Division of Informatics, University of Edinburgh

Centre for Intelligent Systems and their Applications

Rough Set-Aided Keyword Reduction for Text Categorisation

by

Alexios Chouchoulas, Qiang Shen

Informatics Research Report EDI-INF-RR-0121

Division of Informatics **May 2001**
http://www.informatics.ed.ac.uk/

Rough Set-Aided Keyword Reduction for Text Categorisation

Alexios Chouchoulas, Qiang Shen

Informatics Research Report EDI-INF-RR-0121

DIVISION *of* INFORMATICS
Centre for Intelligent Systems and their Applications

May 2001

Presented in: Applied Artificial Intelligence, 15(9):843-873, 2001

Abstract :

The volume of electronically stored information increases exponentially as the state of the art progresses. Automated Information Filtering (IF) and Information Retrieval (IR) systems are therefore acquiring rapidly increasing prominence. However, such systems sacrifice efficiency to boost effectiveness. Such systems typically have to cope with sets of vectors of many tens of thousands of dimensions. Rough Set (RS) theory can be applied to reducing the dimensionality of data used in IF/IR tasks, by providing a measure of the information content of datasets with respect to a given classification. This can aid IF/IR systems that rely on the acquisition of large numbers of term weights or other measures of relevance.

This paper investigates the applicability of RS theory to the IF/IR application domain and compares this applicability with respect to various existing TC techniques. The ability of the approach to generalise given a minimum of training data is also addressed. The background of RS theory is presented, with an illustrative example to demonstrate the operation of the RS-based dimensionality reduction. A modular system is proposed that allows the integration of this technique with a large variety of different IF/IR approaches. The example application, categorisation of E-mail messages, is described. Systematic experiments and their results are reported and analysed.

Keywords : Text Categorisation, Rough Sets, Dimensionality Reduction, E-mail Classification

Rough Set-Aided Keyword Reduction for Text Categorisation

Alexios Chouchoulas and Qiang Shen
{alexios,qiangs}@dai.ed.ac.uk

School of Artificial Intelligence, Division of Informatics,
University of Edinburgh, 80 South Bridge, Edinburgh EH1 1QN, UK.

Abstract: The volume of electronically stored information increases exponentially as the state of the art progresses. Automated Information Filtering (IF) and Information Retrieval (IR) systems are therefore acquiring rapidly increasing prominence. However, such systems sacrifice efficiency to boost effectiveness. Such systems typically have to cope with sets of vectors of many tens of thousands of dimensions. Rough Set (RS) theory can be applied to reducing the dimensionality of data used in IF/IR tasks, by providing a measure of the information content of datasets with respect to a given classification. This can aid IF/IR systems that rely on the acquisition of large numbers of term weights or other measures of relevance.

This paper investigates the applicability of RS theory to the IF/IR application domain and compares this applicability with respect to various existing TC techniques. The ability of the approach to generalise given a minimum of training data is also addressed. The background of RS theory is presented, with an illustrative example to demonstrate the operation of the RS-based dimensionality reduction. A modular system is proposed that allows the integration of this technique with a large variety of different IF/IR approaches. The example application, categorisation of E-mail messages, is described. Systematic experiments and their results are reported and analysed.

1 Introduction

It is a well known fact that the volume of electronically stored information increases exponentially with time. Sorting through even a fraction of available information information can be very difficult for any human being. Information Filtering (IF) and Information Retrieval (IR) systems are therefore acquiring rapidly increasing prominence as automated aids in quickly sifting through information.

Many techniques involving text categorisation (TC) describe documents by using vectors of real numbers that exhibit extremely high dimensionality — typically one value per word or pair of words in a document or corpus of documents (Rijsbergen, 1979). These vector ordinates are used as preconditions to rules or similarity metrics which decide what category the document belongs to. There are several tens of thousands of such ordinates in all but the simplest applications. This makes effective TC an extremely hard, if not intractable, problem for even the most powerful computers, unless the task is simplified. The use of the cosine of the angle between two vectors as a common comparison metric (Rijsbergen, 1979) further increases the number of operations needed to categorise a given document. Better design of TC systems is therefore desirable. This work, based on the earlier work reported by the authors in (Chouchoulas and Shen, 1999c), introduces a novel

technique, using Rough Sets (Pawlak, 1982), to significantly reduce the dimensionality of TC datasets without reducing their information content.

Rough Set theory is a formal mathematical tool that can be applied to reducing the dimensionality of datasets (Pawlak, 1991; Shen and Chouchoulas, 1999b) by providing a measure of the information content of datasets with respect to a certain classification. Rough Sets have already been applied to a very wide variety of application domains with success (Shen and Chouchoulas, 1999a; Martienne and Quafafou, 1998; Keen and Rajasekar, 1994; Das-Gupta, 1988). Different facets of the theory can aid in identifying information-rich portions of datasets, as well as the equivalence relations between attributes in datasets. Applying RS to TC provides assistance in locating those parts of TC datasets that possess the necessary information for the application domain, thereby reducing the amount of data to be handled by text classifiers. In effect, the technique locates minimal sets of co-ordinate keywords that best characterise the documents.

The paper begins by introducing Rough Set theory from the perspective of dimensionality reduction. To clarify the workings of the dimensionality reduction technique employed herein, an example is followed through. A text categorisation system is then proposed via the use of Rough Set-based keyword reduction A detailed description of the system modules is given and the application domain used for experimental verification of the system is introduced. Comprehensive experimental results are provided and analysed. The paper is finally concluded with a discussion of its contributions and future work. In particular, the following important issues are covered: the effectiveness of dimensionality reduction, the preservation of important content, the implications of dimensionality reduction on the existing TC techniques, the efficiency of the proposed system, and its ability to generalise when training data is not in abundance. The experimental results demonstrate the success of the technique.

2 An Overview of Rough Set-Assisted Attribute Reduction

2.1 Basic Concepts

Suppose that a dataset $\mathbb{D}$ is viewed as a table or matrix where attributes are columns and objects are rows. Let $\mathbb{U}$ denote the set of all objects in the dataset, $\mathbb{A}$ the set of all attributes, $\mathbb{C}$ the set of conditional (or input) attributes and $\mathbb{D}$ the set of decision attributes. $\mathbb{C}$ and $\mathbb{D}$ are mutually exclusive and $\mathbb{C} \cup \mathbb{D} = \mathbb{A}$. Let $f(x, q)$ denote the value of attribute $q \in \mathbb{A}$ in object $x \in \mathbb{U}$. Thus, $f(x, q)$ defines an equivalence relation over $\mathbb{U}$. With respect to a given q, this value can be used to partition the universe of discourse into a set of disjoint subsets of $\mathbb{U}$:

$$R_q = \{x : x \in \mathbb{U} \ \wedge \ f(x, q) = f(x_0, q) \ \forall \ x_0 \in \mathbb{U}\} \tag{1}$$

For instance, given the example dataset of table 1 (drawn from the work of Pawlak (1991)), $\mathbb{U} = \{0, 1, 2, 3, 4, 5, 6, 7\}$, $\mathbb{A} = \{a, b, c, d, e\}$, $\mathbb{C} = \{a, b, c, d\}$, and $\mathbb{D} = \{e\}$, the following partitions may be obtained:

$$R_a \quad = \quad \{\{1, 7\}, \{0, 3, 4\}, \{2, 5, 6\}\}$$

Table 1: An example dataset.

$x \in \mathbb{U}$	a	b	c	d	$\Rightarrow$	e
0	1	0	2	2		0
1	0	1	1	1		2
2	2	0	0	1		1
3	1	1	0	2		2
4	1	0	2	0		1
5	2	2	0	1		1
6	2	1	1	1		2
7	0	1	1	0		1

$$
\begin{aligned}
R_b &= \{\{0,2,4\},\{1,3,6,7\},\{5\}\} \\
R_c &= \{\{2,3,5\},\{1,6,7\},\{0,4\}\} \\
R_d &= \{\{4,7\},\{1,2,5,6\},\{0,3\}\} \\
R_e &= \{\{0\},\{2,4,5,7\},\{1,3,6\}\}
\end{aligned}
$$

Given a subset of attributes, $\mathbb{P} \subseteq \mathbb{A}$, two objects x and y in $\mathbb{U}$ are deemed to be *indiscernible* with respect to $\mathbb{P}$ if and only if $f(x,q) = f(y,q) \ \ \forall \ \ q \in \mathbb{P}$. If $\text{IND}(P)$ denotes the indiscernibility relation for all $\mathbb{P} \subseteq \mathbb{A}$, then $\mathbb{U}/\text{IND}(\mathbb{P})$ is used to denote the partition of $\mathbb{U}$ given $\text{IND}(\mathbb{P})$, and is calculated as:

$$\mathbb{U}/\text{IND}(\mathbb{P}) = \bigotimes \{q \in \mathbb{P} : \mathbb{U}/\text{IND}(\{q\})\}, \text{ where} \tag{2}$$

$$A \otimes B = \{X \cap Y : \forall \ \ X \in A, \ \forall \ \ Y \in B, \ X \cap Y \neq \emptyset\} \tag{3}$$

For instance, if $\mathbb{P} = \{b, c\}$, objects 0 and 4 are indiscernible; 1, 6 and 7 likewise. The rest of the objects are not. This applies to the example dataset as follows:

$$
\begin{aligned}
\mathbb{U}/\text{IND}(\mathbb{P}) &= \mathbb{U}/\text{IND}(\{b\}) \otimes \mathbb{U}/\text{IND}(\{c\}) && (4) \\
&= \{\{0,2,4\},\{1,3,6,7\},\{5\}\} \otimes \{\{2,3,5\},\{1,6,7\},\{0,4\}\} \\
&= \{\{0,2,4\} \cap \{2,3,5\}, \ \{0,2,4\} \cap \{1,6,7\}, \ \cdots, \ \{5\} \cap \{0,4\}\} \\
&= \{\{2\},\{0,4\},\{3\},\{1,6,7\},\{5\}\} && (5)
\end{aligned}
$$

Rough Set theory involves the approximation of traditional sets using a pair of other sets, named the *lower* and *upper approximations*) of the set in question.
If $\mathbb{P} = \{a, b, c\}$, then, similarly:

$$\mathbb{U}/\text{IND}(\mathbb{P}) = \mathbb{U}/\text{IND}(\{a\}) \otimes \mathbb{U}/\text{IND}(\{b\}) \otimes \mathbb{U}/\text{IND}(\{c\})$$

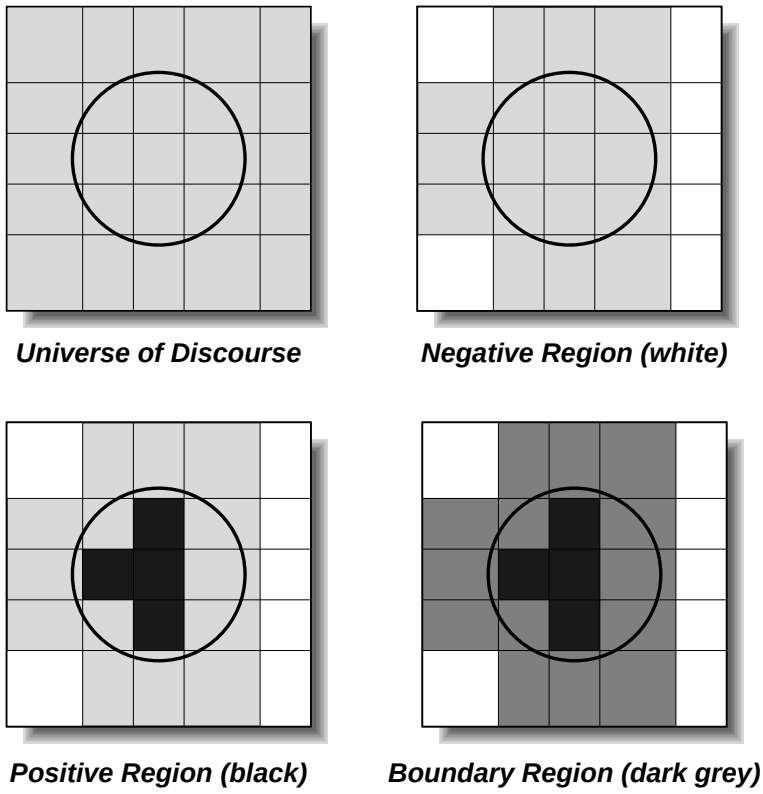

Figure 1: Approximating a circle with rough sets. From left to right, top to bottom: the universe of discourse (a circle on an irregular, two-dimensional grid); the positive region marked in white; the negative region in black; boundary region (the actual approximation) in dark grey.

Formally, given an equivalence relation $\mathrm{IND}(\mathbb{P})$, the lower approximation $\underline{\mathbb{P}}Y$ and the upper approximation $\overline{\mathbb{P}}Y$ are defined as:

$$\underline{\mathbb{P}}Y = \bigcup \{X : X \in \mathbb{U}/\mathrm{IND}(\mathbb{P}), X \subseteq Y\} \tag{6}$$

$$\overline{\mathbb{P}}Y = \bigcup \{X : X \in \mathbb{U}/\mathrm{IND}(\mathbb{P}), X \cap Y \neq \emptyset\} \tag{7}$$

Assuming that $\mathbb{P}$ and $\mathbb{Q}$ are equivalence relations in $\mathbb{U}$, the *positive*, *negative* and *boundary regions* of $\mathbb{Q}$ with respect to $\mathbb{P}$ are denoted by $\mathrm{POS}_P(\mathbb{Q})$, $\mathrm{NEG}_P(\mathbb{Q})$ and $\mathrm{BN}_P(\mathbb{Q})$ respectively and defined as:

$$\mathrm{POS}_P(\mathbb{Q}) = \bigcup_{X \in \mathbb{Q}} \underline{\mathbb{P}}X \tag{8}$$

$$\mathrm{NEG}_P(\mathbb{Q}) = \mathbb{U} - \bigcup_{X \in \mathbb{Q}} \overline{\mathbb{P}}X \tag{9}$$

$$\mathrm{BN}_P(\mathbb{Q}) = \bigcup_{X \in \mathbb{Q}} \overline{\mathbb{P}}X - \bigcup_{X \in \mathbb{Q}} \underline{\mathbb{P}}X \tag{10}$$

The positive region contains all objects in $\mathbb{U}$ that can be classified in attributes $\mathbb{Q}$ using the information in attributes $\mathbb{P}$. The negative region is the set of objects that cannot be classified this way, while the boundary region consists of objects that can possibly be classified in this context. A visual illustration of this is shown in figure 1. For example, assuming $\mathbb{P} = \{b, c\}$ and $\mathbb{Q} = \{e\}$:

$$\begin{aligned}
\mathrm{POS}_{\mathrm{IND}(P)}(\mathrm{IND}(\mathbb{Q})) &= \cup\{\emptyset, \{2,5\}, \{3\}\} = \{2,3,5\} \\
\mathrm{NEG}_{\mathrm{IND}(P)}(\mathrm{IND}(\mathbb{Q})) &= \mathbb{U} - \cup\{\{0,4\}, \{2,0,4,1,6,7,5\}, \{3,1,6,7\}\} = \emptyset \\
\mathrm{BN}_{\mathrm{IND}(P)}(\mathrm{IND}(\mathbb{Q})) &= \mathbb{U} - \{2,3,5\} = \{0,1,3,4,6,7\}
\end{aligned}$$

What this means is that, with respect to conditional attributes b and c, objects 2, 3 and 5 can definitely be classified in terms of decision attribute e. The remaining objects could, possibly, be classified, but it is not certain. This is shown more intuitively in table 2. Based on this, the *degree of dependency* $\gamma_{\mathbb{P}}(\mathbb{Q})$ of a set $\mathbb{Q}$ of decision attributes on a set of conditional attributes $\mathbb{P}$ is defined as:

Table 2: The shaded objects (2, 3 and 5) are discernible and can definitely be classified into e using the selected conditional attributes b and c. The rest of the objects cannot be classified — the information that would make them discernible in this context is missing.

$x \in \mathbb{U}$	b	c	$\Rightarrow$	e
0	0	2		0
1	1	1		2
2	0	0		1
3	1	0		2
4	0	2		1
5	2	0		1
6	1	1		2
7	1	1		1

$$\gamma_{\mathbb{P}}(\mathbb{Q}) = \frac{\parallel \mathrm{POS}_P(\mathbb{Q}) \parallel}{\parallel \mathbb{U} \parallel} \tag{11}$$

Where $\parallel Set \parallel$ is the cardinality of *Set*. The complement of γ gives a measure of the contradictions in the selected subset of the dataset. If $\gamma = 0$, there is no dependence; for $0 < \gamma < 1$, there is a partial dependence. If $\gamma = 1$, there is complete dependence. For instance, in the example:

$$\gamma_{\{b,c\}}(\{e\}) = \frac{\parallel \mathrm{POS}_P(\{e\}) \parallel}{\parallel \mathbb{U} \parallel} = \frac{\parallel \{2,3,5\} \parallel}{\parallel \{0,1,2,3,4,5,6,7\} \parallel} = \frac{3}{8} = 0.375$$

This shows that, of the eight objects, only three can be classified into the decision attribute e, given conditional attributes b and c. The other five objects (the unshaded rows in table 2) represent contradictory information.

From this, a further concept can be defined, the *significance* of an attribute. This is done by calculating the change of dependency when removing the attribute from the set of considered conditional attributes. Given two sets $\mathbb{P}$ and $\mathbb{Q}$ and an attribute $a \in \mathbb{P}$, the significance of a with respect to $\mathbb{Q}$ is defined by:

$$\sigma_{\mathbb{P}}(\mathbb{Q}, a) = \gamma_{\mathbb{P}}(\mathbb{Q}) - \gamma_{\mathbb{P}-\{a\}}(\mathbb{Q}) \tag{12}$$

The higher the change in dependency, the more significant a is regarded to be. For instance, let $\mathbb{P} = \{a, b, c\}$ and $\mathbb{Q} = \{e\}$, a few examples of degrees of dependency can be calculated as follows:

$$
\begin{aligned}
\gamma_{\{a,b,c\}}\big(\{e\}\big) &= \| \{2,3,5,6\} \| / 8 &= 4/8 \\
\gamma_{\{a,b\}}\big(\{e\}\big) &= \| \{2,3,5,6\} \| / 8 &= 4/8 \\
\gamma_{\{b,c\}}\big(\{e\}\big) &= \| \{2,3,5\} \| / 8 &= 3/8 \\
\gamma_{\{a,c\}}\big(\{e\}\big) &= \| \{2,3,5,6\} \| / 8 &= 4/8
\end{aligned}
$$

Using these calculations, it is possible to evaluate the significance of the three conditional attributes a, b and c:

$$
\begin{aligned}
\sigma_{\mathbb{P}}(\mathbb{Q}, a) &= \gamma_{\{a,b,c\}}\big(\{e\}\big) - \gamma_{\{b,c\}}\big(\{e\}\big) &= 1/8 \\
\sigma_{\mathbb{P}}(\mathbb{Q}, b) &= \gamma_{\{a,b,c\}}\big(\{e\}\big) - \gamma_{\{a,c\}}\big(\{e\}\big) &= 0 \\
\sigma_{\mathbb{P}}(\mathbb{Q}, c) &= \gamma_{\{a,b,c\}}\big(\{e\}\big) - \gamma_{\{a,b\}}\big(\{e\}\big) &= 0
\end{aligned}
$$

This shows that attribute a is not indispensable, having a significance of 0.125, while attributes b and c can be dispensed with, as they do not provide any information significant for the classification into e.

Attribute reduction involves removing redundant conditional attributes that have no significance to the classification at hand. An *attribute reduct set* (or simply *reduct*) is then defined as a subset R of the set of conditional attributes $\mathbb{C}$ such that $\gamma_{\mathbb{C}}(\mathbb{D}) = \gamma_R(\mathbb{D})$. For the set of decision attributes $\mathbb{D}$, it is obvious that a dataset may have more than one attribute reduct set. The set $\mathcal{R}$ of all attribute reduct sets R is defined as:

$$
\mathcal{R} = \Big\{ X : X \subseteq \mathbb{C}, \;\; \gamma_{\mathbb{C}}(\mathbb{D}) = \gamma_X(\mathbb{D}) \Big\} \tag{13}
$$

The RSDR will not compromise with a set of conditional attributes that has a large part of the information embedded in the initial conditional attribute set, $\mathbb{C}$ — it *always* attempts to reduce the attribute set while losing *no* information significant to the classification at hand. Rough Set Dimensionality Reduction (RSDR) works by searching for an attribute reduct set of least cardinality. That is, it aims to locate one arbitrary element of the set of *minimal reducts* $\mathcal{R}_{min} \subseteq \mathcal{R}$:

$$
\mathcal{R}_{min} = \big\{ X : X \in \mathcal{R}, \;\; \forall\, Y \in \mathcal{R}, \;\; \| X \| \leq \| Y \| \big\} \tag{14}
$$

Returning to the example and calculating the dependencies for all possible subsets of $\mathbb{C}$:

QUICKREDUCT($\mathbb{C}$, $\mathbb{D}$)
Input: $\mathbb{C}$, the set of all conditional attributes; $\mathbb{D}$, the set of decision attributes.
Output: R, the attribute reduct, $R \subseteq \mathbb{C}$
(1) $R \leftarrow \emptyset$
(2) **do**
(3) $T \leftarrow R$
(4) **foreach** $x \in (C - R)$
(5) **if** $\gamma_{R \cup \{x\}}(\mathbb{D}) > \gamma_T(\mathbb{D})$
(6) $T \leftarrow R \cup \{x\}$
(7) $R \leftarrow T$
(8) **until** $\gamma_R(\mathbb{D}) = \gamma_{\mathbb{C}}(\mathbb{D})$
(9) **return** R

Algorithm 1: The QUICKREDUCT algorithm yields a reduct for a dataset without forming all subsets of $\mathbb{C}$, the set of conditional attributes.

$$
\begin{array}{llll}
\gamma_{\{a,b,c,d\}}(\{e\}) & = 8/8 & \gamma_{\{b,c\}}(\{e\}) & = 3/8 \\
\gamma_{\{a,b,c\}}(\{e\}) & = 4/8 & \gamma_{\{b,d\}}(\{e\}) & = 8/8 \\
\gamma_{\{a,b,d\}}(\{e\}) & = 8/8 & \gamma_{\{c,d\}}(\{e\}) & = 8/8 \\
\gamma_{\{a,c,d\}}(\{e\}) & = 8/8 & \gamma_{\{a\}}(\{e\}) & = 0/8 \\
\gamma_{\{b,c,d\}}(\{e\}) & = 8/8 & \gamma_{\{b\}}(\{e\}) & = 1/8 \\
\gamma_{\{a,b\}}(\{e\}) & = 4/8 & \gamma_{\{c\}}(\{e\}) & = 0/8 \\
\gamma_{\{a,c\}}(\{e\}) & = 4/8 & \gamma_{\{d\}}(\{e\}) & = 2/8 \\
\gamma_{\{a,d\}}(\{e\}) & = 3/8 &
\end{array}
$$

Of the above choices of attributes, the ones that can be chosen as reducts are those with the highest dependency. Hence, the reduct and minimal reduct sets are as follows:

$$
\begin{aligned}
\mathcal{R} &= \{\{a,b,c,d\},\ \{a,b,d\},\ \{a,c,d\},\ \{b,c,d\},\ \{b,d\},\ \{c,d\}\} \\
\mathcal{R}_{min} &= \{\{b,d\},\ \{c,d\}\}
\end{aligned}
$$

2.2 QuickReduct

In terms of computational complexity and memory requirements, the calculation of all possible subsets of a given set is an intractable operation. To solve this problem, the subset search space is treated as a tree traversal. Each node of the tree represents the addition of one conditional attribute to the reduct. Instead of generating the whole tree and picking the best path on it, the path is constructed incrementally using the following heuristic: the next attribute chosen to be added to the reduct is the attribute that adds the most to the reduct's dependency, i.e. the most significant attribute. The search ends when the dependency of the attribute subset equals that of the entire set of conditional attributes. This is dubbed the QUICKREDUCT algorithm as given in algorithm 1 and is similar to the algorithm introduced in Jelonek et al. (1995), where it was used with neural network-based classifiers.

The operation of the QUICKREDUCT algorithm can be illustrated by returning to the example. It starts off with an empty attribute subset: $R = \emptyset$. Now the algorithm evaluates the change in dependency caused by adding an attribute to R:

$$\begin{aligned}
\gamma_{R\cup\{a\}}(\{e\}) = \gamma_{\{a\}}(\{e\}) &= 0/8 \\
\gamma_{R\cup\{b\}}(\{e\}) = \gamma_{\{b\}}(\{e\}) &= 1/8 \\
\gamma_{R\cup\{c\}}(\{e\}) = \gamma_{\{c\}}(\{e\}) &= 0/8 \\
\gamma_{R\cup\{d\}}(\{e\}) = \gamma_{\{d\}}(\{e\}) &= 2/8
\end{aligned}$$

The addition of attribute d provides the highest increase in discernibility. Hence, R becomes $R = \{d\}$ and the algorithm attempts to add another conditional attribute:

$$\begin{aligned}
\gamma_{R\cup\{a\}}(\{e\}) = \gamma_{\{a,d\}}(\{e\}) &= 3/8 \\
\gamma_{R\cup\{b\}}(\{e\}) = \gamma_{\{b,d\}}(\{e\}) &= 8/8 \\
\gamma_{R\cup\{c\}}(\{e\}) = \gamma_{\{c,d\}}(\{e\}) &= 8/8
\end{aligned}$$

Adding either b or c to $\{d\}$ results in perfect discernibility of the dataset. Given a choice of two or more minimal reducts, QUICKREDUCT chooses the first solution encountered, in this case the conditional attribute subset $\{b, d\}$.

There is also a variation of QUICKREDUCT that works in reverse. This revised algorithm starts off with the entire set of conditional attributes, and incrementally removes attributes as long as γ does not decrease. The reverse semantics cause the algorithm to require much more main memory, however, especially for the application at hand, where the dimensionality is in the order of tens of thousands of dimensions.

QUICKREDUCT works like a hill climber. However, γ_R monotonically increases as $\| R \|$ increases (see proof in the Appendix). Hence, there are no local minima and hill climbing is an easy task in this case. Further, because of this monotonic nature, QUICKREDUCT always locates the shortest possible reduct of a dataset when it terminates.

3 Proposed System

3.1 Overview

To address the dimensionality problem typically associated with TC, a modular system was built. Modularity is a major concern in developing this system, as it should be able to co-operate with various existing or newly developed techniques. The system includes separate components for acquiring candidate keywords for texts, reducing the dimensionality of the keyword datasets, and using the resultant examples of labelled documents to classify documents. The application domain chosen to test the system is E-mail, since real-life corpora of E-mail messages are easy to obtain. Most users of E-mail keep folders of messages related in some way, this provides a wealth of training data for the system.

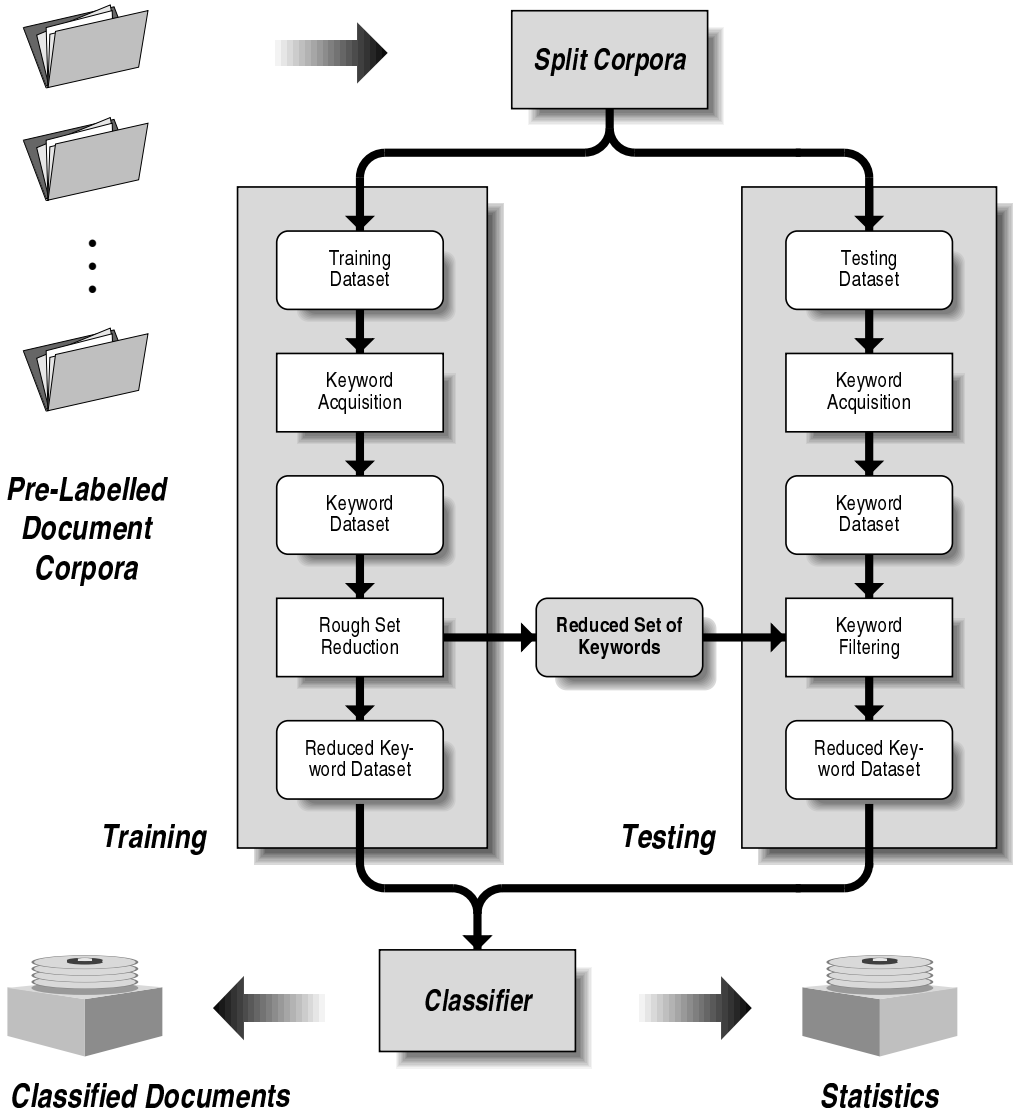

Figure 2: Block diagram showing the flow of data through the proposed system.

The flow of data through the system involves a number of steps, as illustrated in figure 2. Note that this is a proof-of-concept system, and hence the focus is on experimentation, rather than user interaction. Facilities that should be available on a streamlined end user application are absent.

- *Training and Testing Dataset Separation.* This step randomly partitions the messages in a set of folders into two distinct datasets, one for training purposes, one for testing. The partitioning ratio is configurable. The format of the split training and testing datasets is identical. Each dataset represents a set of documents. The labelling of each document is embedded into the dataset. This is used during training, and for validation during testing.

- *Keyword Acquisition.* This module inputs datasets in the format mentioned above. It then acquires keyword sets for each individual document and outputs them as weight-term pairs. Weights are normalised within each document. Four different keyword acquisition modules have been built, as described in section 3.2.

- *Rough Set Dimensionality Reduction.* This reads the dataset and outputs a dataset of reduced dimensionality using the techniques described in section 2.1. The format of this module's output is identical to that of its input, so that this stage can be bypassed for reasons of comparative experimentation on the implicating of attribute reduction.

- *Keyword Filtering.* This simple subsystem filters the keywords discovered by the keyword acquisition module, based on what keywords are deemed important during the training stage. This is functionally equivalent to the RSDR module, with one difference: it reads the reduct set output by the RSDR module and hence does not need to perform dimensionality reduction again.

- *Classification.* Given the results of the previous stage and the testing dataset generated previously, the classifier uses the reduced keyword dataset produced during training as a means of classifying the test data. This stage yields the inferred document classifications and statistics to gauge its success in doing so.

3.2 Keyword Acquisition Module

The modularity of the system allows different keyword acquisition algorithms to be applied. There are currently four different acquisition methods implemented. All four work in similar ways: single words or pairs of consecutive words are considered document indexing terms. For hierarchical domains that use a structure of field-value pairs, the name of the current field is prepended to each term, so that information on where in the document the keyword occurred is retained. All terms are converted to lower case. Each such term is assigned a *weight* proportional to its perceived importance in the current document. The four different weighting methods are:

- The *Boolean Existential Model* metric assigns a weight of one to all keywords that exist in the document. Absent keywords have an implied weight of zero. These weights can ultimately be used by a crisp, Boolean inference engine.

- The *Frequency* metric makes term weights equal to the terms' frequency in the document.

- The *Term Frequency-Inverse Document Frequency* (TF-IDF) metric (Salton and Buckley, 1987) assigns higher weights to keywords which are frequent in the current document, yet uncommon in most others.

- The *Fuzzy Relevance Metric* (FRM) is a highly experimental attempt at judging the relevance of terms based on their place in the frequency histogram. This is, in essence, a distribution-based metric: it applies a bell-curve to assign more relevance to keywords that are more frequent than the average, but not the *most* frequent keywords.

For documents comprising field-value pairs, keyword weights are multiplied by a constant number dependent on the field they were found in. These are field weights, as per the weighted vector representation discussed in Salton and Buckley (1987). Field weighting reflects the relative importance of fields in a document. For example, considering the title, abstract, keywords and body of a paper as fields, the keywords and title possess more information than the abstract and body. Such fields are therefore assigned a higher importance or weight.

Keyword weights are finally normalised within each message to allow for more homogeneous handling of weights in the next stages and to avoid counter-intuitive results as identified by Rijsbergen (1979): documents being erroneously selected in the inference/classification step because a scoring-based similarity metric latched on one very large ordinate in the document vector.

3.3 Classification Modules

For the sake of comparison, and to offer different design choices, three different classification techniques or inference engines were implemented:

- *Boolean Inexact Model*: this uses Boolean matching and scoring methods with inconsistency detection, although no inconsistency threshold is implemented (inconsistencies are detected only if two differing classifications end up with the same score).
- *Vector Space Model*: this uses the angle between document vectors to gauge the similarity of documents (themselves treated as vectors).
- *Fuzzy Reasoner*: this follows the standard approach for constructing fuzzy rule-based systems (Kasabov, 1996). The fuzzy reasoner is not expected to work very well with discontinuous metrics like the Boolean Existential one, since Fuzzy Logic works better with continuous membership values than with crisp, discrete ones. The operation of fuzzy classifiers is not discussed here, for reasons of brevity.

3.4 An Example Application: E-Mail Categorisation

Electronic mail (E-mail) is one of the most venerable services provided on the Internet. E-mail concerns itself with the transmission of information quanta named messages, much like other messaging systems such as conventional mail. The mechanics of transmission are of no interest here, but the important point is that E-mail is transmitted over one or more computer networks using one or more transmission protocols and passing one or more host computers before finally arriving at its destination machine, where it is stored locally for the recipient to access (Hunt, 1997).

Automated E-mail filtering or sorting has many applications. These include:

- Automatically marking certain messages as 'important', or assessing the importance and urgency of messages. Such a filter may be used to bridge two different messaging systems with different characteristics: for instance, to forward urgent messages to their recipient using pager or mobile phone networks.
- Automatically sorting messages into users' *mail folders*. This is suitable where many users share one mailbox (for instance, user support personnel with a single mailbox, but different responsibilities), or where a single user has multiple roles. Additionally, many E-mail users need to classify received messages by their content matter, or based on their sender.
- Auto-responders. These are systems that classify incoming messages based on their content and reply with a pre-composed message. This is sometimes used in corporate environments as a quick and cost-effective method of supporting users' most frequently asked questions about a product.

```
From owner-irr-staff@inf.ed.ac.uk  Tue Aug 17 09:56:33 1999
Date: Tue, 17 Aug 1999 09:58:00 +0100
To: irr-members@inf.ed.ac.uk
From: Austin Tate <a.tate@ed.ac.uk>
Subject: AIAI / IRR Joint Seminar - Nicola Muscettola, NASA Ames - 6-Sep-99

Joint AIAI / IRR Seminar, Division of Informatics, University of
Edinburgh Room F13, 80 South Bridge, Edinburgh EH1 1HN, UK, 3.45pm on
Monday 6th September: Constraint-based planning at 96 million
kilometers from Earth by Nicola Muscettola, RECOM  Technologies,
NASA Ames Research Center Moffet Field, CA 94035-1000, USA.


From owner-irr-staff@inf.ed.ac.uk  Thu Aug 19 17:26:23 1999
Date: Thu, 19 Aug 1999 17:19:16 +0100
From: Julian Richardson <julianr@dai.ed.ac.uk>
To: irr-members@inf.ed.ac.uk
Subject: IRR Seminar, "Current Trends in Game-playing Research"

IRR Seminar announcement: "Current Trends in Game-playing Research" by
Ian Frank, Visiting Fellow, Complex Games Lab, Electrotechnical
Laboratory (ETL) Tsukuba, Japan. 2pm, 23rd August, F13, 80 South
Bridge


From owner-infres@inf.ed.ac.uk  Mon Oct 18 11:46:33 1999
Date: Mon, 18 Oct 1999 11:43:22 +0100 (BST)
From: Alexios Chouchoulas <alexios@dai.ed.ac.uk>
To: infpg@inf.ed.ac.uk, infres@inf.ed.ac.uk, infteach@inf.ed.ac.uk
Subject: REMINDER: Joint IRR/AIAI/ICCS/HCRC Seminar, Tomorrow 19/10

Joint IRR/AIAI/ICCS/HCRC Seminar tomorrow, Tuesday, 19th October 1999,
11am - 12.15pm, South Bridge, Room F10. Alan L. Rector, Professor of
Medical Informatics Department of Computer Science, University of
Manchester. "Clinical ontologies, clinical systems, and clinical
language: Reflections on the GALEN experience"
```

Figure 3: E-Mail corpus α. It contains three messages, each starting with the word `From` followed by the sender and time stamp.

- Filtering out *Unsolicited Commercial E-mail* (UCE, also known as *spam*).

The information quantum of E-mail, the message, is a plain text document. It is semi-structured, in that it contains two sections:

1. A structured *header*, containing a set of *key-value* pairs of character strings in near-arbitrary order. The header contains an arbitrary amount of information, but at the very least the following: the sender's address; the recipient's address; the date the message was sent; and the 'path' it followed to reach its destination.

2. An unstructured *body*, which contains the actual text of the message.

This makes E-mail messages attractive for a number of IF purposes, since the header can easily be parsed to glean information about the message, while more advanced IF techniques have to be applied to the body.

An example is included to show how the proposed system can be trained to differentiate between two classes of E-mail documents, α and β, all the while minimising the set of keywords required to recognise each class. The two corpora/classes of E-mail are given in figures 3 and 4 respectively. The two classes contain seminar announcements from the University of Edinburgh that have been slightly altered for brevity. Corpus α comprises announcements for seminars in the Institute for Representation and Reasoning (IRR) seminar series; β holds announcements meant for the Institute for Communicating and Collaborative Systems and Human Communication Research Centre (ICCS/HCRC). Corpus α contains three messages; β contains two.

```
From owner-infres@inf.ed.ac.uk  Mon Nov 15 09:35:21 1999
Date: Mon, 15 Nov 1999 09:29:39 GMT
From: Seminar Support Team <sst@cogsci.ed.ac.uk>
To: infmsc@inf.ed.ac.uk, infteach@inf.ed.ac.uk
Subject: ICCS/HCRC seminar: Chris Brew talk Fri 19 Nov.

Seminar Series: ICCS/HCRC. Stochastic Feature Grammars:
are they ready yet? Chris Brew, Language Technology Group, Human
Communication Research Centre, University of Edinburgh. 11:00am,
Friday 19 November, 1999. Faculty Room South, David Hume Tower.

For term schedule see
http://www.cogsci.ed.ac.uk/seminars/programme.html


From owner-infres@inf.ed.ac.uk  Fri Nov 12 10:10:53 1999
Date: Fri, 12 Nov 1999 10:06:20 GMT
From: sst@cogsci.ed.ac.uk
To: infmsc@inf.ed.ac.uk, infpg@inf.ed.ac.uk
Subject: ICCS/HCRC seminar *Reminder*: Talk NOW: Bob Logie

Seminar Series: Institute for Communicating and Collaborative Systems,
and Human Communication Research Centre Mental Discovery and
Visuo-Spatial Working Memory. Professor Robert H. Logie, Department of
Psychology, University of Aberdeen.  11:00am, Friday 12 November.
Faculty Room South, David Hume Tower

Anyone wanting to meet with the speaker should email Padraic Monaghan:
pmon@cogsci.ed.ac.uk. For term schedule see
http://www.cogsci.ed.ac.uk/seminars/programme.html
```

Figure 4: E-mail corpus β. It contains two messages, each starting with the word `From` followed by the sender and time stamp.

The two corpora are fed to the system during training. Keyword weights are obtained using the Boolean Existential metric. Running the system first results in a set of keywords for each message. Using field weighting, the keyword extraction method gives the highest weights to the contents of the `From` fields of all the training messages. This is no surprise, since the Boolean metric assigns a weight of 1 to all existing keywords, so that any differences between weights are effectively determined by the field weights themselves. Using any other metric, however, would obtain very different weights for the same keywords. An sample of the keywords for two of the messages is shown below:

Corpus α, message 1	
w	**keyword**
1.00	`from:austin tate`
1.00	`from:a.tate@ed.ac.uk`
	⋮
0.78	`subject:seminar`
0.78	`subject:nasa`
	⋮
0.00	`[body]:room f13`
0.00	`[body]:million kilometers`

Corpus β, message 2	
w	**keyword**
1.00	`from:sst@cogsci.ed.ac.uk`
	⋮
0.78	`subject:iccs/hcrc`
0.78	`subject:seminar`
0.78	`subject:reminder`
	⋮
0.00	`[body]:speaker`
0.00	`[body]:mental discovery`

As described earlier, the form of the keywords contains within it the name of the field where the keyword was found. The field name `[body]` refers to the body of the message. The weights look somewhat discrete because of the Boolean metric and normalisation. Other metrics provide continuous weights.

These keyword sets are then converted into a dataset. Each object in the dataset is one documenet vector, so that the dataset created for this domain will have 5 objects. The dimensionality happens to be 408, one dimension for each unique keyword found in the corpora. Keywords are ordered so that the unordered keyword sets may be converted to ordered sparse vectors (i.e. with missing values). Due to the size of the dataset, it is

not feasible to show the entire dataset for even this simplistic example. A subset of the example will thus be tackled by evaluating a considerably simplified dataset for the two corpora shown above (and their respective keywords).

The first step of keyword acquisition is to obtain an axis ordering, effectively an ordering of the set of keywords. The proposed system sorts keywords by decreasing maximal weight. A sample of the sorted keywords is given below:

Column	Weight	Keyword
c1	1.00	`from:austin tate`
c2	1.00	`from:a.tate@ed.ac.uk`
c3	1.00	`from:sst@cogsci.ed.ac.uk`
c4	0.78	`subject:seminar`
c5	0.78	`subject:nasa`
c6	0.78	`subject:iccs/hcrc`
c7	0.78	`subject:seminar`
c8	0.78	`subject:reminder`
c9	0.00	`[body]:room f13`
c10	0.00	`[body]:million kilometers`
c11	0.00	`[body]:speaker`
c12	0.00	`[body]:mental discovery`

This axis ordering allows the sets of keywords to be converted into an equivalent dataset: a sparse matrix or a set of sparse document vectors. This dataset does not actually have information on what each column represents. As such, it is suitable for generic inference and feature selection systems (although, by using the table above, column numbers may be mapped back to the keywords they represent):

Object	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	Class
α_1	1.0	1.0	—	0.78	0.78	—	—	—	0.0	0.0	—	—	α
β_2	—	—	1.0	—	—	0.78	0.78	0.78	—	—	0.0	0.0	β

In this simplistic example, all keywords are mutually exclusive, but this is never the case in anything more complicated. Running RSDR in this case would reduce the dataset to just one conditional attribute: the first one. In fact, any of the 12 attributes listed above would do, but RSDR picks the first minimal reduct it finds for reasons of speed. This is why it is needed to order the keywords: the intuitively most suitable ones are towards the left edge of the dataset.

For this simplified example, the application of a rule induction algorithm to this reduced dataset would lead to the following simple ruleset:

$$c1 = 1.0 \Rightarrow \alpha$$

$$c1 \neq 1.0 \Rightarrow \beta$$

In reality, of course, datasets, reduced datasets and induced rulesets are generally far more complicated. The next section presents typical experimental results for applying the proposed system to real E-mail classification.

4 Experimental Results

4.1 Datasets Used

Six E-mail folders belonging to the authors were used as experimentation datasets. The folders were chosen because they offered a wide spectrum of different features to be tested: different natural languages, homogeneous and heterogeneous writing styles and formats, different content and similar linguistic elements in different contexts. Folders ranged in size from 159 to 380 messages, with an average of 225 messages. In terms of memory footprint, they ranged from 265 to 1,251 kbytes, with an average of 634 kbytes. There are combinations of extremely long and extremely short messages in the folders.

Three of the folders contained messages in English; two contained messages in Greek; one comprised mostly English messages, with occasional messages in Greek. Greek was not an arbitrary choice: like many languages, it is expected to provide a challenge to any keyword acquisition system: its complex system of prefixes and suffixes should increase the number of different keywords with similar semantics. The lack of a one-to-one mapping between the Latin and Greek alphabets causes more confusion, as some people prefer to write phonetically, others simply type as they would on a Greek typewriter, yet others mix and match depending on personal taste. It should be noted that all such Greek messages use the Latin alphabet, not a Greek character encoding like the ISO-8859-7, since very few machines at the University of Edinburgh offer such a feature.

As this is a real-world set of E-mail folders, speling mistakes are inevitable, a fact that makes it difficult for language-dependent systems to operate. All folders have been hand-classified by the authors with some reasonable rationale: personal folders contain messages from the same person and mailing list folders contain messages from the same list. Other E-mail users may partition their E-mail messages differently, for instance based on content matter. It is expected that the system should be able to generalise rules to classify new messages according to this partitioning. It is, of course, not expected that the system will be able to distinguish between semantic differences in texts. For instance, it is unlikely to be successful to train the system to detect humorous messages, unless such messages use a different vocabulary from that of other types of training messages.

4.2 Dimensionality Reduction

Numerous sets of two to six folders were selected at random. The folders of each set were split into training and testing sets at a 1:4 ratio. The training sets were used to create sets of classification means using each of the four keyword acquisition metrics, as described previously. The results of dimensionality reduction are shown in table 3. The first two numeric columns show the average dimensionality before and after reduction. The reduction itself is dramatic, decreasing the width of datasets by approximately 3.5 orders of magnitude (i.e. reduction by a factor of around 3,162). This indicates that the text classification domain has a very high degree of redundancy. This can be taken advantage of in an effort to improve the efficiency of TC/IF/IR systems by the use of the present approach.

As expected, the results are very similar for all four techniques which give a measure of the information

Table 3: Average dimensionality reduction for different keyword acquisition metrics.

Metric	Average Attributes before	Average Attributes after	Average Reduction (Orders of Magnitude)
Existential	33,305	10.16	3.52
Frequency	35,328	9.63	3.56
TF-IDF	33,327	8.97	3.57
FRM	35,328	9.63	3.56

content of a term. As long as this measure is reasonably accurate, RSDR is able to remove redundant terms from the dataset. It is interesting to observe that post-reduction dimensionality is not proportional to dimensionality before reduction. RSDR does not simply remove a certain proportion of attributes; rather, it removes *all* redundant or useless information from the data. The reduction factor is more or less constant. This implies that around one in every 3,162 document terms can be used as a co-ordinate keyword in E-mail messages.

The average post-reduction dimensionality displays an interesting pattern: it provides an indirect measure of the success of each of the four metrics at locating co-ordinate keywords. The better the metric, the less the post-reduction dimensionality (i.e. the more informative the surviving individual attributes). In this regard, TF-IDF yields the best results, followed closely by the Frequency and FRM metrics and then the Existential metric. This is as generally expected. Since the Existential metric is Boolean, it granulates the keyword relevance more than the other techniques. This requires more keywords to make up for the lesser information content of the weights. The similarity of the Frequency and FRM metrics can also be easily explained: FRM is a function of the term frequency, representing frequency using fuzzy memberships.

For complex fuzzy TC applications, hence, the FRM could be avoided: normalised term frequency would be enough to provide the required information. This conforms to the assertion of Rijsbergen (1979) that even simply using the frequency as a relevance metric yields good text categorisation results. It may therefore be argued that the frequency metric is preferable to TF-IDF wherever efficiency is the primary concern — it provides almost the same results as TF-IDF, but without the computational cost of the latter.

4.3 Information Content of Rough Set Reducts

Having established that the RSDR method significantly reduces the dimensionality of keyword datasets, it should be established whether the reduct set of keywords provides more information for the classification of E-mail messages than any other arbitrary set of keywords. For this, three message folders were chosen at random. They were again split into training and test sets at a 1:4 ratio. The reduct set was obtained for the training set and used to classify the E-mail messages in the test dataset. For simplicity, the Boolean Existential metric was used as the keyword acquisition metric and the BIM classifier was employed to classify the messages. RSDR reduced the dataset to just eight keywords.

Random sets of one to twenty keywords were chosen from the training set and thus twenty keyword datasets containing the corresponding attributes were constructed. These were then used to classify the same set of three folders as above using the same combination of metric and inference module. 100 runs of

this procedure were performed to provide a representative sample. It would have been desirable to evaluate classification accuracies for all combinations of attributes, but this is completely infeasible, given the pre-reduction dimensionality of the keyword datasets. The results are shown in figure 6. The Rough Set reduct (of eight keywords) obtained an accuracy of 75%. However, the random choices of attributes exhibited significantly reduced classification accuracy, at most managing 40%, but with the lowest accuracy dropping to zero. Please note that, as demonstrated, in the next experiment, these results are not typical for the present methodology's. They were due to the arbitrary choice of keyword acquisition metric and inference method. A better choice for these components can yield accuracy as high as 98–99.5%.

It might be expected that the average classification rate would increase slightly as the dimensionality was increased. This did not happen, but twenty attributes are a fraction of the initial dimensionality of around 30,000, so any potential trends are not evident at this scale.

As this is only a sampling of the set of all combinations of attributes, it may be argued that freak cases of high accuracy cannot be ruled out. However, for lack of heuristic to lead to those, the RSDR technique will have to do, whilst an exhaustive, brute force search of all combinations is, to say the least, intractable.

4.4 Comparative Study of TC Methodologies

Results presented above have shown the success of the present work in reducing the dimensionality of datasets used in TC tasks, all the while retaining the information content of the original datasets. To measure the effect of the work on the actual classification task, a comparative study of the integration of the RSDR technique with different keyword acquisition methods and inference algorithms was carried out. All twelve combinations of the four keyword acquisition metrics and the three inference engines were investigated. Each experiment involves selecting random sets of 2 to 6 folders and training the system to classify previously unseen messages into these categories.

Traditionally, IF and IR systems are benchmarked based on their *precision* and *recall*. Precision is defined as the ratio of the number of retrieved documents that are relevant to a user's query over the total number of documents retrieved as a response to the query. Recall is defined as the ratio of the number of relevant documents retrieved over the total number of relevant documents indexed by the system.

However, a *relevant/irrelevant* partition is a dual-class categorisation task. The present work attempts to classify documents into an arbitrary number of classes. Hence the generation of precision/recall graphs would convey considerably less information about the success of the system.

A more conventional metric, classification accuracy, is used instead. This is valid because the datasets used have similar sizes for the underlying document classes.

Each set of folders was split into training and test data, at the usual 1:4 ratio. Keyword datasets were acquired using one of the four keyword acquisition subsystems; the datasets were reduced and a classification test was performed using the test dataset. Classification accuracy and dimensionality reduction statistics were gathered to provide an insight into the working of the corresponding pairing of keyword acquisition and inference method, in combination with the RSDR.

The system was run 20 times for each set of folders within each of the twelve experiments, in order to obtain

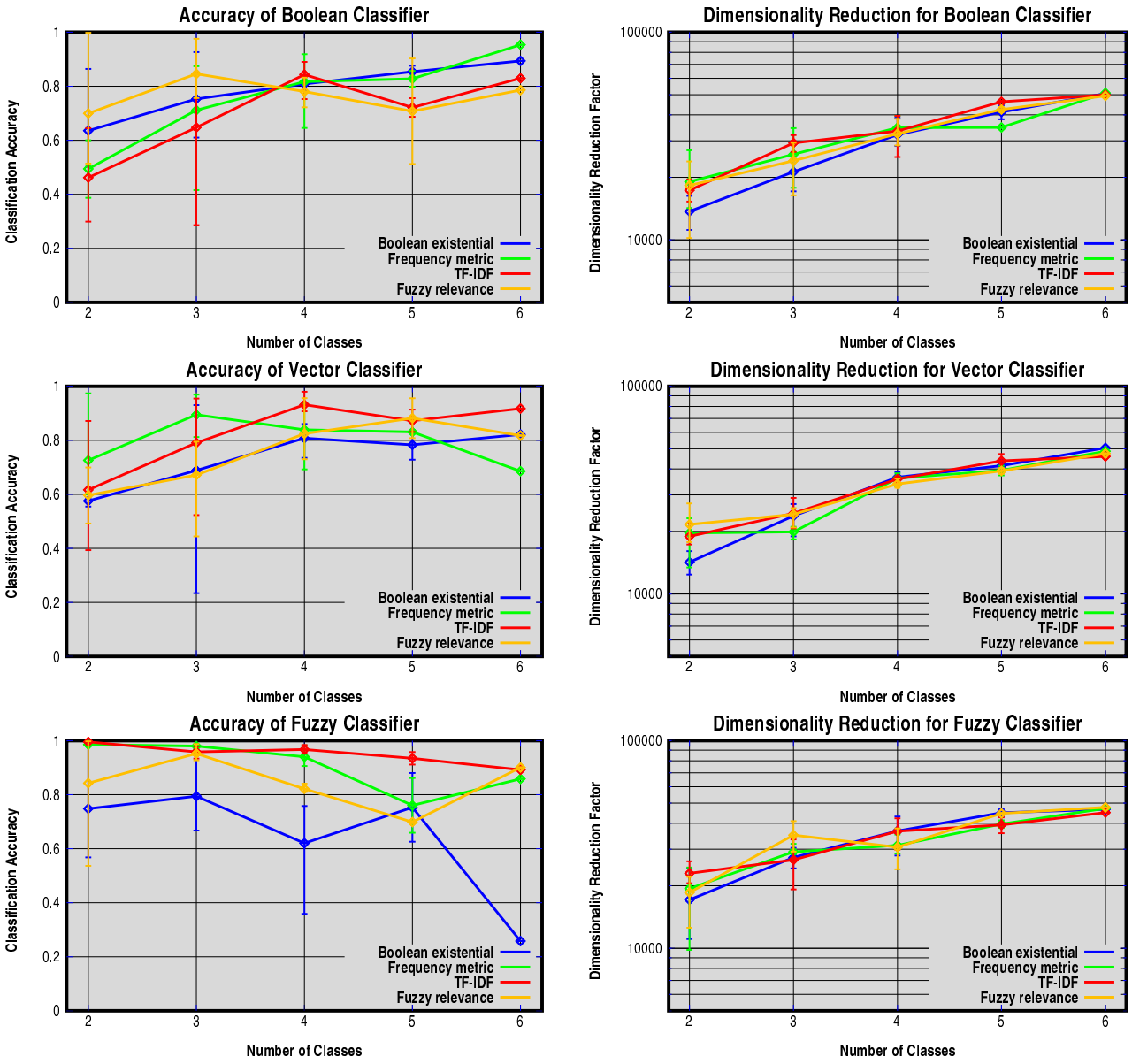

Figure 5: Comparison of the success of the four keyword acquisition metrics used with the three different inference engines. The left column shows classification accuracy; the right dimensionality reduction.

a more representative statistical sample. The results are presented in figure 5. Six graphs are shown overall; the left column shows average classification accuracy for each of the three types of classifier or inference engine. The four keyword acquisition metrics are plotted in different colours. The vertical axis represents classification accuracy; the horizontal axis represents the number of classes (the number of E-mail folders used). The right column of the figure gives average dimensionality reduction plotted on a logarithmic scale against the number of folders.

The most obvious observation is that dimensionality reduction is similar in all twelve combinations of techniques. It increases as the number of folders increases. This implies that, despite the increase in total number of acquired keywords, there is little or no increase in the number of information-rich keywords. Hence, the RSDR technique is expected to provide consistent results independently of the number of corpora of documents.

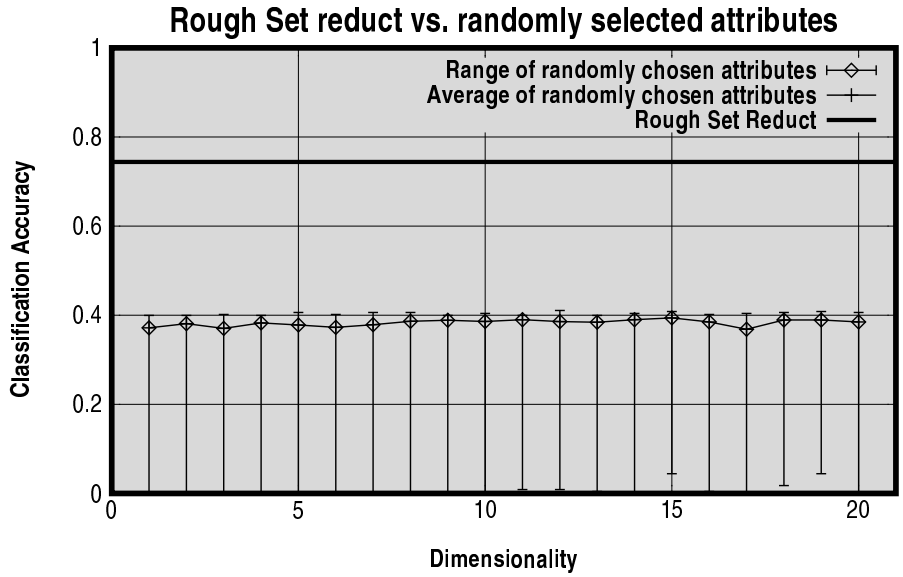

Figure 6: Verifying the information content of the Rough Set reduct. The thick line represents the accuracy of the Rough Set reduct. Minimum, maximum and average classification accuracies for randomly selected keyword datasets of one to twenty dimensions are also shown.

These results show that the TF-IDF metric performs best in almost all tests. The Boolean classifier gave good results with the FRM. The FRM was more or less consistent in its results with this classifier, with a rough average accuracy of 75%: not excellent, but acceptable, although certain individual runs went very close to perfect classification. As the number of folders increased, the Boolean and frequency metrics gave better and more consistent results. Classifying all six folders with a combination of the frequency metric and the Boolean classifier yielded excellent results, around 95% accuracy.

The VSM classifier proved more predictable: runs with the traditional combination of TF-IDF and the VSM offered good accuracy in the range 60%–99%. Again, overall accuracy seems to increase as more corpora are added to the problem. The Fuzzy classifier proved to give excellent classification using the TF-IDF metric: 90%–100% accuracy with consistent behaviour and very little variance. The FRM did acceptably well, but was outperformed by the frequency metric, coming relatively close to the TF-IDF. As expected, the Boolean metric failed to provide consistent results, displaying widely varying accuracy. Overall, the fuzzy classifier has so far provided the best results for this system, especially when used with the TF-IDF metric.

An attempt was made to compare the Rough Set-assisted systems with more conventional TC systems which did not incorporate strong dimensionality reduction techniques, by bypassing RSDR and executing the inference engines on the unreduced datasets. Unfortunately, this experiment failed to the point where no results have been obtained. The unreduced datasets had, on average, around 33,000 keywords to be used by the inference engines as rule conditions or document vector ordinates. With approximately 500 messages in the training set, pre-reduction inference required unacceptably long periods of time and had to be terminated before any usable results could be gathered.

A 500-example dataset of 33,000 keywords contains 33,001 space delimited single- or double-digit numbers and measures in excess of 32 Mbytes of peripheral memory. Evaluating complex similarity metrics on each and every one of those 500 examples of 33,000 keywords to measure their similarity against a new document takes

a rather long time. Fortunately, an average dimensionality decrease from 33,000 to 10 attributes manages to reduce the dataset to a few hundreds of bytes of uncompressed, ASCII keyword weights. CPU processing times decrease similarly, which is especially significant for the more complicated systems, like those incorporating the VSM.

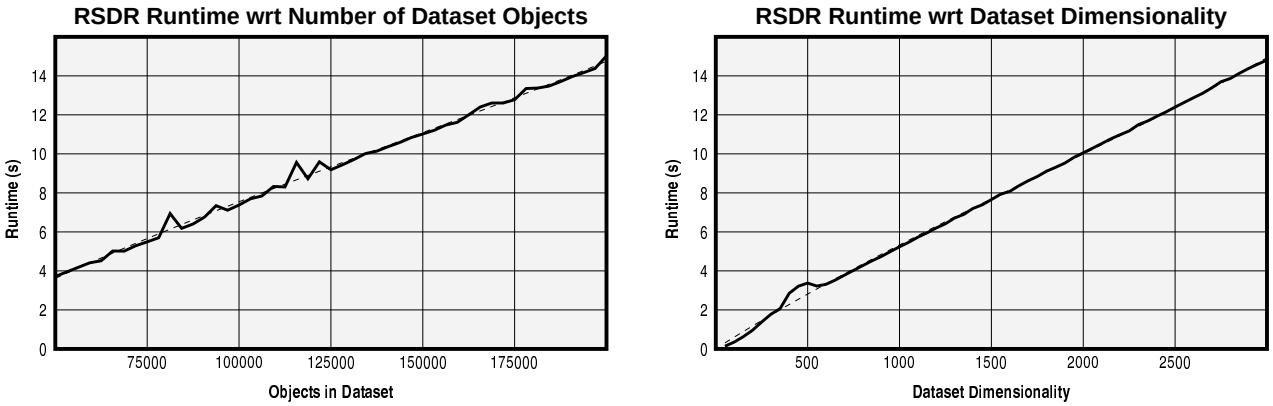

Figure 7: RSDR's runtime efficiency with respect to the number of dataset objects (dataset length) and dataset dimensionality (dataset width). The streamlined, experimental version of the algorithm manages to obtain very nearly linear operation. The dashed thin lines show projected linear runtime and clearly match the observed runtimes of the algorithm. All times are in seconds.

4.5 Efficiency Considerations of RSDR

Although the RSDR method alleviates many of the efficiency considerations of TC approaches, such as runtime and main and peripheral memory footprints, it would be prudent to know RSDR's own efficiency. RSDR's complexity should be lesser than that of TC techniques for it to be effective on a theoretical basis. For practical purposes, the algorithm's actual runtime should also be small enough for its use to offer speed gains.

To test this, two experiments were performed. In the first, a dataset of four conditional attributes and one decision attribute was used. The dataset contained 150 objects. A large number of RSDR runs were carried out. In each, RSDR was required to reduce a larger version of the same dataset, formed by repeatedly increasing the length of the dataset. This was done by duplicating the dataset's objects (without altering the information content of the data, of course). Each reduction was performed three times, with the average time calculated. A graph of the number of dataset objects against the time required to reduce the dataset is shown in figure 7 (left graph). Barring a few irregularities (most likely caused by the algorithm running on a multi-tasking UNIX system), the graph exhibits linearity. Reduction times are also very satisfactory, with around 7.5 seconds needed per 100,000 objects.

The second experiment involved the dimensionality of the dataset. The dataset contained 210 objects, with dimensionality increasing progressively by the addition of columns of noisy data. Numerous different datasets were constructed with 50,000 to 200,000 attributes. Three runs per dataset were performed, with average time shown in the right graph in figure 7. Runtime was once more close to linear. Although an intuitive understanding of QuickReduct implies that, for a dimensionality of n, $n!$ evaluations of γ may be performed for the worst-case dataset, the actual observed runtime remains apparently linear at the scales shown here.

Thus, although the *worst-case* run-time complexity is O($n!$), the average case requires considerably less effort. In fact, if all the information rich attributes are in the first few columns of the dataset, then it does not matter to QUICKREDUCT how many other attributes exist in the dataset). Therefore, the best-case complexity is close to O(1). Average complexity was experimentally determined to be approximately $O(n)$.

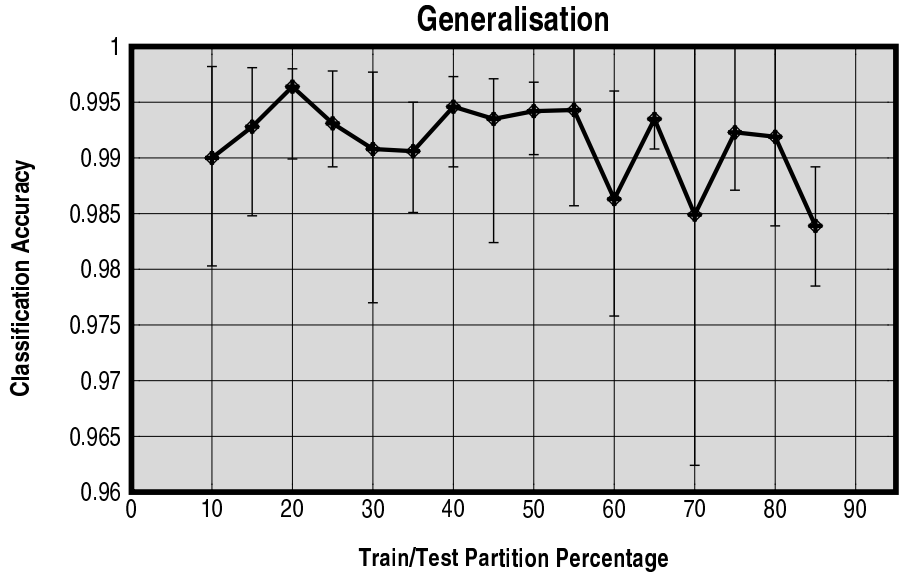

Figure 8: The system's ability to generalise given different training/test partition ratios.

4.6 Generalisation

To accelerate and simplify the training process, the proposed system should be able to generalise using as few training examples as possible. This aids in cases where there is a lack of training examples. To test the system's ability to generalise, two folders were chosen at random. The folders were split into training and test datasets, the system was trained and its classification accuracy tested. TF-IDF was used to measure the relevance of acquired keywords and the fuzzy reasoner was used to classify the unseen E-mail messages. A large number of runs of the experiment were performed with varying training/testing ratios. There were five runs per ratio, with ratios ranging from 1:10 to 8.5:10. The results are shown in figure 8.

These results are surprising in that accuracy is consistently high, with no clearly discernible trend, although there exists a reasonable amount of variation between different runs. Even when trained with merely 10% of messages in a folder, the system managed to generalise enough to be 98%–99.5% accurate.

The reason for this high accuracy rests in the fact that E-mail messages in human-classified folders typically follow a relatively simple set of rules. For example, a personal folder might contain messages from a single sender. The RSDR quickly locates what messages have in common and uses it for the classification. Assuming the training sample is representative of the folder, generalisation of the system becomes almost perfect.

5 Conclusion

This work has proposed a method, based on Rough Set theory, of significantly reducing the dimensionality of datasets used in Text Categorisation tasks, without removing important information. The applicability of this Rough Set Dimensionality Reduction technique to various different types of TC systems has been successful. Data dimensionality can be substantially reduced, which in turn allows for considerable improvement in system efficiency and reduction of the storage requirements of the original datasets.

The system is able to locate intuitively obvious (but difficult for humans to extract) classification means. This transparency is an appealing feature, since, by contrast, VSM-based techniques are relatively opaque, leading to immense files of real numbers. Rough Set-reduced prototypical documents tend to be very small and easily read by a human aware of the format used.

The RSDR technique discussed herein has also proved to be able to generalise based on a minimum of training data. Although most inference techniques and keyword acquisition methods offer acceptable accuracy, the famous and traditional TF-IDF metric provides the most information content. The combination of TF-IDF and a trained fuzzy classifier gives very accurate classification results.

Unfortunately, since the corresponding experiment failed to produce usable results, it is not possible to measure the drop in accuracy incurred by using the RS approach due to the exceedingly large dimensionality of the original keyword sets. However, with 90%–100% post-reduction accuracy, any existing drop in accuracy (at most 10%) is within acceptable bounds, given the significant increase in inference speed and the reduction of the size of datasets. It also seems that the reasonable computational effort added by the use of Rough Set techniques is well-justified in that it reduces the workload of the inference engine, which may employ algorithms of high complexity with respect to the dimensionality of the data. The RSDR-based training process is not normally invoked often, since the document classes do not change quickly. Training may thus be updated after relatively long periods of time. The system's ability to generalise, at least given reasonably constructed training data, is also satisfactory and allows it to be trained on a minimum of information. This reduces the time required for a TC system to become productive.

There are several different directions along which this work could proceed. An application-oriented extension would be to develop a fully interactive application. It would work by initialising its knowledge base, building classification means for the user's current collection of E-mail folders, and then attempting to classify incoming mail. In fact, this has been experimentally established to be feasible and viable by the present work. Extensions should allow inconsistencies to be detected and brought to the user's attention for a more educated opinion. The user's decision would provide feedback to the system to improve its future accuracy.

The use of Rough Set techniques in this work assumes that the system has been given a complete description of the domain, as often assumed by many existing TC approaches. This is, however, hardly the case for complex domains like E-mail. Many unknown values may exist. This will require the development of RS-based methods for rigorous handling of incomplete training data — a subject that has recently attracted attention in Stefanowski and Tsoukiàs (1999). Similar work along this direction is also being carried out at Edinburgh.

A further, experimental direction would be to apply the proposed approach to domains other than E-mail

messages. Another piece of interesting future work, combining the theoretical, implementation-related and application-dependent aspects, would be to investigate the feasibility of an RSDR algorithm with incremental training capabilities. This is not possible for the system described herein, but such an extension would be welcome in further reducing the computational cost of obtaining successful TC systems that work alongside the user for better results. Promising work that integrates incremental rule learning and Rough Set reduction has been established (Shen and Chouchoulas, 1999b,a), though applied to domains rather different from text categorisation. It is envisaged that similar work could be adapted to suit this future development.

A Proof of QuickReduct Monotonicity

Theorem: Suppose that R is a subset of the set of conditional attributes, x is an arbitrary conditional attribute that belongs to the dataset and D is the set of decision attributes that dataset objects are classified into. Then, $\gamma_{R\cup\{x\}}(D) \geq \gamma_R(D)$.

Proof: From expression 2, where the $\otimes$ operator is defined as a variation of the Cartesian product, it is known that:

$$\left\| \bigotimes \{q \in (R \cup \{x\}) : \mathbb{U}/\mathrm{IND}(\{q\})\} \right\| \geq \left\| \bigotimes \{q \in R : \mathbb{U}/\mathrm{IND}(\{q\})\} \right\| \tag{15}$$

This, in conjunction with the definition of the positive region given in equation 6, leads to the following:

$$\left\| \bigcup \{A : A \in \mathbb{U}/\mathrm{IND}(R \cup \{x\}), A \subseteq (R \cup \{x\})\} \right\| \geq \left\| \bigcup \{A : A \in \mathbb{U}/\mathrm{IND}(R), A \subseteq R\} \right\| \tag{16}$$

This holds because the cardinality of a union of sets monotonically increases as the cardinality of the sets increases. From this, and from the definition of the positive region given in equation 6, it follows that

$$\left\| \bigcup_{A\in D} \underline{(R \cup \{x\})A} \right\| \geq \left\| \bigcup_{A\in D} \underline{R}A \right\| \tag{17}$$

Given any non-empty universe $\mathbb{U}$, $\|\mathbb{U}\| \geq 1$. Thus,

$$\frac{\|\mathrm{POS}(R \cup \{x\})\|}{\|\mathbb{U}\|} \geq \frac{\|\mathrm{POS}(R)\|}{\|\mathbb{U}\|}, \tag{18}$$

That is: $\gamma_{R\cup\{x\}}(D) \geq \gamma_R(D)$.

References

Chouchoulas, A. and Qiang Shen. A rough set-based approach to text classification. In *Proceedings of the Seventh International Workshop on Rough Sets (Lecture Notes in Artificial Intelligence, No. 1711)*, pages 118–127, November 1999. ISBN 3-540-66645-1.

Das-Gupta, P. Rough sets and information retrieval. In *Proceedings of the Eleventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Set Oriented Models, pages 567–581, 1988.

Hunt, C. *TCP/IP Network Administration.* O'Reilly and Associates, second edition, December 1997. ISBN 1-56592-322-7.

Jelonek, J., Krzysztof Krawiec, and Roman Slowinski. Rough set reduction of attributes and their domains for neural networks. *Computational Intelligence*, 11(2):339–347, 1995.

Kasabov, N. K. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering.* The MIT Press, 1996. ISBN 0-262-11212-4.

Keen, D. and A. Rajasekar. Rough sets and data dependencies. In V. S. Alagar, S. Bergler, and F. Q. Dong, editors, *Proceedings of the Workshop on Incompleteness and Uncertainty in Information Systems (IUIS'93)*, Workshops in Computing, pages 87–101, London, UK, October 1994. Springer. ISBN 3-540-19897-0.

Martienne, E. and Mohamed Quafafou. Learning fuzzy relational descriptions using the logical framework and rough set theory. In *Proc. of the Seventh IEEE International Conference on Fuzzy Systems.* IEEE Press, 1998.

Pawlak, Z., Rough sets. *International Journal of Computer and Information Sciences*, 11(5):341–356, October 1982.

Pawlak, Z., *Rough Sets: Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishers, Dordrecht, 1991.

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

Shen, Q. and Alexios Chouchoulas. Data-driven fuzzy rule induction and its application to systems monitoring. In *Proceedings of the Eighth IEEE International Conference on Fuzzy Systems*, pages 928–933, Seoul, Korea, August 1999. IEEE Press.

Shen, Q. and Alexios Chouchoulas. Combining rough sets and data-driven fuzzy learning. *Pattern Recognition*, 32(12):2073–2076, 1999.

Stefanowski, J. and Alexis Tsoukiàs. On the extension of rough sets under incomplete information. In *Proceedings of the Seventh International Workshop on Rough Sets (Lecture Notes in Artificial Intelligence, No. 1711)*, pages 73–81, November 1999. ISBN 3-540-66645-1.

van Rijsbergen, C. J. *Information Retrieval.* Butterworths, London, United Kingdom, 1979.