.J. (Eds.): Machine Learning – *Proceedings of the Twelfth International Conference on Machine Learning* (*ICML-95, Morgan Kaufmann Publishers*), Tahoe City, CA, USA, July 1995, pages 331 – 339. (ISBN 1-55860-377-8)

[Lavrac *et al.*, 1999] Lavrac, N., Flach, P., and Zupan, B. (1999) Rule evaluation measures: A unifying view. In: Dzeroski, S., and Flach, P.A. (Eds.): *Proceedings of the 9th International Workshop on Inductive Logic Programming* (*ILP-99, Springer-Verlag*), Bled, Slovenia, June 1999, pages 174 – 185. (LNCS 1634, ISBN 3-540-66109-3)

[Lawrence and Giles, 1999] Lawrence, S., and Giles, C.L. (1999) Accessibility of information on the web. *Nature* 400: 107 – 109.

[Lenzerini, 2002] Lenzerini, M. (2002) Data integration: A theoretical perspective. In: Popa, L. (Editor): *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (*PODS-02, ACM Press*), Madison, WI, USA, June 2002, pages 233 – 246.

[Leopold and Kindermann, 2002] Leopold, E., and Kindermann, J. (2002) Text categorization with support vector machine. How to represent texts in input space? *Machine Learning* 46(2002): 423 – 444.

[Lewis, 1992] Lewis, D.D. (1992) An evaluation of phrasal and clustered representations on a text categorization task. In: Belkin, N.J., Ingwersen, P., and Pejtersen, A.M. (Eds.): *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-92, ACM Press*), Copenhagen, Denmark, June 1992, pages 37 – 50. (ISBN 0-89791-523-2)

[Lewis and Ringuette, 1994] Lewis, D.D., and Ringuette, M. (1994) Comparison of two learning algorithms for text categorization. In: *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval* (*SDAIR-94*), Las Vegas, NV, USA, April 1994, pages 81 – 93.

[Lewis *et al.*, 1996] Lewis, D.D., Schapire, R.E., Callan, J.P., and Papka, R. (1996) Training algorithms for linear text classifiers. In: Frei, H.-P., Harman, D., Schauble, P., and Wilkinson, R. (Eds.): *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-96, ACM Press*), Zurich, Switzerland, August 1996, pages 298 – 306. (ISBN 0-89791-792-8)

[Li *et al.*, 2001] Li, W., Han, J., and Pei, J. (2001) CMAR: Accurate and efficient classification based on multiple class-association rules. In: Cercone, N., Lin, T.Y., and Wu, X. (Eds.): *Proceedings of the 2001 IEEE International Conference on Data Mining* (*ICDM-01, IEEE Computer Society*), San Jose, CA, USA, November-December 2001, pages 369 – 376. (ISBN 0-7695-1119-8)

[Li and Liu, 2003] Li, X., and Liu, B. (2003) Learning to classify texts using positive and unlabeled data. In: Gottlob, G., and Walsh, T. (Eds.): *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence* (*IJCAI-03, Morgan Kaufmann Publishers*), Acapulco, Mexico, August 2003, pages 587 – 594.

[Liao *et al.*, 2003] Liao, C., Alpha, S., and Dixon, P. (2003) Feature preparation in text categorization. In: *Proceedings of the 2nd Australasian Data Mining Workshop* (*ADM-03*), Canberra, Australia, December 2003.

[Lin and Kedem, 1998] Lin, D.-I., and Kedem, Z.M. (1998) Pincer search: A new algorithm for discovering the maximum frequent set. In: Schek, H.-J., Saltor, F., Ramos, I., and Alonso, G. (Eds.): *Advances in Database technology – Proceedings of the 6th International Conference on Extending Database Technology* (*EDBT-98, Springer-Verlag*), Valencia, Spain, March 1998, pages 105 – 119. (LNAI 1377, ISBN 3-540-64264-1)

[Lin *et al.*, 2003] Lin, J., Vlachos, M., Keogh, E., and Gunopulos, D. (2003) Multi-resolution k-means clustering of time series and application to images. In: *Proceedings of the 4th International Workshop on Multimedia Data Mining – a Workshop in Conjunction of the Ninth ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining (*MDM/KDD-03, ACM Press*), Washington, DC, USA, August 2003.

[Lin *et al.*, 2004] Lin, C.-R., Liu, N.-H., Wu, Y.-H., and Chen, A.L.P. (2004) Music classification using significant repeating patterns. In: Lee, Y.-J., Li, J., Whang, K.-Y., and Lee, D. (Eds.): *The 9th International Conference on Database Systems for Advanced Applications* (*DASFAA-2004, Springer-Verlag*), Jeju Island, Korea, March 2004, pages 506 – 518. (LNCS 2973, ISBN 3-540-21047-4)

[Liu *et al.*, 1998] Liu, B., Hsu, W., and Ma, Y. (1998) Integrating classification and association rule mining. In: Agrawal, R., Stolorz, P.E., and Piatetsky-Shapiro, G. (Eds.): *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (*KDD-98, AAAI Press*), New York, NY, USA, August 1998, pages 80 – 86. (ISBN 1-57735-070-7)

[Liu *et al.*, 2002] Liu, J., Pan, Y., Wang, K., and Han, J. (2002) Mining frequent item sets by opportunistic projection. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD-02, ACM Press*), Edmonton, Alberta, Canada, July 2002, pages 229 – 238. (ISBN 1-58113-567-X)

[Liu *et al.*, 2004] Liu, B., Li, X., Lee, W.S., and Yu, P.S. (2004) Text classification by labeling words. In: McGuinness, D.L., and Ferguson, G. (Eds.): *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence* (*AAAI/IAAI-04, AAAI/MIT Press*), San Jose, CA, USA, July 2004, pages 425 – 430. (ISBN 0-262-51183-5)

[Liu *et al.*, 2005] Liu, N., Zhang, B., Yan, J., Chen, Z., Liu, W., Bai, F., and Chien, L. (2005) Text representation: From vector to tensor. In: Han, J., Wah, B.W., Raghavan, V., Wu, X., and Rastogi, R. (Eds.): *Proceedings of the 5th IEEE International Conference on Data Mining* (*ICDM-05, IEEE Computer Society*), Houston, TX, USA, pages 725 – 728. (ISBN 0-7695-2278-5)

[Liu, 2007] Liu, B. (2007) *Web data mining: Exploring hyperlinks, contents, and usage data*. Springer-Verlag Berlin Heidelberg. (ISBN 3-540-37881-2)

[Lodhi *et al.*, 2002] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002) Text classification using string kernels. *Journal of Machine Learning Research* 2: 419 – 444.

[Lowd and Domingos, 2005] Lowd, D., and Domingos, P. (2005) Naïve bayes models for probability estimation. In: Raedt, L.D., and Wrobel, S. (Eds.): *Proceedings of the 22nd International Conference on Machine Learning* (*ICML-05, ACM Press*), Bonn, Germany, August 2005, pages 529 – 536. (ISBN 1-59593-180-5)

[Mani, 2001] Mani, I. (2001) *Automatic summarization*. John Benjamins Publishing Company. (ISBN 1588110605)

[Mani and Maybury, 1999] Mani, I., and Maybury, M.T. (1999) *Advances in automatic text summarization*. The MIT Press. (ISBN 0-262-13359-8)

[Mannila *et al.*, 1994] Mannila, H., Toivonen, H., and Verkamo, A.I. (1994) Efficient algorithms for discovering association rules. In: Fayyad, U.M., and Uthurusamy, R. (Eds.): *Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop* (*KDD-1994, AAAI Press*), Seattle, WA, USA, July 1994, pages 181 – 192. (ISBN 0-929280-73-3)

[Maron, 1961] Maron, M.E. (1961) Automatic indexing: An experimental inquiry. *Journal of the ACM* (*JACM*) 8(3): 404 – 417. (ISSN 0004-5411)

[McElligott and Sorensen, 1993] McElligott, M., and Sorensen, H. (1993) An emergent approach to information filtering. *Abakus. U.C.C. Computer Science Journal* 1(4): 1 – 19.

[McKay and Fujinaga, 2004] McKay, C., and Fujinaga, I. (2004) Automatic genre classification using large high-level musical features sets. In: *Proceedings of the 5th International Conference on Music Information Retrieval* (*ISMIR-04*), Barcelona, Spain, October 2004.

[Melucci and Orio, 2003] Melucci, M., and Orio, N. (2003) A novel method for stemmer generation based on hidden markov models. In: *Proceedings of the 2003 ACM International Conference on Information and Knowledge Management* (*CIKM-03, ACM Press*), New Orleans, LA, USA, November 2003, pages 131 – 138. (ISBN 1-58113-723-0)

[Michalski, 1980] Michalski, R.S. (1980) Pattern recognition as rule-guided inductive inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1980: 774 – 778.

[Michalski *et al.*, 1983] Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (1983) *Machine learning: An artificial intelligence approach*. Tioga Publishing Company, Palo Alto, CA, USA. (ISBN 0-935382-05-4)

[Michalski *et al.*, 2006] Michalski, R.S., Mitchel, T.W., and Mitchell, T.M. (2006) *Machine learning: An artificial intelligence approach* (*Volume I*), Morgan Kaufmann Publishers, San Francisco, CA, USA. (ISBN 0934613095)

[Miller and Han, 2001] Miller, H.J., and Han, J. (Eds.) (2001) *Geographic data mining and knowledge discovery*. Taylor & Francis. (ISBN 0-415-23369-0)

[Mitchell, 1997] Mitchell, T.M. (1997) *Machine learning*. McGraw-Hill, New York, USA. (ISBN 0-07-042807-7)

[Mladenic, 1998] Mladenic, D. (1998) *Machine learning on non-homogeneous, distributed text data*. Ph.D. Thesis, University of Ljubljana, Slovenia.

[Mladenic, 1999] Mladenic, D. (1999) Text-learning and related intelligent agents: A survey. *IEEE Intelligent Systems* 14(4): 44 – 54.

[Moore and McCabe, 1998] Moore, D.S., and McCabe, G.P. (1998) *Introduction to the practice of statistics* (*Third Edition*). W. H. Freeman and Company, USA. (ISBN 0-7167-3502-4)

[Moschitti and Basili, 2004] Moschitti, A., and Basili, R. (2004) Complex linguistic features for text classification: A comprehensive study. In: McDonald, S., and Tait, J. (Eds.): *Advances in Information Retrieval – Proceedings of the 26th European Conference on IR Research* (*ECIR-04, Springer-Verlag*), Sunderland, UK, April 2004, pages 181 – 196. (LNCS 2997, ISBN 3-540-21382-1)

[Namburu *et al.*, 2005] Namburu, S.M., Tu, H., Luo, J., and Pattipati, K.R. (2005) Experiments on supervised learning algorithms for text categorization. In: *Proceedings of the 2005 IEEE Aerospace Conference* (*IEEE Computer Society*), March 2005, pages 1 – 8. (ISBN 0-7803-8870-4)

[Navigli, 2005] Navigli, R. (2005) Ontology learning from a domain web corpus. In: Scime, A. (Editor): *Web Mining: Applications and Techniques*. Idea Group Inc., pages 69 – 98. (ISBN 1-59140-414-2)

[Ng *et al.*, 1997] Ng, H.T., Goh, W.B., and Low, K.L. (1997) Feature selection, perceptron learning, and a usability case study for text categorization. In: *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-97, ACM Press*), Philadelphia, PA, USA, July 1997, pages 67 – 73. (ISBN 0-89791-836-3)

[Nigam and McCallum, 1998] Nigam, K., and McCallum, A. (1998) Pool-based active learning for text classification. In *Proceedings of the Conference on Automated Learning and Discovery* (*CONALD-98*), Pittsburgh, PA, USA, June 1998.

[Nomoto and Matsumoto, 2001] Nomoto, T., and Matsumoto, Y. (2001) A new approach to unsupervised text summarization. In: Croft, W.B., Harper, D.J., Kraft, D.H., and Zobel, J. (Eds.): *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-01, ACM Press*), New Orleans, LA, USA, September 2001, pages 26 – 34. (ISBN 1-58113-331-6)

[Olson, 2003] Olson, C.F. (2003) Image mining by matching exemplars using entropy. In: *Proceedings of the 4th International Workshop on Multimedia Data Mining – a Workshop in Conjunction of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*MDM/KDD-03, ACM Press*), Washington, DC, USA, August 2003.

[Ozgur *et al.*, 2005] Ozgur, A., Ozgur, L., and Gungor, T. (2005) Text categorization with class-based and corpus-based keyword selection. In: Yolum, P., Gungor, T., Gurgen, F.S., and Ozturan, C.C. (Eds.): *Proceedings of the 20th*

International Symposium on Computer and Information Sciences (*ISCIS-05, Springer-Verlag*), Istanbul, Turkey, October 2005, pages 606 – 615. (LNCS 3733, ISBN 3-540-29414-7)

[Pachet *et al.*, 2001] Pachet, F., Westermann, G., and Laigre, D. (2001) Musical data mining for electronic music distribution. In: *Proceedings of the Fist International Conference on WEB Delivering of Music* (*WEDELMUSIC-01, IEEE Computer Society*), Florence, Italy, November 2001, pages 101 – 106. (ISBN 0769512844)

[Pan *et al.*, 2005] Pan, H., Li, J., and Wei, Z. (2005) Mining interesting association rules in medical images. In: Li, X., Wang, S., and Dong, Z.Y. (Eds.): *Proceedings of the First International Conference of Advanced Data Mining and Applications* (*ADMA-05, Springer-Verlag*), Wuhan, China, July 2005, pages 598 – 609. (LNCS 3584, ISNB 3-540-37025-0)

[Park *et al.*, 1995] Park, J.S., Chen, M.-S., and Yu, P.S. (1995) An effective hash-based algorithm for mining association rules. In: Carey, M.J., and Schneider, D.A. (Eds.): *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data* (*SIGMOD-95, ACM Press*), San Jose, CA, USA, May 1995, pages 175 – 186.

[Pei *et al.*, 2000] Pei, J., Han, J., and Mao, R. (2000) CLOSET: An efficient algorithm for mining frequent closed itemsets. In: Gunopulos, D., and Rastogi, R. (Eds.): *Proceedings of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (*SIGMOD-DMKD-01*), Dallas, TX, USA, May 2000, pages 21 – 30.

[Peng and Schuurmans, 2003] Peng, F., and Schuurmans, D. (2003) Combining naïve bayes and *n*-gram language models for text classification. In: Sebastiani, F. (Editor): *Advances in Information Retrieval – Proceedings of the 25th European Conference on IR Research* (*ECIR-03, Springer-Verlag*), Pisa, Italy, April 2003, pages 335 – 350. (LNCS 2633, ISBN 3-540-01274-5)

[Peng *et al.*, 2003] Peng, F., Schuurmans, D., and Wang, S. (2003) Language and task independent text categorization with simple language models. In: *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (*HLT-NAACL-03*), Edmonton, Canada, May-June 2003, pages 110 – 117.

[Piatetsky-Shapiro and Frawley, 1991] Piatetsky-Shapiro, G., and Frawley, W.J. (1991) *Knowledge discovery in databases*. AAAI/MIT Press. (ISBN 0262660709)

[Piatetsky-Shapiro, 2000] Piatetsky-Shapiro, G. (2000) Knowledge discovery in databases: 10 years after. *ACM SIGKDD Explorations* 1(2): 59 – 61.

[Plisson *et al.*, 2004] Plisson, J., Lavrac, N., and Mladenic, D. (2004) A rule based approach to word lemmatization. In: *Proceedings of the 2004 Conference on Data*

Mining and Warehouses (*SiKDD-04*) *– Held at the 7th International Multi-conference on Information Society* (*IS-04*), Ljubljana, Slovenia, October 2004.

[Porter, 1980] Porter, M. (1980) An algorithm for suffix stripping. *Program* 14(3): 130 – 137.

[Quinlan, 1993] Quinlan, J.R. (1993) *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, San Francisco, CA, USA. (ISBN 1-55860-238-0)

[Quinlan and Cameron-Jones, 1993] Quinlan, J.R., and Cameron-Jones, R.M. (1993) FOIL: A midterm report. In: Brazdil, P. (Editor): *Proceedings of the 1993 European Conference on Machine Learning* (*ECML-93, Springer-Verlag*), Vienna, Austria, April 1993, pages 3 – 20. (LNCS 667, ISBN 3-540-56602-3)

[Raghavan, 2005] Raghavan, S.N.R. (2005) Data mining in e-commerce: A survey. *Sadhana* 30(2&3): 275 – 289.

[Rajaraman and Tan, 2001] Rajaraman, K., and Tan, A.-H. (2001) Topic detection, tracking, and trend analysis using self-organizing neural networks. In: Cheung, D., Williams, G.J., and Li, Q. (Eds.): *Knowledge Discovery and Data Mining – Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (*PAKDD-05, Springer-Verlag*), Hong Kong, China, April 2001, pages 102 – 107. (LNAI 2035, ISBN 3-540-41910-1)

[Rajman and Besancon, 1998] Rajman, M., and Besancon, R. (1998) Text mining: Natural language techniques and text mining applications. In: Spaccapietra, S., and Maryanski, F.J. (Eds.): *Data Mining and Reverse Engineering: Searching for Semantics, IFIP TC2/WG2.6 – Proceedings of the 7th IFIP Working Conference on Database Semantics* (*DS-7, Chapam & Hall*), Leysin, Switzerland, October 1997, pages 50 – 65. (ISBN 0-412-82250-4)

[Rish, 2001] Rish, I. (2001) An empirical study of the naïve bayes classifier. In: *Proceedings of the 2001 IJCAI Workshop on Empirical Methods in Artificial Intelligence*, Seattle, WA, USA, August 2001.

[Roberto and Bayardo, 1998] Roberto, J., and Bayardo, Jr. (1998) Efficiently mining long patterns from databases. In: Hass, L.M., and Tiwary, A. (Eds.): *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data* (*SIGMOD-98, ACM Press*), Seattle, WA, USA, June 1998, pages 85 – 93. (ISBN 0-89791-995-5)

[Rolland and Ganascia, 2002] Rolland, P.-Y., and Ganascia, J.-G. (2002) Pattern detection and discovery: The case of music data mining. In: Hand, D.J., Adams, N.M., and Bolton, R.J. (Eds.): *Pattern Detection and Discovery – Proceedings of the ESF Exploratory Workshop* (*Springer-Verlag*), London, UK, September 2002, pages 190 – 198. (LNAI 2447, ISBN 3-540-44148-4)

[Rymon, 1992] Rymon, R. (1992) Search through systematic set enumeration. In: Nebel, B., Rich, C., and Swartout, W.R. (Eds.): *Proceedings of the 3rd*

International Conference on Principles of Knowledge Representation and Reasoning (*KR-92, Morgan Kaufmann Publishers*), Cambridge, MA, USA, October 1992, pages 539 – 550. (ISBN 1-55860-262-3)

[Rypielski *et al.*, 2002] Rypielski, C., Wang, J.-C., and Yen, D.C. (2002) Data mining techniques for customer relationship management. *Technology in Society* 24 (2002): 483 – 502.

[Salton *et al.*, 1975] Salton, G., Wong, A., and Yang, C.S. (1975) A vector space model for automatic indexing. *Information Retrieval and Language Processing* 18(11): 613 – 620.

[Salton and Buckley, 1988] Salton, G., and Buckley, C. (1988) Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24(5): 513 – 523.

[Savasere *et al.*, 1995] Savasere, A., Omiecinski, E., and Navathe, S. (1995) An efficient algorithm for mining association rules in large databases. In: *Proceedings of the 21st International Conference on Very large Data Bases* (*VLDB-95, Morgan Kaufmann Publishers*), Zurich, Switzerland, September 1995, pages 432 – 444. (ISBN 1-55860-379-4)

[Scaringella *et al.*, 2006] Scaringella, N., Zoia, G., and Mlynek, D. (2006) Automatic genre classification of music content: A survey. *IEEE Signal Processing Magazine* 23(2): 133 – 141. (ISSN 1053-5888)

[Scheffer and Wrobel, 2002] Scheffer, T., and Wrobel, S. (2002) Text classification beyond the bag-of-words representation. In: *Proceedings of the Workshop on Text Learning, held at the Nineteenth International Conference on Machine Learning* (*ICML-TextML-02*), Sydney, Australia, July 2002.

[Schütze *et al.*, 1995] Schütze, H., Hull, D.A., and Pedersen, J.O. (1995) A comparison of classifiers and document representations for the routing problem. In: Fox, E.A., Ingwersen, P., and Fidel, R. (Eds.): *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-95, ACM Press*), Seattle, WA, USA, July 1995, pages 229 – 237. (ISBN 0-89791-714-6)

[Scime, 2005] Scime, A. (2005) *Web mining: Applications and techniques*. Idea Group Inc. (ISBN 1-59140-414-2)

[Scott and Matwin, 1999] Scott, S., and Matwin, S. (1999) Feature engineering for text classification. In: Bratko, I., and Dzeroski, S. (Eds.): *Proceedings of the Sixteenth International Conference on Machine Learning* (*ICML-99, Morgan Kaufmann Publishers*), Bled, Slovenia, June 1999, pages 379 – 388. (ISBN 1-55860-612-2)

[Sebastiani, 2002] Sebastiani, F. (2002) Machine learning in automated text categorization. *ACM Computing Surveys* 34(1): 1 – 47. (ISSN 0360-0300)

[Sebastiani, 2005] Sebastiani, F. (2005) Text categorization. In: Zanasi, A. (Editor): *Text Mining and Its Applications to Intelligence, CRM and Knowledge Management* (*Advances in Management Information*), WIT Press, Southampton, UK, pages 109 – 129. (ISBN 185312995X)

[Senellart and Blondel, 2004] Senellart, P.P., and Blondel, V.D. (2004) Automatic discovery of similar words. In: Berry, M.W. (Editor): *Survey of Text Mining – Clustering, Classification, and Retrieval*, Springer-Verlag New York, Inc., pages 25 – 43. (ISBN 0-387-95563-1)

[Shidara *et al.*, 2007] Shidara, Y., Nakamura, A., and Kudo, M. (2007) CCIC: Consistent common itemsets classifier. In: Perner, P. (Editor): *Proceedings of the 5th International Conference on Machine Learning and Data Mining* (*MLDM-07, Springer-Verlag*), Leipzig, Germany, July 2007, pages 490 – 498. (LNAI 4571, ISBN 978-3-540-73498-7)

[Silberschatz and Tuzhillin, 1995] Silberschatz, A., and Tuzhillin, A. (1995) On subjective measures of interestingness in knowledge discovery. In: Fayyad, U.M., and Uthurusamy, R. (Eds.): *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (*KDD-95, AAAI Press*), Montreal, Canada, August 1995, pages 275-281. (ISBN 0-929280-82-2)

[Smith, 2005] Smith, G.D. (2005) Meta-heuristics in the KDD process. In: Coenen, F. (Editor): *Proceedings of the First UK Knowledge Discovery in Data Symposium* (*UKKDD-05*), Liverpool, UK, April 2005.

[Soucy and Mineau, 2005] Soucy P., and Mineau, G.W. (2005) Beyond TFIDF weighting for text categorization in the vector space model. In: Kaelbling, L.P., and Saffiotti, A. (Eds.): *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence* (*IJCAI-05, Professional Book Center*), Edinburgh, Scotland, UK, July-August 2005, pages 1130 – 1135. (ISBN 0938075934)

[Spärck Jones, 1972] Spärck Jones, K. (1972) Exhaustivity and specificity. *Journal of Documentation* 28: 11 – 21. (Reprinted in 2004, 60: 493–502)

[Srikant and Agrawal, 1996] Srikant, R., and Agrawal, R. (1996) Mining quantitative association rules in large relational tables. In: Jagadish, H.V., and Mumick, I.S. (Eds.): *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (*SIGMOD-96, ACM Press*), Montreal, Quebec, Canada, June 1996, pages 1 – 12. (SIGMOD Record 25(2))

[Steinbach *et al.*, 2000] Steinbach, M., Karypis, G., and Kumar, V. (2000) A comparison of document clustering techniques. In: *Proceedings of the 2000 ACM SIGKDD Workshop on Text Mining*, Boston, MA, USA, August 2000.

[Sumathi and Sivanandam, 2006] Sumathi, S., and Sivanandam, S.N. (2006) *Introduction to data mining and its applications*. Springer-Verlag Berlin Heidelberg. (ISBN 3-540-34350-4)

[Sun and Lim, 2001] Sun, A., and Lim, E.-P. (2001) Hierarchical text classification and evaluation. In: Cercone, N., Lin, T.Y., and Wu, X. (Eds.): *Proceedings of the 2001 IEEE International Conference on Data Mining* (*ICDM-01, IEEE Computer Society*), San Jose, CA, USA, November-December 2001, pages 521 – 528. (ISBN 0-7695-1119-8)

[Tan, 1999] Tan, A.-H. (1999) Text mining: The state of the art and the challenges. In: *Proceedings of the PAKDD Workshop on Knowledge Discovery from Advanced Databases* (*KDAD-99*), Beijing, China, April 1999, pages 71 – 76.

[Thabtah *et al.*, 2005] Thabtah, F., Cowling, P., and Peng, Y. (2005) The impact of rule ranking on the quality of associative classifiers. In: Bramer, M., Coenen, F., and Allen, T. (Eds.): *Research and Development in Intelligent Systems XXII – Proceedings of AI-2005, the Twenty-fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence* (*AI-05, Springer-Verlag*), Cambridge, UK, December 2005, pages 277 – 287. (ISBN 1-84628-225-X)

[Thuraisingham, 1999] Thuraisingham, B. (1999) *Data mining: Technologies, techniques, tools, and trends*. CRC Press LLC, USA. (ISBN 0-8493-1815-7)

[Toivonen, 1996] Toivonen, H. (1996) Sampling large databases for association rules. In: Vijayaraman, T.M., Buchmann, A.P., Mohan, C., and Sarda, N.L. (Eds.): *Proceedings of the 22nd International Conference on Very Large Data Bases* (*VLDB-96, Morgan Kaufmann Publishers*), Mumbai (Bombay), India, September 1996, pages 134 – 145. (ISBN 1-55860-382-4)

[Uejima *et al.*, 2003] Uejima, H., Miura, T., and Shioya, I. (2003) Improving text categorization by resolving semantic ambiguity. In: *Proceedings of the 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* (*PACRIM-03, IEEE Computer Society*), August 2003, Volume 2, pages 796 – 799. (ISBN 0-7803-7978-0)

[Ullman, 1997] Ullman, J.D. (1997) Information integration using logical views. In: Afrati, F.N., and Kolaitis, P.G. (Eds.): *Database Theory — Proceedings of the 6th International Conference on Database Theory* (*ICDT-97, Springer-Verlag*), Delphi, Greece, January 1997, pages 19 – 40. (LNCS 1186, ISBN 3-540-62222-5)

[Van Rijsbergen, 1979] Van Rijsbergen, C.J. (1979) *Information retrieval* (*Second Edition*), Butterworth-Heinemann, Newton, MA, USA. (ISBN 0408709294)

[Walls *et al.*, 1999] Walls, F., Jin, H., Sista, S., and Schwartz, R. (1999) Topic detection in broadcast news. In: *Proceedings of the DARPA Broadcast News Workshop*, Herndon, VA, USA, February-March 1999, pages 193 – 198.

[Wang *et al.*, 2003a] Wang, J., Han, J., and Pei, J. (2003) CLOSET+: Searching for the best strategies for mining frequent closed itemsets. In: Getoor, L., Senator, T.E., Domingos, P., and Faloutsos, C. (Eds.): *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD-03,* *ACM Press*), Washington, DC, USA, August 2003, pages 236 – 245. (ISBN 1-58113-737-0)

[Wang *et al.*, 2003b] Wang, L., Bayan, M., Khan, L., and Rao, V.B. (2003) A new hierarchical approach for image clustering. In: *Proceedings of the 4th International Workshop on Multimedia Data Mining – a Workshop in Conjunction of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*MDM/KDD-03, ACM Press*), Washington, DC, USA, August 2003.

[Wang and Wang, 2005] Wang, Y., and Wang, X.-J. (2005) A new approach to feature selection in text classification. In: Yeung, D.S., Liu, Z.-Q., Wang, X.-Z., and Yan, H. (Eds.): *Advances in Machine Learning and Cybernetics – Proceedings of the 4th International Conference on Machine Learning and Cybernetics, Revised Selected Papers* (*ICMLC-05, Springer-Verlag*), Guangzhou, China, August 2005, pages 3814 – 3819. (LNAI 3930, ISBN 3540335846)

[Wang *et al.*, 2005] Wang, J.T.L., Zaki, M.J., Toivonen, H.T.T., and Shasha, DE. (2005) *Data mining in bioinformatics*. Springer-Verlag London Limited. (ISBN 1-85233-671-4)

[Wang *et al.*, 2006] Wang, Y.J., Coenen, F., Leng, P., and Sanderson, R. (2006) Text classification using language-independent pre-processing. In: Bramer, M., Coenen, F., and Tuson, A. (Eds.): *Research and Development in Intelligent Systems XXIII – Proceedings of the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence* (*AI-06, Springer-Verlag*), Peterhouse College, Cambridge, UK, December 2006, pages 413 – 417. (ISBN 1-84628-662-X)

[Wang *et al.*, 2007a] Wang, Y.J., Xin, Q., and Coenen, F. Mining efficiently significant classification association rules. In: Lin, T.Y., Wasilewska, A., Petry, F., and Xie, Y. (Eds.): *Data Mining: Foundations and Practice*, Springer-Verlag, *Accepted for publication, to appear.*

[Wang *et al.*, 2007b] Wang, Y.J., Xin, Q., and Coenen, F. (2007) A novel rule ordering approach in classification association rule mining. In: Perner, P. (Editor): *Proceedings of the 5th International Conference on Machine Learning and Data Mining* (*MLDM-07, Springer-Verlag*), Leipzig, Germany, July 2007, pages 339 – 348. (LNAI 4571, ISBN 978-3-540-73498-7)

[Washio *et al.*, 2005] Washio, T., Kok, J.N., and Raedt, L.C. (2005) *Advances in mining graphs, trees and sequences*. IOS Press. (ISBN 1586035282)

[Weiss and Indurkhya, 1997] Weiss, S.M., and Indurkha, N. (1997) *Predictive data mining: A practical guide*. Morgan Kaufmann Publishers, San Francisco, CA, USA. (ISBN 1-55860-403-0)

[Weiss *et al.*, 2004] Weiss, S.M., Indurkhya, N., Zhang, T., and Damerau, F.J. (2004) *Text mining: Predictive methods for analyzing unstructured information*. Springer Science+Business Media, Inc. USA. (ISBN 0-387-95433-3)

[Wiener *et al.*, 1995] Wiener, E., Pedersen, J.O., and Weigend, A.S. (1995) A neural network approach to topic spotting. In: *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval* (*SDAIR-95*), Las Vegas, NV, USA, April 1995, pages 317 – 332.

[Witten and Frank, 2005] Witten, I.H., and Frank, E. (2005) *Data mining: Practical machine learning tools and techniques* (*Second Edition*). Morgan Kaufmann Publishers, San Francisco, CA, USA. (ISBN 0-12-088407-0)

[Wu *et al.*, 2002] Wu, H., Phang, T.H., Liu, B., and Li, X. (2002) A refinement approach to handling model misfit in text categorization. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD-02, ACM Press*), Edmonton, Alberta, Canada, July 2002, pages 207 – 215. (ISBN 1-58113-567-X)

[Wu *et al.*, 2007] Wu, K., Lu, B.-L., Uchiyama, M., and Isahara, H. (2007) A probabilistic approach to feature selection for multi-class text categorization. In: Liu, D., Fei, S., Hou, Z.-G., Zhang, H., and Sun, C. (Eds.): *Advances in Neural Networks – Proceedings of the 4th International Symposium on Neural Networks* (*ISNN-07, Springer-Verlag*), Nanjing, China, June 2007, pages 1310 – 1317. (LNCS 4491, ISBN 978-3-540-72382-0)

[Yang and Pedersen, 1997] Yang, Y., and Pedersen, J.O. (1997) A comparative study on feature selection in text categorization. In: Fisher, D.H. (Editor): *Proceedings of the Fourteenth International Conference on Machine Learning* (*ICML-97, Morgan Kaufmann Publishers*), Nashville, TN, USA, July 1997, pages 412 – 420. (ISBN 1-55860-486-3)

[Yang and Liu, 1999] Yang, Y., and Liu, X. (1999) A re-examination of text categorization methods. In: Hearst, M.A., Gey, F.C., Tong, R.M. (Eds.): *Proceedings of the 22nd ACM SIGIR International Conference on Research and Development in Information Retrieval* (*SIGIR-99, ACM Press*), Berkley, CA, USA, August 1999, pages 42 – 49. (ISBN 1581130961)

[Yang *et al.*, 2001] Yang, L., Widyantoro, D.H., Ioerger, T., and Yen, J. (2001) An entropy-based adaptive genetic algorithm for learning classification rules. In: *Proceedings of the 2001 Congress on Evolutionary Computation* (*CEC-01, IEEE Computer Society*), Seoul, South Korea, May 2001, pages 790 – 796 (Volume 2). (ISBN 0-7803-6657-3)

[Yin and Han, 2003] Yin, X., and Han, J. (2003) CPAR: Classification based on predictive association rules. In: Barbará, D., and Kamath, C. (Eds.): *Proceedings of the Third SIAM International Conference on Data Mining* (*SDM-03, SIAM*), San Francisco, CA, USA, May 2003, pages 331 – 335. (ISBN 0-89871-545-8)

[Yoon and Lee, 2005] Yoon, Y., and Lee, G.G. (2005) Practical application of associative classifier for document classification. In: Lee, G.G., Yamada, A., Meng, H., and Myaeng, S.-H. (Eds.): *Information Retrieval Technology – Proceedings of the Second Asia Information Retrieval Symposium* (*AIRS-05, Springer-Verlag*), Jeju Island, Korea, October 2005, pages 467 – 478. (LNCS 3689, ISBN 3-540-29186-5)

[Zaïane and Antonie, 2002] Zaïane, O.R., and Antonie, M.-L. (2002) Classifying text documents by associating terms with text categories. In: Zhou X. (Editor): *Database Technologies 2002 – Proceedings of the 13th Australasian Database Conference* (*ADC-02, CRPIT 5 Australian Computer Society*), Melbourne, Victoria, Australia, January-February 2002, pages 215 – 222. (ISBN 0-909-92583-6)

[Zaki *et al.*, 1997] Zaki, M.J., Parthasarathy, S., Ogihara, M., and Li, W. (1997) New algorithms for fast discovery of association rules. In: Heckerman, D., Mannila, H., Pregibon, D., and Uthurusamy, R. (Eds.): *Proceedings of the third International Conference on Knowledge Discovery and Data Mining* (*KDD-97, AAAI Press*), Beach, CA, USA, August 1997, pages 283 – 286. (ISBN 1-57735-027-8)

[Zaki and Hsiao, 2002] Zaki, M.J., and Hsiao, C.-J. (2002) CHARM: An efficient algorithm for closed itemset mining. In: Grossman, R.L., Han, J., Kumar, V., Mannila, H., and Motwani, R. (Eds.): *Proceedings of the Second SIAM International Conference on Data Mining* (*SDM-02, SIAM*), Arlington, VA, USA, April 2002, Part IX No.1. (ISBN 0-89871-517-2)

[Zavrel and Daelemans, 1999] Zavrel, J. and Daelemans, W. (1999) Recent advances in memory-based part-of-speech tagging. In: *VI Simposio Internacional de Comunicacion Social*, Santiago de Cuba, 1999, pages 590 – 597.

[Zhang and Zhou, 2004] Zhang, D., and Zhou, L. (2004) Discovering golden nuggets: Data mining in financial application. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews* 34(4): 513 – 552.

[Zhang *et al.*, 2005a] Zhang, Y., Zincir-Heywood, N., and Milios, E. (2005) Narrative text classification for automatic key phrase extraction in web document corpora. In: Bonifati, A., and Lee, D. (Eds.): *Proceedings of the Seventh ACM International Workshop on Web Information and Data Management* (*WIDM-05, ACM Press*), Bremen, Germany, November 2005, pages 51 – 58. (ISBN 1-59593-194-5)

[Zhang *et al.*, 2005b] Zhang, Y., Gong, L., and Wang, Y. (2005) An improved TF-IDF approach for text classification. *Journal of Zhejing University SCIENCE* 6A(1): 49 – 55. (ISSN 1009-3095)

[Zheng and Srihari, 2003] Zheng, Z., and Srihari, R. (2003) Optimally combining positive and negative features for text categorization. In: *Proceedings of the 2003 ICML Workshop on Learning from Imbalanced Data Sets II*, Washington, DC, USA, August 2003.

[Zhuang *et al.*, 2005] Zhuang, D., Zhang, B., Yang, Q., Yan, J., Chen, Z., and Chen, Y. (2005) Efficient text classification by weighted proximal SVM. In: Han, J., Wah, B.W., Raghavan, V., Wu, X., and Rastogi, R. (Eds.): *Proceedings of the 5th IEEE International Conference on Data Mining* (*ICDM-05, IEEE Computer Society*), Huston, TX, USA, November 2005, pages 538 – 545. (ISBN 0-7695-2278-5)

Appendix A

Keywords using All Potential Significant Words

With respect to the strategy of Keywords ("bag of words" based on LTGFR or LTGSR) it is possible to use all the identified potential significant words as significant words (i.e. K = 4,000, more than the expected number of potential significant words). An individual experiment was run based on the parameter settings given in Table A.1.

Support Threshold (σ)	0.1%
Confidence Threshold (α)	35%
Significance Threshold (G)	3
Upper Noise Threshold (UNT)	7%
Lower Noise Threshold (LNT)	0.2%
Max # Significant Words (K)	**4,000**

Table A.1: Parameter settings for the experiment with K = 4,000 (Keywords)

	LTGFR		LTGSR	
	Unique	All	Unique	All
Accuracy	73.2	73.8	73.1	74.2
Accuracy (K = 1,500)	75.1	75.8	74.4	75.6
# Keywords	2,921	3,609	3,198	3,697
Time	184	106	102	106
# Empty Documents	48	29	55	41

Table A.2: Experimental result obtained with K = 4,000 (Keywords)

Table A.2 shows the experimental result, using the strategy of Keywords, with K = 4,000. Remember that if using all available potential significant words there is no option to distribute ("dist"). The result shows that the accuracy is not as good as in the case where only the 1,500 most significant words are used (accuracy = 75.8%). The reason for this is that with K = 4,000 many words are included that are not good indicators of class.

Appendix B

Reducing the Number of Attributes for DelSOcontGN/GW

With respect to the DelSOcontGN strategy the number of identified attributes (phrases) can be reduced by increasing the *G* value from 3 to 8. Recall that DelSOcontGN did not result in a classifier because the 2^{15} attribute limit was reached. The intuition here was that this would result in fewer keywords. An individual experiment was run based on the parameter settings presented in Table B.1.

Support Threshold (σ)	0.1%
Confidence Threshold (α)	35%
Significance Threshold (*G*)	**8**
Upper Noise Threshold (UNT)	7%
Lower Noise Threshold (LNT)	0.2%
Max # Significant Words (*K*)	1,500

Table B.1: Parameter settings for the experiment with *G* = 8 (DelSOcontGN)

	LTGFR		LTGSR	
	Unique	All	Unique	All
Accuracy	49.6	49.6	59.8	59.8
# Ordinary Words	5,036	5,036	5,163	5,163
# Keywords	671	671	544	544
# Attributes	28,924	28,924	21,570	21,570
Time	269	257	203	204
# Empty Documents	2,051	2,051	2,479	2,479
# Rules	472	472	460	460
Levels in T-tree	3	3	2	2

Table B.2: Experimental result obtained with *G* = 8 (DelSOcontGN)

Table B.2 shows the experimental result, using DelSOcontGN, with *G* = 8. Note that the number of attributes now fits into the 2^{15} limit, but very few (less than 1,500) keywords were identified. The classification accuracy, for each particular case, is poor. Therefore it was decided to abandon the DelSOcontGN strategy.

With respect to the DelSOcontGW strategy, experiments were undertaken to reduce the number of attributes by decreasing the *K* value, in steps of 250, from 1,500 to 750. A set of experiments was run based on the parameter settings given in Table B.3.

Support Threshold (σ)	0.1%
Confidence Threshold (α)	35%
Significance Threshold (*G*)	3
Upper Noise Threshold (UNT)	7%
Lower Noise Threshold (LNT)	0.2%
Max # Significant Words (*K*)	**750 ~ 1,500**

Table B.3: Parameter settings for the set of experiments with *K* = 750 ~ 1,500 (DelSOcontGW)

	LTGFR				LTGSR			
	Unique		All		Unique		All	
	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*
K = 1,500	32,913	34,079	32,549	35,090	32,000	32,360	31,542	31,629
K = 1,250	26,305	28,170	26,285	28,058	25,883	26,615	25,608	26,618
K = 1,000	20,503	22,035	20,965	22,089	20,642	20,475	19,845	20,335
K = 750	14,749	15,763	14,861	16,051	14,163	14,575	14,031	14,575

Table B.4: Number of attributes generated with *K* = 750 ~ 1,500 (DelSOcontGW)

	LTGFR				LTGSR			
	Unique		All		Unique		All	
	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*
K = 1,500			**70.9**		70.4	66.0	**71.2**	68.9
K = 1,250	69.6	69.0	**70.9**	59.1	70.0	66.5	**71.2**	67.9
K = 1,000	69.6	57.7	**71.7**	56.7	69.5	65.6	**70.8**	66.3
K = 750	68.5	55.2	**69.5**	51.8	67.6	62.0	**70.2**	62.0

Table B.5: Classification accuracy obtained with *K* = 750 ~ 1,500 (DelSOcontGW)

Tables B.4 and B.5 show the number of attributes generated and the classification accuracy obtained, using DelSOcontGW, with *K* = 750 ~ 1,500. The results indicate that there is no noticeable improvement in classification accuracy when decreasing the number of attributes. Therefore the DelSOcontGW strategy was also abandoned.

Appendix C

Increasing the Number of Attributes for DelSNcontGO/GW

In both the DelSNcontGO and the DelSNcontGW strategies stop marks and noise words are used as delimiters, while phrases are made up of at least one significant word and ordinary or wild card characters (representing ordinary words). As a result many fewer attributes are produced than with the other two (DelSO) phrase identification strategies. Note that DelSNcontGW produced less attributes than DelSNcontGO because phrases that are distinct in DelSNcontGO are collapsed in DelSNcontGW.

Intuitively the more attributes (phrases) that are identified the better the documentbase representation and the higher the classification accuracy (provided that good attributes are identified). The number of attributes for both DelSNcontGO and DelSNcontGW can be increased by increasing the value of *K* as in the experiment of Appendix A.

With respect to the DelSNcontGO strategy the number of attributes was increased by increasing the *K* value, in steps of 250, from 1,500 to 2,000. In this context, it should be noted that the 2^{15} limit will be reached when setting the value of *K* at 2,250 or higher. A set of experiments was run based on the parameter settings given in Table C.1.

Support Threshold (σ)	0.1%
Confidence Threshold (α)	35%
Significance Threshold (G)	3
Upper Noise Threshold (UNT)	7%
Lower Noise Threshold (LNT)	0.2%
Max # Significant Words (*K*)	**1,500 ~ 2,000**

Table C.1: Parameter settings for the set of experiments with *K* = 1,500 ~ 2,000 (DelSNcontGO)

	LTGFR				LTGSR			
	Unique		All		Unique		All	
	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*
K = 1,500	1,399	1,423	1,518	1,347	1,448	1,553	1,548	1,597
K = 1,750	1,592	1,618	1,723	1,595	1,631	1,748	1,743	1,825
K = 2,000								4,070

Table C.2: Number of rules generated with *K* = 1,500 ~ 2,000 (DelSNcontGO)

	LTGFR				LTGSR			
	Unique		All		Unique		All	
	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*
K = 1,500	75.3	73.2	**76.7**	71.9	75.9	73.0	**77.0**	74.3
K = 1,750	74.7	74.0	**76.9**	75.6	76.2	74.3	**76.7**	75.1
K = 2,000								75.3

Table C.3: Classification accuracy obtained with *K* = 1,500 ~ 2,000 (DelSNcontGO)

Tables C.2 and C.3 show the number of rules generated and the classification accuracy obtained, using DelSNcontGO, with *K* = 1,500 ~ 2,000. By increasing the number of attributes many more rules are generated. However this does not lead to better classification accuracy because less good additional significant words are included.

With respect to the DelSNcontGW strategy the number of possible attributes was increased by increasing the *K* value, in steps of 250, from 1,500 to 3,000. A set of experiments was run based on the parameter settings presented in Table C.4.

Support Threshold (σ)	0.1%
Confidence Threshold (α)	35%
Significance Threshold (*G*)	3
Upper Noise Threshold (UNT)	7%
Lower Noise Threshold (LNT)	0.2%
Max # Significant Words (*K*)	**1,500 ~ 3,000**

Table C.4: Parameter settings for the set of experiments with *K* = 1,500 ~ 3,000 (DelSNcontGW)

	LTGFR				LTGSR			
	Unique		All		Unique		All	
	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*
K = 1,500	1,774	1,769	1,912	1,648	1,839	1,935	1,945	1,989
K = 1,750	1,971	1,997	2,135	1,921	2,026	2,149	2,154	2,240
K = 2,000	2,169	2,186	2,288	2,090	2,238	2,328	2,353	2,480
K = 2,250	2,385	2,428	2,488	2,356	2,418	2,503	2,557	2,596
K = 2,500	2,662	2,588	2,638	2,538	2,617	2,631	2,735	2,840
K = 2,750	2,862	2,837	2,842	2,819	2,783	2,798	2,912	3,008
K = 3,000	6,022	6,022	6,058	5,998	5,290	5,948	6,182	6,346

Table C.5: Number of rules generated with *K* = 1,500 ~ 3,000 (DelSNcontGW)

	LTGFR				LTGSR			
	Unique		All		Unique		All	
	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*	Dist	Top *K*
K = 1,500	75.1	71.6	**76.2**	68.5	74.9	71.3	**75.8**	72.3
K = 1,750	72.6	71.9	**75.0**	72.9	73.2	71.6	**74.1**	72.2
K = 2,000	72.6	71.3	**73.9**	73.5	72.8	72.5	**73.6**	71.4
K = 2,250	71.8	72.7	**72.8**	73.5	71.8	71.9	**72.9**	72.3
K = 2,500	72.4	71.6	**73.2**	73.0	72.0	72.0	**74.0**	72.5
K = 2,750	71.7	71.9	**73.5**	74.7	72.4	71.1	**74.5**	73.4
K = 3,000	71.7	71.7	**73.4**	74.0	71.6	71.1	**73.9**	73.2

Table C.6: Classification accuracy obtained with *K* = 1,500 ~ 3,000 (DelSNcontGW)

Tables C.5 and C.6 show the number of rules generated and the classification accuracy obtained, using DelSNcontGW, with *K* = 1,500 ~ 3,000. Again, by increasing the number of attributes many more rules are generated. However this does not lead to better classification accuracy.

Appendix D

Change in Classification Accuracy with Change in *G*

Table D.1 shows the effect on classification accuracy with changes in the value of *G*, when LNT = 0.2%, UNT = 7%, and *K* = 1,500. Figures D.1 and D.2 represent Table D.1 in graph form, and show that there is little effect until the value of *G* reaches a value of approximately 6. The drop is slightly less severe using the LTGFR strategy when compared with the LTGSR strategy.

Significant Word Selection Strategy	**Documentbase Pre-processing Strategy**	***G* Value**									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
LTGFR All Words Dist	DelSNcontGO	76.7	76.7	76.7	76.2	75.0	65.2	60.9	51.6	43.9	35.0
	DelSNcontGW	75.9	75.9	75.9	74.6	72.5	63.4	60.4	51.4	43.5	35.4
	Keywords	75.5	75.5	75.5	75.9	74.8	65.2	64.5	60.7	51.6	43.9
LTGSR All Words Dist	DelSNcontGO	77.3	77.3	77.0	76.8	74.5	71.7	67.1	60.4	47.4	0
	DelSNcontGW	75.9	75.9	75.5	73.7	72.3	70.8	66.9	61.0	48.5	0
	Keywords	75.7	75.7	75.3	74.9	73.4	71.3	66.6	60.6	48.6	0

Table D.1: Relationship between *G* and classification accuracy obtained for NGA.D10000.C10 with LNT = 0.2%, UNT = 7%, and *K* = 1,500

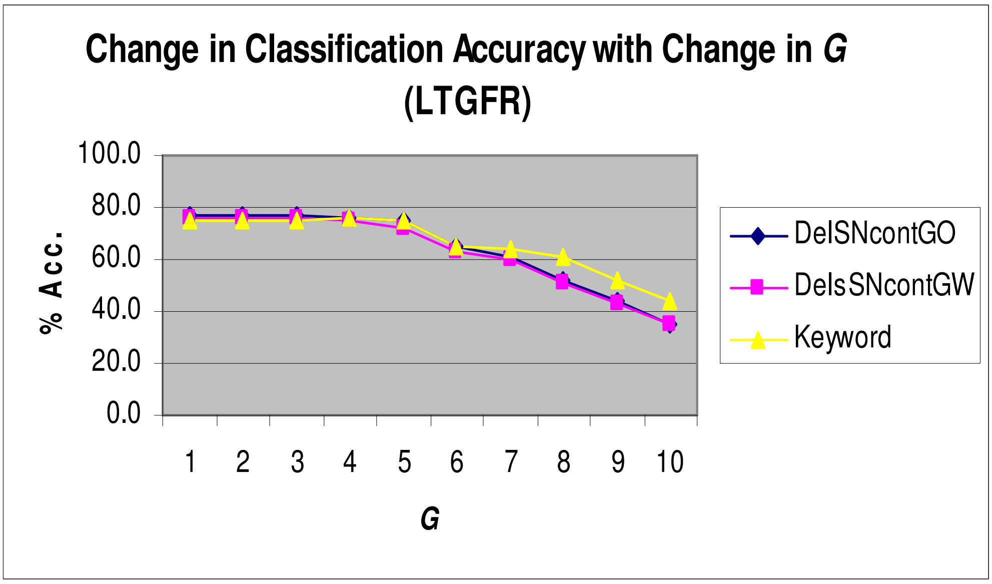


Figure D.1: Relationship between *G* and classification accuracy obtained for NGA.D10000.C10 with LNT = 0.2%, UNT = 7%, and *K* = 1,500 (LTGFR contribution calculation)

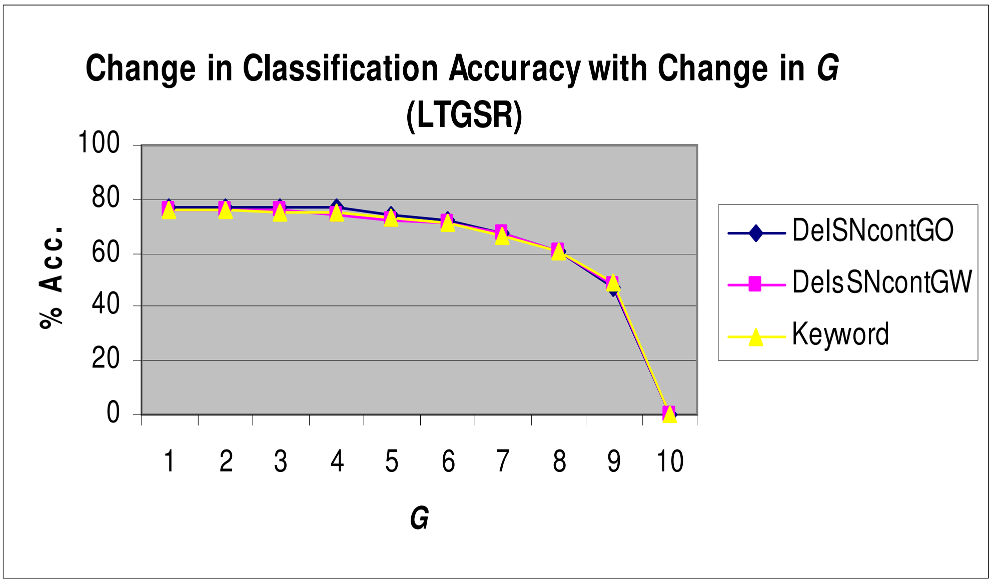


Figure D.2: Relationship between *G* and classification accuracy obtained for NGA.D10000.C10 with LNT = 0.2%, UNT = 7%, and *K* = 1,500 (LTGSR contribution calculation)