Classification of Text Documents Using a Logical Analysis Approach.

Salvador Nieto sanchez
Louisiana State University and Agricultural & Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_disstheses

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_disstheses/7115

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Historical Dissertations and Theses by an authorized administrator of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0800

UMI®

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
CLASSIFICATION OF TEXT DOCUMENTS USING
A LOGICAL ANALYSIS APPROACH

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in
The Interdepartmental Program in Engineering Science

by
Salvador Nieto Sánchez
B.S., Centro Nacional de Enseñanza Técnica Industrial, México, 1978
B.S., Centro Nacional de Enseñanza Técnica Industrial, México, 1980
M.S. in Industrial Engineering, Louisiana State University, 1995
M.S. in Engineering Science, Louisiana State University, 1998
December 1999

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
ACKNOWLEDGMENTS

With this work, I would like to express my gratitude to the following sets of persons:

Set 1: God = G
Set 2: My Family = A = {Maria, Itzel, Moises, Pablo, Ana}
Set 3: My Parents = B = {Jose, Simona, Miguel, Margarita}
Set 4: Sponsors = C = {LSU-Grad School, LSU-ISO, LSU-IMSE, Institute of International Education, Fulbright, CONACyT, DOE}
Set 5: Professors = D = {All my professors}
Set 6: Operation Research = E = {Researchers in this field}
Set 7: Professors who preach with "The Turtle and the Hare" tale = T.

Furthermore, it can be easily verified that these sets satisfy the following properties:

Property 1: A ∪ B ∪ C ∪ D ∪ E ∪ “me” ∈ G (proof is let to the reader).
Property 2: | A | = 4.
Property 4: A ∩ B ∩ E = φ.
Property 5: A ∪ B ∪ C ∪ D = Their support helped me to achieve this academic degree.
Property 6: D ∩ E = His instruction and support were decisive to get this degree.
Property 7: | D | ≈ 75.
Property 8: | E | = Tends to be a very large number.
Property 9: | C ∩ D ∩ E ∩ T | = 1.

I want to express my gratitude to Drs. E. Triantaphyllou, D. Gosh, D. Kraft, T.W. Liao, T. Ray, J. Chen, and B. Marx (Dean’s representative), my thesis committee members, for their invaluable suggestions to complete this work. Special thanks to Dr. Triantaphyllou for his constant encouragement throughout all this time at LSU. He also guided me to complete two Master degrees at LSU. Indeed, it is a pleasure to work with him.

I also want to thank the Institute of International Education, the Fulbright Commission, and CONACyT (Mexico) for the partial financial support to conduct this research. I also want to express my gratitude to Mr. Tom Curtis from the U.S. Department
of Energy (DOE), Office of Declassification, for all his valuable assistance. The U.S. DOE expressed the need to solve a challenging research problem and provided the financial support in terms of a contract via the Southern University. I also would like to recognize the great assistance at various stages of this research from Dr. Emmanuel Osagie, the Vice-Chancellor for Research at Southern University in Baton Rouge who was a close partner of this project.
RECONOCIMIENTOS

Con este trabajo, quisiera expresar mi gratitud a los siguientes conjuntos de personas:

Conjunto 1: Dios = G
Conjunto 2: Mi Familia = A = {María, Itzel, Moises, Pablo, Ana}
Conjunto 3: Mis Padres = B = {José, Simona, Miguel, Margarita}
Conjunto 4: Patrocinadores = C = {Escuela de Graduados de LSU, LSU-ISO, LSU-IMSE, Instituto para la Educación Internacional, Fulbright, CONACyT, DOE}
Conjunto 5: Profesores = D = {A todos mis profesores}
Conjunto 6: Investigación de Operaciones = E = {Investigadores en este campo}
Conjunto 7: Profesores que predicen con el cuento de "La Tortuga y la Liebre" = T.

Adicionalmente, se puede demostrar fácilmente que estos conjuntos satisfacen las siguientes propiedades:

Propiedad 1: A ∪ B ∪ C ∪ D ∪ E ∪ "Yo" ∈ G (la prueba se deja al lector).
Propiedad 2: |A| = 4.
Propiedad 4: A ∩ B ∩ E = φ.
Propiedad 5: A ∪ B ∪ C ∪ D = Su soporte me ayudó a obtener este grado académico.
Propiedad 6: D ∩ E = Su apoyo fue decisivo para lograr este grado.
Propiedad 7: |D| = 75.
Propiedad 8: |E| = Tiende a ser un número muy grande.
Propiedad 9: |C ∩ D ∩ E ∩ T| = 1.

También quiero expresar mi gratitud a los Drs. E. Triantaphyllou, D. Gosh, D. Kraft, T.W. Liao, T. Ray, J. Chen, y B. Marx (representante del "Dean"), los miembros del comité de mi tesis, por su invaluable soporte y sugerencias para completar este trabajo. Especialmente, quiero expresar mi gratitud al Dr. Triantaphyllou por su constante soporte y aliento durante todo este tiempo en LSU. El también me guió para
completar dos grados de Maestría en LSU. Verdaderamente, es un placer trabajar con Dr. Triantaphyllou.

También quiero agradecer al Instituto de Educación Internacional, a la Comisión Fulbright, y a CONACyT (México) por su soporte económico para conducir esta investigación. También quiero expresar mi gratitud al Sr. Tom Curtis de la Oficina de Desclasificación del Departamento de Energía (DOE) de los E.U.A por su invaluable asistencia. Este departamento expresó la necesidad de resolver un problema y otorgó el soporte económico por medio de un contrato con la “Southern University” en Baton Rouge, Louisiana. También quisiera reconocer toda la ayuda que el Dr. Emmanuel Osagi me brindó durante varias etapas de esta investigación. El Dr. Osagi es el Vice-Chancellor de Investigación de “Southern University”.
# TABLE OF CONTENTS

**ACKNOWLEDGMENTS** .................................................................................................... ii

**RECONOCIMIENTOS** ................................................................................................... iv

**LIST OF TABLES** ......................................................................................................... viii

**LIST OF FIGURES** ........................................................................................................ ix

**ABSTRACT** ....................................................................................................................... xi

1 **INTRODUCTION** ......................................................................................................... 1

2 **PROBLEM DESCRIPTION** .......................................................................................... 7
   2.1 Background ............................................................................................................... 7
   2.2 Problem Definition ................................................................................................. 8

3 **LITERATURE REVIEW** .............................................................................................. 11
   3.1 Problem 1: Selection of Text Descriptors ............................................................... 11
      3.1.1 The Principle of Least Effort ..................................................................... 11
      3.1.2 The Vector Space Model .......................................................................... 12
      3.1.3 A Genetic Algorithm Approach ................................................................. 12
      3.1.4 An All Words Approach ........................................................................... 14
      3.1.5 Summary of Methodologies for Selecting a Set of Keywords .................. 14
      3.1.6 Indexing of Documents ........................................................................... 15
   3.2 Problem 2: Classification of Documents ............................................................... 17
      3.2.1 A Fuzzy Set Approach ............................................................................. 17
      3.2.2 A Semantic Analysis Approach ................................................................. 19
      3.2.3 A Neural Networks Approach .................................................................. 20
      3.2.4 A Logical Analysis Approach ................................................................... 21
      3.2.5 A Vector Space Model Approach ............................................................... 26
      3.2.6 Summary of Classification Methodologies .............................................. 28
   3.3 Problem 3: The Guided Learning Approach ............................................................ 29
   3.4 Problem 4: Incremental Learning with the OCAT Algorithm ............................... 32

4 **METHODOLOGY** ....................................................................................................... 35
   4.1 Methodology for Problem 1: Selection of the Best Content Descriptors .......... 35
      4.1.1 Selection of a Sample of Training Documents .......................................... 35
      4.1.2 Methodology for Indexing the Documents .................................................. 36
         4.1.2.1 Construction of a Dictionary of Unique Words .................................. 36
         4.1.2.2 Construction of a Dictionary of Unique Patterns of Words .................. 37
         4.1.2.3 Indexing of the Documents ................................................................ 38

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
LIST OF TABLES

Table 3.1. Measures of Vector Similarity ................................................................. 13

Table 4.1. Number of Documents that were Randomly Extracted from each Class ... 35

Table 6.1. Rules Constructed for the above Eight Surrogates ............................... 58

Table 6.2. Average Number of Indexing Words Used in each Experiment ............... 59

Table 6.3. Summary of First Experimental Condition:
Leave-One-Out Cross Validation ............................................................................. 60

Table 6.4. Summary of Second Experimental Condition: 30/30 Cross Validation ... 60

Table 6.5. Comparison of the Classification Accuracy of
the VSM and OCAT Algorithms ........................................................................... 63

Table 6.6. Sign Test for the Accuracy of the OCAT and VSM Algorithms ............. 64

Table 6.7. Percentage of Training Documents at which the OCAT Algorithm
became 100% Accurate ......................................................................................... 65

Table 6.8. Training Documents Needed to Extract Rules that Could Classify
Correctly all Documents ......................................................................................... 70

Table 6.9. Data for the Statistical Comparison of the Classification
Accuracy Between NILE and ILE approaches ....................................................... 70

Table 6.10. Number of Clauses at the End of the Experiment ................................. 73

Table 6.11. Data for the Statistical Comparison of the Number
of Clauses Constructed by the NILE and ILE approaches ................................. 73

Table 6.12. CPU Time in Seconds at the End of the Experiment ............................ 76

Table 6.13. Data for the Statistical Comparison of the CPU Times Employed
by the NILE and ILE approaches to Construct a Boolean Function ... 77
LIST OF FIGURES

Figure 1.1. An IF-THEN rule to verify the origin of documents A and B .............. 5

Figure 3.1. A collection of N documents and their surrogates .......................... 16

Figure 3.2. The One Clause at a Time (OCAT) algorithm for the CNF case ........... 22

Figure 3.3. The heuristic combined with the OCAT approach
[Deshpande and Triantaphyllou, 1998] .................................................... 23

Figure 3.4. A set of four positive examples and a set of six negative examples ...... 25

Figure 3.5. A set of six positive examples and a set of four negative examples ...... 25

Figure 3.6. The Vector Space Model (VSM) algorithm ........................................ 26

Figure 3.7. Possible outcomes when the classification decisions
of the positive and negative rules are considered simultaneously ........ 31

Figure 4.1. Algorithm for updating a Boolean function that rejects
erroneously a positive example ............................................................... 47

Figure 4.2. Algorithm for updating a Boolean function that accepts
erroneously a negative example ............................................................... 49

Figure 5.1. Set of documents to be analyzed .................................................... 52

Figure 6.1. Average classification accuracy using patterns of 1, 2, 3, 4,
and 5 consecutive words .......................................................................... 57

Figure 6.2. Comparison of the classifications under both algorithms ................. 61

Figure 6.3. Results when the GUIDED and RANDOM approaches were used
on the (DOE vs. ZIPF) class-pair ............................................................... 66

Figure 6.4. Results when the GUIDED and RANDOM approaches were used
on the (WSJ vs. ZIPF) class-pair ............................................................... 66

Figure 6.5. Results when the GUIDED and RANDOM approaches were used
on the (AP vs. DOE) class-pair ................................................................. 67

ix

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 6.6. Accuracy of the NILE and ILE approaches for the class-pair (DOE vs. ZIPF) ........................................ 71

Figure 6.7. Accuracy of the NILE and ILE approaches for the class-pair (WSJ vs. ZIPF) ........................................ 71

Figure 6.8. Accuracy of the NILE and ILE approaches for the class-pair (AP vs DOE) ........................................ 72

Figure 6.9. Number of clauses under the NILE and ILE approaches for the class-pair (DOE vs. ZIPF) ......................... 74

Figure 6.10. Number of clauses under the NILE and ILE approaches for the class-pair (WSJ vs. ZIPF) ......................... 75

Figure 6.11. Number of clauses under the NILE and ILE approaches for the class-pair (AP vs DOE) ......................... 75

Figure 6.12. CPU times under the NILE and ILE approaches for the class-pair (DOE vs. ZIPF) ............................... 78

Figure 6.13. CPU times under the NILE and ILE approaches for the class-pair (WSJ vs. ZIPF) ............................... 78

Figure 6.14. CPU times under the NILE and ILE approaches for the class-pair (AP vs DOE) ............................... 79
ABSTRACT

The main problem investigated in this dissertation is as follows: Given are two samples of documents each from one of two disjoint collections of documents. The question is how to obtain a set of patterns of text features that make a document in the two samples (and other unclassified documents) to be classified correctly in one and only one document class. A sample of 2,897 documents from the TIPSTER collection was used to investigate this problem.

This problem was divided into the following four subproblems. The first subproblem consists of identifying the set keywords to describe the documents' content. Computational results of twenty experiments suggested that single-word keywords addressed the main problem effectively.

The second subproblem requires a methodology to construct classification rules to infer the class of unclassified documents. A logical analysis approach called the One Clause At a Time algorithm (OCAT) is used to address this problem. Its accuracy is compared to the one of the Vector Space Model (VSM), a benchmarking methodology in document classification processes. Under identical experimental conditions, some computational results suggests that the OCAT algorithm is more accurate than the VSM to solve the main problem.

The third subproblem consists of providing a methodology to construct better rules as more documents become available. This problem has been investigated using the OCAT algorithm under a guided and a random learning approach. Computational results on three
samples of 510 documents indicate that the guided learning approach constructs more accurate rules.

In the fourth subproblem an incremental version of the OCAT algorithm is required. The algorithm is needed to speed up the construction of the classification rules. Computational results on three samples of 336 documents each show that: (i) the classification rules become accurate more rapidly, (ii) the CPU times are substantially reduced, and (iii) the rules become more complex as more documents were added to the experiment.

In summary, the results of this research suggest with high confidence that the incremental OCAT algorithm can perform better than the VSM and that it can deliver better and faster results for the classification of large collections of documents.
1 INTRODUCTION

The problem investigated in this research dealt with the classification of text documents into two disjoint classes. A typical application of this type of classification is in the declassification of documents in which a decision of whether or not to declassify a previously considered secret document is needed. Traditional classification methodologies can address this problem by grouping documents that share similar content. However, because the documents' content cannot be precisely defined, these methodologies may place a document in more than one class. For instance, a document titled "Studies in Mathematical Linguistics" may be placed in the classes: mathematics, linguistics, computer science, and natural language processing because its content is similar to the content of these classes. In contrast, the methodology investigated in this dissertation addressed this problem by classifying this document in one and only one class.

More precisely, this research investigated the following problem: Given are two samples of documents each from one of two disjoint classes. Each sample is assumed to belong to one class. The question is how to construct a set of patterns of text features that will make the documents in the samples (and other unclassified documents) to be classified correctly in one and only one document class.

This problem of classifying documents into two disjoint classes is new. It is new in the sense that a document must be placed in one and only one class. It is also new because now a classification decision may be triggered by the presence of a few or even just one special word. An example of this type of classification was recently raised in [DynMeridian, 1996]
and [DOE, 1996] who stated that "a document mis-classification may have grave consequences, such as affecting national security."

To illustrate the focus of this new classification problem, consider the following three sentences "An atomic test to be conducted at Site X," "An atomic test to be conducted at 1:00 p.m.,” and “An atomic test to be conducted at Site X at 1:00 p.m.” which came from three hypothetical documents, say, A, B, and C, respectively. According to the classification guidelines in [DynMeridian, 1996] and [DOE, 1996], only document C is both specific and sensitive, and therefore it will trigger a classification decision because it includes the place and time of the test. Documents A and B will not trigger such a classification because they are not specific.

It is interesting to notice that the utilization of traditional classification methodologies for the classification of the above three documents may be a misleading strategy. This can be illustrated by considering that these methodologies identify the content of a document based on the keywords that an expert classifier attaches to each document. For example, the content of the above three sentences can be described using the keywords “atomic,” “test,” and “Site X.” Thus, according to traditional methodologies all three documents will be grouped together because they share a similar content. However, according to the classification guidelines mentioned above, documents A and B belong to one group, while document C must belong to a second and disjoint group despite its content similarity with the previous two documents.

The methodology developed in this research solved this new classification problem by using a logical analysis algorithm called the One Clause At a Time (OCAT) algorithm.
[Triantaphyllou, 1994] and [Triantaphyllou and Soyster, 1996-1]. This algorithm takes as input binary valued statements (e.g., documents expressed as binary vectors) from two disjoint classes and constructs a set of logical rules. Then, it uses these rules to classify new statements (i.e., documents) into two mutually exclusive classes. An appealing feature of this methodology, and of other logical analysis methodologies, is that these rules can be conveniently expressed as IF-THEN type of rules.

The main motivation for using the OCAT algorithm to solve this new classification problem resided in the representation of text documents as binary vectors. According to Salton [1989], the management of actual text documents can be simplified by expressing them as surrogates. A surrogate is a list of keywords that is assigned to a document to represent its content. It is constructed on the assumption that the presence or absence of some keywords is a sufficient indicator of a document’s content (see, for example, [Meadow, 1992] and [Cleveland and Cleveland, 1983]). Thus, under this assumption the presence (denoted by 1) or absence (denoted by 0) of keyword $T_i$ in the binary vector $D_k$ may be used to represent the content of document $k$. The construction of two binary surrogates can be illustrated by considering the following list of twelve keywords and the following documents $A$ and $B$ (both were extracted from the Prodigy Service System on 03/12/95).

a) The Keywords: “astronauts,” “astronomers,” “budget,” “Clinton,” “Congress,” “Cynus Loop,” “explosion,” “Houston,” “NASA,” “scientist,” “space agency,” “supernova.”
That is, keywords $T_1 = \textit{"astronauts"}$, $T_2 = \textit{"astronomers"}$, $T_3 = \textit{"budget"}$, $T_4 = \textit{"Clinton"}$, 
..., $T_{12} = \textit{"supernova"}$.

b) The Documents:

Document A: “WASHINGTON, DC.—NASA is undergoing a complete makeover: slimming down, speeding up and becoming more youthful, the space agency administrator said Friday. Daniel Goldin said that National Aeronautics and Space Administration is under orders from the Clinton administration and from Congress to hold its budget to $13 billion by 2000 and the agency is trimming and finding new ways to operate in order to meet that goal. ‘We are meeting the mandate to cut our budget’ Goldin said at a briefing for reporters. ‘When we come out of this restructuring, the agency will be entirely different.’”

Document B: “SPACE CENTER, Houston, TX.—Like detectives piecing together clues to an ancient mystery, astronomers used Endeavor’s ultraviolet telescopes Saturday to gauge the effects of a star that exploded 20,000 years ago. Astronauts pointed that shuttle instruments toward a supernova known as the Cygnus Loop so scientists can figure out how shock waves from the long-ago explosion are affecting clouds of interstellar dust.”

When documents A and B are expressed as binary vectors, then the following two surrogates are formed: Document A = [001110001000] and Document B = [110001110101]. The binary information in the first surrogate indicates that the content of document A is defined by the presence of the keywords: $T_3 = \textit{"budget"}$, $T_4 = \textit{"Clinton"}$, $T_5 = \textit{"Congress"}$, and $T_6 = \textit{"NASA"}$ and by the absence of the keywords $T_1$, $T_2$, $T_6$, $T_7$, $T_8$, $T_{10}$, $T_{11}$, and $T_{12}$. Similarly, the second surrogate indicates that the content of document B is
defined by the presence of the keywords: $T_1 = \text{"astronauts"}$, $T_2 = \text{"astronomers"}$, $T_6 = \text{"Cynus Loop"}$, $T_7 = \text{"explosion"}$, $T_8 = \text{"Houston"}$, $T_{10} = \text{"scientist"}$, and $T_{12} = \text{"supernova"}$

and by the absence of the keywords $T_3$, $T_4$, $T_5$, and $T_9$.

**Figure 1.1.** An IF-THEN rule to verify the origin of documents $A$ and $B$.

As indicated earlier, the binary representation of these two documents was the key concept for selecting the OCAT algorithm to classify documents into two disjoint classes. To illustrate this concept, consider that it is given that documents $A$ and $B$ belong to classes $Science$ and $Politics$, respectively. In addition, consider that one is interested in deriving a set of relationships (or rules) to verify the origin of each document. Furthermore, suppose that the OCAT algorithm was used to construct the set of rules illustrated in Figure 1.1.

When these rules are used to process the surrogates $document A = [0011100010\ 00]$ and $document B = [110001110101]$, then the origin of each document can be found as follows. Because the surrogate for $document A$ includes only the words $T_j = \text{"budget"}$, $T_4 = \text{"Clinton"}$, $T_5 = \text{"Congress"}$, and $T_9 = \text{"NASA"}$ as indicated in the rule, this document is assigned to the class $Science$. On the contrary, $document B$ is assigned to class $Politics$ because it does not contain any of the keywords that form the rule. Of course this rule can be used to determine the class of other documents.

In order to measure the effectiveness of the OCAT algorithm for addressing this type of classification problems, its accuracy has been tested with a sample of almost 3,000
documents that were randomly selected from four classes of the TIPSTER collection. (The TIPSTER collection is a standard data set for experimentation with information retrieval systems [Harman, 1995] and [Voorhees, 1998].) Its accuracy was also compared with the classification performance of the Vector Space Model (VSM), which is often used as a benchmarking methodology for the classification of text documents [Salton, 1989].

It is important to mention here that other methodologies such as the fuzzy set approach (FSA), neural networks (NN), and computational semantic analysis (SA) can also be used to address the new document classification problem. However, the literature suggested that some promising approaches such as NN and SA are still limited to solve problems of small size, due to the time complexity of their algorithms or because the resulting sizes of their outputs are still unacceptable (see, for example, [Chen, 1996] and [Macleod, 1991]).

The remaining of this dissertation is organized as follows. In Section 2, the problem investigated in this research is formally presented. In Section 3, some relevant literature on the classification of documents is reviewed. In Section 4, the methodology for solving the new classification problem is described. Section 5 presents a numerical example that illustrates the utilization of the OCAT algorithm and the VSM model. In Section 6, some computational results of the four subproblems are presented and discussed. In Section 7, the contributions of this research are put forward. In Section 8, some suggestions for future research are described. Finally, this dissertation ends with a summary section.
2 PROBLEM DESCRIPTION

In this section, a formal definition of the main problem investigated in this research is presented.

2.1 Background

Recently, DynMeridian [1996] and DOE [1996] raised a new classification problem that states that "a document mis-classification may have grave consequences for national security." This problem is new in the sense that a document must belong to one and only one class. It is also new because now a classification decision can be triggered by the presence of few or even just one sensitive word that may occur in the text of the documents. These are two new requirements that are in contrast with the procedures of traditional classification methodologies, in which a classification decision is triggered by comparing the surrogate of a document against the surrogates of other documents.

At this point it should be stated that the U.S. Department of Energy (DOE) recognizes more than two classification classes. Such classes include categories such as: "Top Secret", "Secret", etc. However, in this research we considered only two classes. This is without loss of generality since any multiple class problem can be easily transferred into a sequence of two classes problem.

The importance of this new classification problem can be illustrated by considering the possible consequences of a mis-classification of the following two hypothetical documents titled "Test to be conducted at Site X on 3/15/98 at 1:00 p.m." and "Studies in Mathematical Linguistics." According to the guidelines of the above sources, the mis-classification of the first document may put at risk national security because it provides

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
specific information about the test. That is, the sentence provides the place, date, and the
time of the test. In contrast, the mis-classification of the second document may not have
such grave consequences. Therefore, to comply with the new classification requirements of
this new problem, a deterministic classification of a document in one and only one of two
possible classes was pursued.

2.2 Problem Definition

The main problem investigated in this research can be described as follows. Consider
that samples of documents from two disjoint classes are made available to a computerized
system with the indication of which class the documents came from. Also, consider that the
remaining documents in the two classes are temporarily hidden to the system. Next, suppose
that a hidden document becomes available without the indication of which class it came
from. Hence, the main problem investigated in this research work is best summarized as
follows:

Given are two samples from two mutually exclusive collections of documents. Each
sample is assumed to belong to one class. The question is how to obtain a set of patterns
of text features that make such documents (and other unclassified ones) to be classified in
one and only one document class.

An examination of this problem showed that actually the following four closely related
problems needed to be addressed.

Problem 1. This problem answered the question: How can a computerized system
identify the set of text descriptors that best reflect the type of information
contained in each document?
The expected result of investigating this problem was to be able to determine the set of keywords that best indexed the documents in the experiments.

Problem 2. This problem answered the question: *How can a computerized system construct a set of classification rules from samples of documents from two mutually exclusive classes?*

The expected result of investigating this problem was to be able to provide a system capable of constructing a set of classification rules by using the keywords that were used to index each document.

At this point, it can be argued that because these rules are constructed using only a relatively small sample of documents, it is quite possible that they can make mistakes when classifying unseen documents. Such mistakes would indicate that the rules do not possess enough knowledge to classify those documents, and therefore they would indicate that the rules need to be updated as more documents become available.

Thus, at this point the following problem needed to be addressed. Suppose that there is a vast number of documents to consider next during the initial training phase of the computerized system. (In fact, [DynMeridian, 1996] has indicated there may be hundreds of millions of such documents in various repositories.) Also, suppose that the actual class membership of each document is unknown. Therefore, the third problem investigated in this research was how to decide which document to consider next in the training session. This is a familiar problem in machine learning which is usually called the “Guided Learning” problem (see, for example, [Valiant, 1984], [Angluin, 1988], [Haussler and Warmuth, 1993], and [Holland, 1986]).
One strategy to answer this question is to randomly select the next document from the collection of the remaining unclassified documents. However, it is believed that if a guided approach is designed to select the next document, then: (i) better classification rules can be obtained and (ii) such rules will be constructed at a much faster rate than if a random approach is implemented. Thus, the third problem investigated in this research is best summarized as follows:

Problem 3. This problem answered the question: *How to design a "guided learning" approach for selecting the next document to classify during the training session.*

The expected result of studying this third problem was to be able provide a feedback system to the OCAT algorithm such that its classification accuracy can be improved as more documents become available.

Problem 4. This problem answered the question: *How to use an incremental learning methodology with the OCAT algorithm in order to improve its efficiency for the construction of new rules as more examples become available.*

The need for investigating this fourth problem was the large CPU times rendered in solving Problem 3. Furthermore, the main goal of investigating this problem was to provide a methodology to expand (also called repair) only the faulty term(s) of the classification rule that cause a mis-classification of a new document.
3 LITERATURE REVIEW

This section reviews the relevant literature associated with the four problems stated in the previous section.

3.1 Problem 1: Selection of Text Descriptors

According to [Cleveland and Cleveland, 1983], “the basic intellectual problem for constructing an index with a set of keywords is to accurately represent the content of a document with a dozen or so of terms.” Meadow [1991] also indicates that “the words in such an index must reflect the subjects of the documents from which they were selected.” The literature revealed the following four indexing methodologies for computerized information retrieval systems (manual systems were not reviewed in this dissertation work): the principle of least effort, the vector space approach, a genetic algorithm approach, and an all words approach.

3.1.1 The Principle of Least Effort

The Principle of Least Effort (PLE) is a two-step heuristic approach for constructing an indexing vocabulary with only the most meaningful words (or, keywords). It was introduced by Zipf [1949] and later enhanced by Luhn [1957-58]. The PLE works as follows. In the first step, it extracts a list of the words that cooccur in a sample of documents. It then divides this list into the following three categories: (i) words cooccurring infrequently (delimiting a lower cutoff), (ii) words cooccurring very frequently (delimiting an upper cutoff), and (iii) words with moderate cooccurring frequencies (or words in-between the two cutoffs). Zipf [1949] christened the words in the last category as the most

In the second step, only the words with moderate cooccurring frequencies are selected as the best set of keywords. According to [Zipf, 1949] these words convey the maximum lexical meaning. Other words such as the common words (i.e., those in the upper cutoff) and the rare words (i.e., those in the lower cutoff) are discarded for further consideration because their lexical contribution is minimum.

3.1.2 The Vector Space Model

Another method for selecting the best set of keywords is the Vector Space Model (VSM) ([Salton and Wong, 1975] and [Salton, 1989]). In contrast with the underlying principle of the PLE, the VSM selects these keywords by minimizing (maximizing) the following Equation (3.1):

$$F = \sum_{i \neq j}^{n} \sum_{i \neq j}^{n} \text{sim}(D_i, D_j).$$

The superscript $n$ corresponds to the number of words extracted from the documents in the collection, $\text{sim}(D_i, D_j)$ is a measure of the similarity (to be discussed in Section 3.2.1) between surrogates $D_i$ and $D_j$. The minimization of Equation (3.1) is used when the emphasis is in finding the set of words that will create clusters of various documents. Contrary, when the emphasis is in creating as many clusters as documents, then maximization is preferred.

3.1.3 A Genetic Algorithm Approach

In a more recent approach, Chen [1996] selected a set of keywords by using a Genetic Algorithm (GA) approach. In his experiments with the GANNET (Genetic Algorithms
Neural Networks) system, he first presented the GA with a set of thirty-three concept descriptors or keywords. Then, the GA created several populations of keywords that were used to index 1, 2, 3, 4, 5, and 10 documents. These documents were indexed using one population of keywords at a time. Finally, the GA selected as the close-to-optimal-solution, the population of keywords that minimized an average pairwise similarity (as measured by the Jaccard Coefficient in Table 3.1) across all documents in the experiments.

Table 3.1. Measures of Vector Similarity.

<table>
<thead>
<tr>
<th>Similarity Measure</th>
<th>Evaluation for Binary Term Vectors</th>
<th>Evaluation for Weighted Term Vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim(X, Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inner Product (IP)</td>
<td>(</td>
<td>X \cap Y</td>
</tr>
<tr>
<td>Dice Coefficient (DC)</td>
<td>(2 \frac{</td>
<td>X \cap Y</td>
</tr>
<tr>
<td>Cosine Coefficient (CC)</td>
<td>(\frac{</td>
<td>X \cap Y</td>
</tr>
<tr>
<td>Jaccard Coefficient (JC)</td>
<td>(\frac{</td>
<td>X \cap Y</td>
</tr>
</tbody>
</table>

\(X = (x_1, x_2, ..., x_t)\).
\(|X| = \text{number of terms in } X.\)
\(|X \cap Y| = \text{number of terms appearing jointly in } X \text{ and } Y\).
This table was taken from [Salton, 1989], Chapter 10.
3.1.4 An All Words Approach

In the problem raised by [DynMeridian, 1996] and [DOE, 1996], it was emphasized that human experts usually search for at least one occurrence of sensitive information in the documents’ text before they attempt any classification decision. (Sensitive information can be, for example, either one or more single words, names, numbers, dates, places, phrases, etc. that describe the specific conditions of an event.) Both references have also suggested that classification experts focus their attention on the type of information words may convey, rather than on the words’ cooccurring frequency. These new considerations suggest that in real classification situations an expert classifier must read at least all the words in a document before he/she attempts a classification decision. These two observations allowed us to hypothesize that in the worst case scenario, all the words that cooccur in a document text are needed for indexing purposes. We call this indexing approach, the All Words Approach (AWA), which we discussed in more details in Section 4.1.2.

3.1.5 Summary of Methodologies for Selecting a Set of Keywords

A common characteristic of the PLE, VSM, and GA indexing methodologies is the elimination of some words that cooccur in the text of a collection of documents. This elimination can be attributed either to heuristics or to optimal algorithms for selecting the best set of indexing terms. However, according to [DynMeridian, 1996] and [DOE, 1996] the elimination of these words may not be adequate for addressing the new classification problem because some of them may convey sensitive information.

An oversimplified example of the consequence of eliminating such words can be illustrated by considering the word “A.” According to [Meadow, 1991] and [Fox, 1990],
the grammatical article "A (or a)" is one of the most frequent words used in English writing. Moreover, both references also have suggested to discard the word "A" because its contribution to the lexical meaning of a document's content is minimum. However, if it is assumed that the content under consideration is on nutrition, then the elimination of the word "A" may be inadequate because it can be the name of the vitamin "A." In this research, the AWA approach was implemented as the indexing methodology in order to reduce the risk of discarding unnecessary words. This approach is detailed in Section 4.

3.1.6 Indexing of Documents

The goal of indexing a document is to guide the human efforts during the storing and retrieving of collections of document (see, for example, [Cleveland and Cleveland, 1983] and [Salton, 1989]). The index of a document is constructed by a list of preselected words that are believed to represent the document’s content. Usually, such an index is attached to the document. [Cleveland and Cleveland, 1983] and [Salton, 1989] refer to this list either as a "surrogate" or as a "document surrogate." An example of a surrogate to index this dissertation, which contains thousands of words, can be illustrated with the following five phrases "Document Classification," "Text Analysis," "Vector Space Model," "Neural Networks," and "Logical Analysis." Such an index can also be used to store (retrieve) this dissertation from some document category.

A surrogate can be further simplified by expressing it in one of two manners. First, it can be expressed as a binary vector to indicate the presence (denoted by 1) or absence (denoted by 0) of some keywords that cooccur in the text of a document. The vectors for documents \( A = [001110001000] \) and \( B = [110001110101] \) presented in Section 1 are two
examples of this type of surrogates, whose elements are referred to as the weight of a keyword.

In the second manner, a surrogate can be expressed as a weighted vector to indicate the relative importance of a keyword $T_i (i = 1, 2, 3, \ldots, t)$ in document $D_j (j = 1, 2, 3, \ldots, N)$. The value of this relative importance is a real number in the range $[0, 1]$, which is also referred to as the weight of a keyword. For practical purposes, the weight of keyword $T_i$ in document $D_j$, $w_{ij}$, is computed according to the inverted frequencies described in [Salton, 1989], as follows:

$$w_{ij} = f_{ij} \left( \log\left(\frac{N}{N_i}\right) - 1 \right), \text{ for } i = 1, 2, 3, \ldots, N \text{ and } j = 1, 2, 3, \ldots, t. \quad (3.2)$$

Where $f_{ij}$ is the cooccurring frequency of keyword $T_i$ in document $D_j$, $N$ is the number of documents in the collection, $N_i$ is the number of documents in which keyword $T_i$ occurs, and $t$ is the number of keywords.

Figure 3.1 shows a popular way to summarize a collection of $N$ surrogates [Salton, 1989]. In what follows, the words surrogate and document will be used interchangeably and will be denoted as $D_k (k = 1, 2, 3, \ldots, N)$. In order to comply with the requirements of the

![Figure 3.1. A collection of N documents and their surrogates.](image-url)
logical analysis algorithm that was used to solve the new classification problem, all surrogates in a collection will be expressed as binary vectors.

3.2 Problem 2: Classification of Documents

Traditionally, this type of classification is a process by which sets of documents sharing common characteristics are grouped together. The literature revealed manual and computerized or automatic processes for this type of classification processes. It also revealed that manual processes seem to deliver better results when the collection of documents is small [Jacobs, 1993]. On the other hand, [DynMeridian, 1996] has recommended the utilization of automatic and semiautomatic processes for large collections of documents in order to increase the efficiency of expert classifiers. The literature revealed the following four automatic methodologies for addressing the classification problem: fuzzy set methodologies, semantic analysis approaches, artificial intelligence approaches, and the vector space model. To comply with the literature, the vector space model [Salton, 1989] was used in this research as a benchmarking methodology for these types of classification processes.

3.2.1 A Fuzzy Set Approach

This approach is based on the Fuzzy Set Theory (FST) that was introduced by Zadeh [1965]. Zadeh indicated that “the natural language humans use to express their thoughts is often vaguely or imprecisely defined.” The phrase “a tall individual” is an example of a concept that is imprecisely defined because depending on the standard used, an individual can be either tall or short. For instance, if the height 6' 4" is defined as the standard to
determine the tallness of an individual, then all individuals below this height are said to be short. Otherwise, they are said to be tall.

However, under Zadeh's observations an individual can be assigned a degree of tallness with respect to a preestablished standard. For example, if this standard is defined by the set of heights in the range 5' 6" and 6' 4", then a 5' 9" individual will be 0.91 (5' 9" ÷ 6' 4") tall with respect to the upper limit of this range. Similarly, a 6' 3" individual will be 0.98 (6' 3" ÷ 6' 4") tall. In the same vein, if the height of an individual is either 3' 4" or 7' 5", then he/she is said to be 0.0 tall because his/her height is off the range that defines tallness.

FST has also been extended to applications for storing and retrieving documents to and from different classes (see, for example, [Kraft et al., 1994], [Bordogna and Passi, 1995-2], [Lee et al., 1992], and [Molinary and Passi, 1996]). For the classification of documents (i.e., storing), the FST is usually a two-step approach, as follows. In the first step, it takes as input a surrogate and determines its membership in the various classes. The degree of membership a document has in a class is usually determined by one of the following tags: "completely," "strongly," "moderately," "little," or "nothing" (see, for example, [Molinary and Passi, 1996]). The cosine coefficient (CC) as shown in following Table 3.1 can be used to determine these degrees of membership.

Thus, if the TFS determines that a document belongs "completely" to a class, its membership is set to 1. On the contrary, if the document does not belong (i.e., "nothing") to a class, its membership is 0. Other membership such as "strongly," "moderately," and "little" indicate a partial membership and its membership value is defined by a real number between 0 and 1.
In the second step a document is assigned to a class by comparing its membership values with a predetermined threshold value (usually called the \( \alpha \)-value). To address the new classification problem, the FST may proceed as follows. A document may be assigned to the first class if, for example, its membership value is lesser than or equal to the \( \alpha \)-value. Otherwise, the document will be assigned to the second class.

A drawback of FST approaches is that if the \( \alpha \)-value is decreased, then it is likely that the document will be placed, say, in the second class. Contrary, if this value is increased, then the document may still belong in the first class. This situation clearly violates the deterministic classification condition of the new problem.

3.2.2 A Semantic Analysis Approach

Another methodology for the classification of documents into two disjoint classes is the Semantic Analysis (SA) approach. SA can address the new classification problem by semantically discovering the underlying content of a document and then by comparing it to some pre-established knowledge, which usually is stored in frames. A frame is a cluster of facts, objects, and strategies which are used to describe some typical formulation for reasoning about a given situation [Allen, 1985]. A typical frame may contain definitions of the words to be used, the structure of the sentences to be analyzed, and some world knowledge. SA approaches discover the content of a document by extracting the semantic meaning of sets of non-blank characters from its text (see, for example, [Rau and Jacobs, 1991], [Jacobs, 1991], [Pesole et al., 1994], [Croft, 1993], and [Korfhage and Olsen, 1994]).
According to Kim and Moldovan [1995], Korfhage and Olsen [1994], and Rau and Jacobs [1991], the chief limitation of SA approaches for the classification of documents is the extremely large number of frames that are needed in order to determine the content of each document. Moreover, the above two sources have also indicated that (i) SA can only be used for specific and narrow domains and (ii) field experts reckon this methodology as unpractical, very costly, and prone to errors (see, for example, [Kim and Moldovan, 1995]) because frames are man-made and their construction and maintenance can become an engineering bottleneck.

3.2.3 A Neural Networks Approach

A promising approach to solve the classification of documents problem is the utilization of Neural Networks (NN) [Chen, 1996]. NNs are sets of computational procedures inspired by the neurobiological systems found in the brain of various animals (see for example, [Jain et al., 1994], [McCulloch and Pitts, 1943], [Aoe, 1990], and [Haykin, 1994]). According to [Rumelhart et al., 1994], “the strategy with NNs is to secure some mathematical models of brain-like systems to study and understand how various problems can be solved using such models.”

The main problem with an NN approach is how to train the network, so that it can perform efficiently a specific task (e.g., document classification) [Rumelhart et al., 1994] and [Jain et al., 1996]. This training is acquired by determining the numerical values of the weights of the connections between neurons and the numerical values of the thresholds of each neuron. These values are determined by repeatedly presenting the NN with sets of pre-classified examples (i.e., surrogates in this study) of the task to be learned, and by adjusting
the weights and threshold values until the network performs the specific task accurately. Some applications that illustrate the utilization of NN for the classification of text documents of small collections can be found in Chen [1996], Chen et al. [1994], and Macleod [1991].

Although the literature indicated that NNs may be promising methodologies for addressing the new classification problem, Scholtes [1993] has suggested that they are still in an experimentation phase mainly because “the resulting NNs are of an unacceptable size.” Moreover, researches in fields different from text classification have indicated that defining the weights in an NN could be easily understood by the person designing the network but not for the application’s field expert who is often the end user. The literature also suggested that experts (such as medical doctors) sometimes may feel uncomfortable in accepting the recommendations made by a NN due to the lack of justification capabilities.

3.2.4 A Logical Analysis Approach

Another promising method to solve the new classification problem is the One Clause At Time (OCAT) algorithm [Triantaphyllou et al., 1994] and [Triantaphyllou, 1994]. The OCAT algorithm (Figure 3.2) is a logical analysis approach for the classification of examples into two disjoint classes. This algorithm takes as input a sample of training examples from two disjoint classes and constructs a set of logical rules that can be used to verify the class of the training examples (binary surrogates in our case) and to infer the class of other unseen examples. Also, it can express these rules in the form of a Boolean function either in CNF (Conjunctive Normal Form) or in DNF (Disjunctive Normal Form) form.
The OCAT algorithm is greedy in nature. It is greedy in the sense that in the first iteration, it forms a clause that for the CNF case accepts all the examples in one of the classes (often referred to as $E^+$) and rejects as many examples as possible in the other class (often referred to as $E^-$). Similarly, in the second iteration it forms another clause that accepts again all the examples in $E^+$ and rejects as many examples, that were not rejected by the previous clause(s), in $E^-$ as possible. The algorithm repeats this process until the generated sequence of clauses reject all the examples in $E^-$. 

The core of this algorithm is Step 2 in Figure 3.2. In [Triantaphyllou, 1994] and [Triantaphyllou and Soyster, 1996a and 1996b] a branch-and-bound (B&B) approach to solve the task in Step 2 was implemented. The goal of this B&B strategy was to construct a set of clauses of minimal or close to minimal size. This B&B approach has been used for solving small problems mainly because of the very long CPU times it takes to find the optimal set of clauses for larger problems.

| Input: | Two mutually exclusive sets of examples (denoted as $E^+$ and $E^-$). |
| Output: | A Boolean function in CNF (or DNF) form given as $C$. |

```
begin
  i = 0; C = ø;
  do while ($E^- ≠ ø$)
    Step 1: $i = i + 1$; /* $i$ indicates the $i$th iteration */
    Step 2: find a clause $c_i$ which accepts all members of $E^+$ while it
             rejects as many members of $E^-$ as possible;
    Step 3: let $E^-(c_i)$ be the set of the members of $E^-$ which are rejected by $c_i$;
    Step 4: let $C = C ∪ c_i$;
    Step 5: let $E^- = E^- \setminus E^-(c_i)$;
    repeat;
  end;
```

Figure 3.2. The One Clause at a Time (OCAT) algorithm for the CNF case.
Recently, [Deshpande and Triantaphyllou, 1998] implemented a heuristic approach of polynomial time for Step 2. This heuristic allows for the solution of larger classification problems and delivers a set of clauses of small size (as opposed to the minimal or near minimal size that can be delivered by the B&B). This heuristic is illustrated in the following Figure 3.3.

A common feature of the above two algorithms is that they produce only one Boolean function. As indicated in the above references, such a function can only accept or reject an input; $E$ and $E'$. Output: A Boolean function either in the CNF or DNF form.

\[
\begin{align*}
q &= 1; \\
&\text{do while } (E' \neq \ø) \\
&\quad \text{Let } E^+ \text{ be the original set of positive examples;} \\
&\qquad K_q = \ø; \quad \text{initializing a clause} \\
&\quad \text{do while } (E^+ \neq \ø) \\
&\text{Step 1: Calculate the } POS(a_j) / NEG(a_j) \text{ ratio for all atoms } a_j \in a \text{ (where } a_j \text{ is either } A_j \text{ or } A^c_j). \text{ If } NEG(a_j) = 0 \text{ set this value to } POS(a_j) \times 1000 \text{ (a high value). Also, if } POS(a_j) = 0 \text{ and } NEG(a_j) = 0 \text{ set the value to -1;} \\
&\text{Step 2: Choose the } a_j \text{ according to } \max\{POS(a_j) / NEG(a_j)\}. \text{ Break ties arbitrarily;} \\
&\text{Step 3: Let } K_q = K_q \cup a_j; \\
&\text{Step 4: Let } E^+(a_j) \text{ be the set of members of } E^+ \text{ which are accepted when } a_j \text{ is included in the current clause } K_q; \\
&\text{Step 5: Let } E^+ = E^+ - E^+(a_j); \\
&\quad \text{repeat;} \\
&\text{Step 6: Let } E^-(K_q) \text{ be the set of members of } E^- \text{ which are rejected by } K_q; \\
&\text{Step 7: Let } E^- = E^- - E^-(K_q); \\
&\quad q = q + 1; \\
&\quad \text{repeat;}
\end{align*}
\]

Figure 3.3. The heuristic combined with the OCAT approach [Deshpande and Triantaphyllou, 1998].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
example. However, because such a function is usually constructed from a relatively small collection of examples, it is possible that it can make an error by rejecting a true positive example or by accepting a true negative example. This may be a disadvantageous situation for these two algorithms because they cannot detect such errors.

On the other hand, the methodology implemented in this research is a novel approach that combines the decision of two complementary rules to infer and to validate the class of new examples. The underlying assumption of this novel approach can be understood more easily by considering the implementation of the OCAT algorithm on the data in Figures 3.4 and 3.5.

From the examples in Figure 3.4 it can be observed that if OCAT is implemented on these two sets, then it will create a Boolean function (either in the CNF or DNF form) that evaluates to true for each of the positive examples and to false for each of the negative examples. We call this Boolean function the positive rule because it only accepts positive examples and rejects the negative ones.

The Figure 3.5 shows again the previous ten examples, but this time the data in the two sets have been interchanged. That is, now the new class $E^+$ consists of the former

$$E^+ = \begin{bmatrix}
0100 \\
1100 \\
0011 \\
1001
\end{bmatrix}$$

$$E^- = \begin{bmatrix}
1010 \\
0001 \\
1111 \\
0000 \\
1000 \\
1110
\end{bmatrix}$$

Figure 3.4. A set of four positive examples and a set of six negative examples.
negative examples while the new class \( E^* \) consists of the former positive examples. With this new setting of examples, it can be easily seen that this time the OCAT algorithm will construct a Boolean function that evaluates to true for all former negative examples and evaluates to false for all former positive examples. We call this Boolean function the negative rules because this time it accepts the former negative examples, while it rejects all the former positive examples.

The advantage of using two rules to classify an example can be illustrated as follows. Because a Boolean function can only accept or reject an example, it is expected that if the positive rules accept(reject) the example, then the negative rules will reject(accept) it. These

\[
E^* = \begin{bmatrix}
1010 \\
0001 \\
1111 \\
0000 \\
1000 \\
1110
\end{bmatrix} \quad
E^- = \begin{bmatrix}
0100 \\
1100 \\
0011 \\
1001
\end{bmatrix}
\]

Figure 3.5. A set of six positive examples and a set of four negative examples.

observations follow from the setting of the examples in Figures 3.4 and 3.5.

However, because both rules are constructed using a limited number of training examples, it may be possible that both rules will simultaneously accept or reject an unseen example. This would be an erroneous result that contradicts the expectations of both rules as they were described above. More importantly, such an erroneous result indicates that only one of the rules does not possess enough knowledge to classify the unseen example. We called this an "Undecided" situation and we used it to detect when the two rules could
not classify correctly an unseen example. In this research, we turned an "Undecided" result into an advantageous situation that was used to guide (i.e., reconstruct) the learning of the rule that triggered the unexpected decision.

3.2.5 A Vector Space Model Approach

The Vector Space Model (VSM) is a traditional methodology for the classification of text documents [Salton and Wong, 1975] and [Salton, 1989]. The VSM model works as follows. First, it takes as input a sample of surrogates to construct clusters of documents that share a similar content. Then, it uses the information in each cluster to classify new documents. Figure 3.6 illustrates the typical three steps of the VSM model.

| Input: A sample of document surrogates |
| Output: Clusters of documents and clusters' centroids |
| **Step 1:** Compute a pairwise similarity coefficient among all surrogates in the sample; |
| **Step 2:** Cluster documents that share similar context (i.e. similar coefficient); |
| **Step 3:** Compute centroids of clusters; |

Figure 3.6. The Vector Space Model (VSM) algorithm.

To address **Step 1** Salton [1989] indicates that a suitable measure for pairwise comparing any two surrogates is the Cosine Coefficient (CC). (The CC and other similarity measures are listed in Table 3.1.) This coefficient measures the geometric angle between two surrogates in a Cartesian plane. Salton also indicates that "the magnitude of this angle can be used to measure the similarity between any two documents." He also suggests that the smaller this angle is, the more similar the two documents are.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In Step 2, the VSM clusters together documents that share a similar content. According to Salton [1989], any clustering technique can be used to group documents with alike surrogates. A collection of clustering techniques for this purpose is given in [Anderberg, 1973], [Van Rijsbergen, 1979], [Aldenderfer, 1984], and [Späth, 1985]. It is important to mention here that under any of these techniques the number of generated classes is always a function of some predefined parameters. This is a contrasting situation with the requirements of the new classification problem in which the number of classes is given. For instance, in this research the number of classes were two. When the number of classes is give, the VSM is said to perform a pseudo-classification.

To address Step 3 [Salton, 1989], [Salton and Wong, 1975], and [Van Rijsbergen, 1979] suggest the computation of a class centroid to be done as follows. Let $w_q \ (j = 1, 2, 3, \ldots, t)$ be the $y$th element of the centroid for class $C_r$ which contains $q$ documents. Then, $w_q$ can be computed as follows:

$$w_q = (1/q) \sum_{i=1}^{q} D_{ij}, \quad \text{for } j = 1, 2, 3, \ldots, t. \quad (3.3)$$

That is, the centroid of class $C_r$ is also a surrogate defined on $t$ keywords.

Finally, the VSM classifies an unseen document by comparing its surrogate against the centroids that were created in Step 3. The CC coefficient can also be used to measure the similarity between a new document and the centroids. Hence, a document will be placed in the class for which the CC index is maximum. If two surrogates are identical, then the angle between them is 0°, and therefore the cosine index is maximum.

The data in Figures 3.4 and 3.5 also show two typical inputs to the VSM model. It can be observed that if the VSM is implemented on these data, then one pair of centroids
will be created for the data in Figure 3.4 and one pair for the data in Figure 3.5. The construction of these two pairs of centroids corresponds to the construction of the positive and negative rules described in Section 3.2.4 for the OCAT algorithm. Moreover, to match the name of the rules, the two centroids for the data in Figure 3.4 were called the positive centroids while the centroids for the data in Figure 3.5 were called the negative centroids.

As with the OCAT algorithm, these two pairs of centroids can also be used to improve the classification accuracy of the VSM as new examples become available. This can be done as follows. First, it is expected that if the positive centroids classify a new document as positive(negative), then the negative centroids will classify it as negative(positive). However, it can also be possible that the two pairs of centroids will erroneously classify the same example as either positive or as negative. As indicated in the previous section, this is an “Undecided” situation that indicates that at least one of the centroids does not possess enough knowledge to classify new documents. More importantly, this situation suggests that either the positive or the negative centroids must be updated in order to improve the accuracy of the VSM algorithm.

3.2.6 Summary of Classification Methodologies

The literature suggested that the main disadvantage of FST approaches is that they may not comply with the deterministic requirements of the new classification problem stated in [DynMeridian, 1996] and [DOE, 1996]. The literature also indicated that although SA approaches may be used to solve the new classification problem, their use is still limited to only small problems mainly because their construction and maintenance are always expensive and highly prone to errors. The literature further suggested that although NNs are
promising approaches to address the new classification problem, their resulting size may be still unacceptable for real text classification problems.

On the other hand, the OCAT algorithm appeared to be the best methodology for addressing the new problem because it can deliver a deterministic classification at reasonable CPU times. But most importantly, this algorithm was selected to address the classification of text documents into two disjoint classes because it has been used successfully in other research areas such as cancer detection (see, for example, [Kovalerchuk et al., 1997]), in which a wrong decision can also have severe consequences. Finally, the VSM was selected to benchmark the classification performance of the OCAT algorithm. It is important to mention here that this is the first time that positive and negative rules and positive and negative centroids are implemented and studied on the OCAT and VSM algorithms, respectively.

3.3 Problem 3: The Guided Learning Approach

The central idea of the Guided Learning Approach (GLA) can be illustrated as follows. Suppose that the collection to be classified contains millions of documents. Also, consider that initially only a small sample of documents from this collection is given to an expert classifier for its correct classification into class $E^+$ or $E^-$. Moreover, suppose that the OCAT algorithm is used to construct positive and negative rules by using these two classes. As mentioned in Section 3.2.4, these rules will only be accurate when classifying the examples in the samples and may be inaccurate when classifying other unseen examples.

The classification accuracy of these rules can be improved by reconstructing them when more training examples are included in $E^+$ and $E^-$. These example can be added to
these sets as follows. First, one can provide an expert classifier with an unclassified document and ask him/her to determine its correct classification. Next, one can place this document in either $E^+$ or $E^-$ according to the expert's recommendation. Then, we can reconstruct the positive and negative rules using the new $E^+$ and $E^-$ sets. This process can be repeated until the rules have achieved the desired accuracy. Therefore, the question GLA attempts to answer is: 

**What is the next document to be inspected by the expert classifier so that the performance of the positive and negative rules can be improved as fast as possible?**

One way to provide the expert with this document is to randomly select one from the remaining unclassified documents. We called this the RANDOM learning strategy. A drawback of this strategy may occur if the classification verdict of the *incumbent* positive and negative rules frequently coincides with the expert's verdict. Under this scenario, the reconstruction of the rules may be inefficient because they may not improve fast enough.

An alternate and more efficient way is to provide the expert with a document that the two incumbent rules already identified as an "Undecided" situation. This appears to be a more efficient way of selecting the document because when an "Undecided" situation occurs, the expert's verdict will always guide the reconstruction of the rule that triggered the mis-classification. We called this the GUIDED learning strategy.

The advantage of using a GUIDED learning strategy over a RANDOM strategy to improve the learning rate of the OCAT algorithm can be hypothesized as follows. These two rules can only have one of three possible classification outcomes: *Correct, Incorrect*, or *Undecided*. The conditions to obtain these three outcomes are illustrated in Figure 3.7.
Furthermore, as indicated in Section 3.2.4 the detection of an "Undecided" situation will always lead to the reconstruction of the faulty rule. Therefore, when an oracle (i.e., an expert classifier) is asked to provide the correct classification of an example that was already identified with an “Undecided” classification by the two incumbent rules, the faulty rule will always be identified and reconstructed. In contrast, the reconstruction of either rule under the RANDOM strategy may be unnecessary if the example to be inspected by the oracle is correctly classified by the two incumbent rules.

The classification of a new example \( D \) is correct if and only if:

- \( D \) is a positive example, and the positive rule accepts it, while the negative rule rejects it.
  Or,
- \( D \) is a negative example, and the positive rule rejects it, while the negative rule accepts it.

The classification of a new example \( D \) is incorrect if and only if:

- \( D \) is a positive example, and the positive rule rejects it, while the negative rule accepts it.
  Or,
- \( D \) is a negative example, and the positive rule accepts it, while the negative rule rejects it.

The classification of a new example \( D \) is undecided if and only if:

- \( D \) is accepted by both the positive and the negative rules.
  Or,
- \( D \) is rejected by both the positive and the negative rules.

These criteria can be used with OCAT or any other classification algorithm.

Figure 3.7. Possible outcomes when the classification decisions of the positive and negative rules are considered simultaneously.
3.4 Problem 4: Incremental Learning with the OCAT Algorithm

The OCAT algorithm, as described in Figure 3.2, indicates that a Boolean function is reconstructed every time a new set of positive and negative examples becomes available. That is, if the available sets of examples $E^+$ and $E^-$ are given, then a Boolean function $\mathcal{F}$ will be obtained. Similarly, if a newer set of examples is composed by $E^+ \cup e^+$ (where $e^+$ is a single positive example) and $E^-$ is the set with the negative examples, then the OCAT algorithm will reconstruct a new Boolean function, say $\mathcal{F}'$. In many applications reconstructing a function in this way may be computationally expensive even if only a single example is added to the previous training sets (see, for example, [Hunt et al., 1966], [Michalski, 1978], [Michalski, 1985], [Reinke and Michalski, 1986], [Schlimmer and Fisher, 1986], [Utgoff, 1989], [Utgoff, 1997]). The literature refers to this type of learning as non-incremental learning (NILE) because it reconstructs the existing knowledge (i.e., the Boolean function) every time a new problem is presented.

On the other hand, incremental learning from examples (ILE) may be an attractive strategy to update such a function because only portions of it are repaired as new examples are observed. Among the first works addressing ILE was the Concept Learning System (CLS) [Hunt et al., 1966]. In CLS, prior observations were selected at random and were replaced with new examples in order to reconstruct the new knowledge. CLS was soon abandoned because the learning rates were slow. In Michalski and Larson, [1978], the AQ ([Michalski, 1973]) systems was adapted to learn incrementally by limiting the number of examples needed to reconstruct the faulty knowledge, which was expressed in "plain" DNF form. The AQ system repaired this knowledge by using a Euclidean distance measure to
identify new examples that were “good” concept representatives. Its goal was to reconstruct only those portions of the knowledge (a set of clauses which describes an individual concept) that caused the mis-classification.

Later, [Reinke and Michalski, 1986] extended the AQ system into the GEM system which repairs only individual terms of DNF expressions. In the GEM system only the faulty conjunctive terms were submitted to a generalization procedure along with the observations it currently covered and those that triggered the classification inconsistency. The results of this system suggested that (i) ILE methods yield more complex concept descriptions than NILE methods and (ii) knowledge base updates are less expensive using ILE than with NILE methods.

A common underlying assumption of the above ILE methods is that they insist in perfect consistency between the existing knowledge and the real world of observations. That is, they do not allow for noise during the training process. (Noise is defined as observations that were incorrectly described due to a faulty perception.) Alternatively, incremental learning systems such as STAGGER ([Schlimmer, 1987]), ID3 (Quinlan, 1986), ID4 (Schlimmer and Fisher, 1986), and ID5 ([Utgoff, 1989]) allow for some noise between the existing knowledge and the observations during the training process. In these systems, noise is allowed because knowledge is represented in a probabilistic manner, and according to Schlimmer [1987], “this probabilistic knowledge representation enables a system to discern and respond to long-term environmental changes.” Other works addressing a probabilistic knowledge representation are described in [Hampson and Kibler, 1983] and [Langley,
1987]. ILE methodologies that allow for some noise are not further discussed in this research.

The incremental learning methodology implemented on the OCAT algorithm (to be described in Section 4.4) is similar to the one of the GEM system. It is similar in the sense that it does not allow for noise between the knowledge and the observations and because it repairs only the disjunctive terms that caused a mis-classification. Nonetheless, it differs from the GEM system in the way the new training examples are selected and used to reconstruct the current knowledge. For example, in the GEM system knowledge is submitted to a generalization process only when a set of mis-classified examples has been collected. In contrast, in the methodology implemented in this work this knowledge is repaired by only considering examples with an “Undecided” situation. Furthermore, the methodology presented here differs from that of the GEM system because it attempts to maintain two rules of small size.
4 METHODOLOGY

The methodology to investigate the four problems stated in Section 2 is presented next.

4.1 Methodology for Problem 1: Selection of the Best Content Descriptors

In order to investigate this problem, the following four tasks were implemented: (i) selection of the training documents, (ii) indexing of the documents, (iii) construction of a matrix of binary surrogates, and (iv) definition of the experimental conditions.

4.1.1 Selection of a Sample of Training Documents

A sample of documents from four classes of the TIPSTER collection (see, for example, [Harman, 1995] and [Voorhees, 1998]) was randomly selected to investigate the performance of the OCAT and the VSM algorithms. (The TIPSTER collection is distributed in three compact disks that contain tens of thousands of text documents for experimentation with information retrieval systems.) Table 4.1 shows the numbers of documents from each class that were used to investigate all four research problems.

<table>
<thead>
<tr>
<th>Class:</th>
<th>DOE</th>
<th>AP</th>
<th>WSJ</th>
<th>ZIPF</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Documents:</td>
<td>1,407</td>
<td>336</td>
<td>624</td>
<td>530</td>
<td>2,897</td>
</tr>
</tbody>
</table>

DOE, AP, and WSJ stand for Department of Energy, Associated Press, Wall Street Journal, respectively. ZIPF is a collection of technical documents on various topics.

Additionally, to simulate two disjoint classes, the following five class-pairs were formed: (DOE vs. AP), (DOE vs. WSJ), (DOE vs. ZIPF), (AP vs. WSJ), and (WSJ vs. ZIPF). These five class-pairs were randomly selected from all possible pair combinations. Furthermore, to comply with the notation presented in the previous sections, the first and second classes of each class-pair was denoted as \( E^+ \) and \( E^- \), respectively.
4.1.2 Methodology for Indexing the Documents

To index the documents in each experiment, two dictionaries were formed. The first dictionary was composed of all the unique words cooccurring in the documents. In Section 3.1.4 we referred to this list as the AWA. This dictionary was needed in order to avoid duplicated words. The second dictionary was constructed using unique patterns of consecutive words; all words in these patterns were validated against the first dictionary. Finally, all documents were indexed using the patterns from the second dictionary. (From now on these patterns will also be referred to as keywords or content descriptors). This methodology is presented next.

4.1.2.1 Construction of a Dictionary of Unique Words

The construction of the dictionary of unique words followed the guidelines in [Hsiao and Haray, 1970], [Glaserfeld and Pisani, 1970], and [Rau and Jacobs, 1991] for inverted files. In this research, however, the “look-up table” proposed in these guidelines was organized according to the prefixes “a, ab, ac, ..., az, b, ba, be, ..., z, za, ..., zu”, as they are found in regular dictionaries. This table also accommodated prefixes such as “1a, 1b, ..., 9z”, “a1, a2, ..., z9”, and “00, 01, 02, ..., 99” that often appear in the technical literature. One dictionary was constructed for each experiment.

The construction of this dictionary was done as follows. At first, a sentence was extracted from each document, and its words were pre-processed to determine if they were:

1. English words
2. Numbers, e.g., 3.1416
3. Abbreviations, such as Mr., Miss., U.S.A., etc.
4. Mixed words, such as MX14, 5/11/97, etc.
5. Stop words, e.g., a, an, the, for, is, etc.
Then, each word was converted to lowercase in order to avoid duplications (see, for example, [Cleveland and Cleveland, 1983]), and any punctuation such as "", ",", ".", ";", ";!", ";?" was removed from it. The punctuation of numbers such as -3416, 5%, or 5E5 was retained.

4.1.2 Construction of a Dictionary of Unique Patterns of Words

To construct this dictionary of pattern of words, one pre-processed sentence was considered at a time (see, for example, [Jacobs, 1993], [Chen et al., 1994], [Abramova et al., 1981], and [Belonogov et al., 1989]). To illustrate the construction of these patterns, consider the following sentence with the three consecutive words "louisiana state university" (in lowercase). From these three words, the following six patterns $T_i$ (for $i = 1, 2, 3, ..., 6$) can be generated: $T_1 = \text{"louisiana"}$, $T_2 = \text{"state"}$, $T_3 = \text{"university"}$, $T_4 = \text{"louisiana state"}$, $T_5 = \text{"state university"}$, and $T_6 = \text{"louisiana state university."}$

Then, these patterns of words were converted into numbers as follows in order to speed up the searching times in this dictionary. This conversion was done as follows. Suppose that the unique words (all in lowercase) "louisiana", "state", and "university" were hypothetically located in rows 5, 125, and 79, respectively in the dictionary of unique words. In this case, pattern $T_1 = \text{"louisiana"}$ was converted into $T_1 = [5]$. As another example the pattern $T_6 = \text{"louisiana state university"}$ was converted into $T_6 = [5, 125, 79]$. These numerical patterns were recorded in the dictionary of unique content descriptors in this numeric format. As another illustrative example, the keyword $T_8 = \text{"state of louisiana"}$ which came from a different sentence was converted into $T_8 = [125, 5]$ when the above hypothetical row locations are also considered. (Observe that the word "of" was
intentionally left out because according to [Fox, 1990] the preposition "of" is a stop word.)

This process was repeated until all the sentences in a document were processed.

4.1.2.3 Indexing of the Documents

A document $D_i$ (where $1 \leq i \leq N$, and $N$ is the number of documents in the experiment) was indexed with the patterns that only cooccurred in its text. This indexing process was accomplished as follows. Suppose that pattern $T_a = "louisiana state university"$ (or, equivalently, $T_a = [5, 125, 79]$) occurred 20 times in document $D_a$. Furthermore, suppose that $T_a$ was located in row 6 in the dictionary of unique content descriptors. In this case, the presence of $T_a$ in $D_a$ was identified as $[6, 20]$ to denote that $T_a$ occurred 20 times in $D_a$. As another example, the pattern $T_a = "state louisiana"$ (i.e., $T_a = [125, 5]$) also occurs in $D_a$ only one time. In this case, the presence of $T_a$ in $D_a$ was defined as $[8, 1]$.

4.1.3 Construction of the Matrix of Binary Surrogates

The third step in the methodology for indexing documents consisted of creating a matrix of binary surrogates from the data in the dictionary of unique content descriptors. One matrix was constructed for each class in an experiment. The construction of these surrogates followed the guidelines provided in Section 4.2.1 (see, also, [Salton, 1989]). That is, in this case the binary weight $w_{ij}$ of keyword $T_j$ in surrogate $D_i$ was determined according to the following rule:

$$w_{ij} = \begin{cases} 
1 & \text{if the frequency of keyword } T_j \text{ in } D_i \text{ was } > 0, \\
0 & \text{otherwise.}
\end{cases} \quad (4.1)$$
An application of this rule can be illustrated by considering the patterns $T_6$ and $T_8$ in $D_a$ as defined in Section 4.1.2.3. (The frequency of other patterns is assumed to be zero).

In this case, the data in the dictionary of unique content descriptors for this document is as follows:

- [1,0] for $T_1$
- [2,0] for $T_2$
- [3,0] for $T_3$
- [4,0] for $T_4$
- [5,0] for $T_5$
- [6,20] for $T_6$
- [7,0] for $T_7$
- [8,1] for $T_8$

As a result, the above rule evaluates to 0 for the patterns $T_1$, $T_2$, $T_3$, $T_4$, $T_5$, and $T_7$, and it evaluates to 1 for patterns $T_6$ and $T_8$. Moreover, when these evaluations are collected in a vector, then the surrogate for document $D_a$ is defined as $[0 0 0 0 0 1 0 1]$.

**4.1.4 Experimental Conditions for Problem 1: Extraction of Word Patterns**

As indicated in Section 2, Problem 1 consists of identifying the best set of patterns of words to address the new classification problem. To determine these patterns, the classification accuracy of the OCAT and the VSM algorithms was studied under a leave-one-out cross validation (CV) on 20 experiments, each with 40 documents (20 from each class-pair). The numbers of experiments and documents to address Problem 1 were selected on pure intuition but the final results were tested for statistical significance.

The goal of implementing a CV approach on these experiments was to investigate the effect of using patterns of up 5 consecutive words on the classification accuracy of both algorithms. A maximum of 5 consecutive words was selected in order to avoid excessively long lists of words. In some preliminary experiments with the same 40 documents and patterns of up to 7 consecutive words, this list was comprised of more than 30,000
keywords. For this experimentation, classification accuracy for the OCAT case was defined as the proportion of documents accepted by only one rule, while for the VSM case it was defined as the proportion of documents accepted by only one pair of centroids.

4.2 Methodology for Problem 2: Classification Using Text Patterns

The methodology for the second problem consisted of the following three tasks: (i) construction of the positive and negative rules, (ii) construction of the positive and negative centroids, and (iii) definition of the experimental conditions.

4.2.1 Construction of the Classification Rules by Using the OCAT Algorithm

The construction of positive and negative rules in CNF form was achieved by using the heuristic in [Deshpande and Triantaphyllou, 1998]. Figure 3.3 (on page 26) illustrates this heuristic. Under this heuristic, a CNF expression was constructed as follows:

1. A clause was formed such that it accepted all the examples in the $E^+$ set, while it attempted to reject many (as opposed to reject as many as possible required by the original B&B approaches) examples in $E^-$. A single CNF clause was completed when all the examples in $E^+$ were accepted.

2. The atoms in each clause were obtained according to an evaluative function, and only the atoms with the highest value were included in that clause. Ties between the values of this evaluative function for different atoms were broken arbitrarily.

3. The construction of these clauses was repeated until all the examples in the $E^-$ set had been rejected. (A CNF expression was completed when all its clauses when taken together rejected all the examples in the set $E^-$.)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The ratio $\frac{\text{POS}(a)}{\text{NEG}(a)}$ in Step 1 (i.e., the number of positive examples accepted by the atom $a$, divided by the number negative examples accepted by the same atom) in this heuristic represents an evaluative function for the selection of the best set of attributes to be included in a clause. In this dissertation work, an attribute $a$, that made $\text{NEG}(a) = 0$ (i.e., no negative examples are accepted by $a$) was rewarded by making $\frac{\text{POS}(a)}{\text{NEG}(a)} = \text{POS}(a) \times 1,000$. The multiplication by 1,000 (a sufficiently high number) was introduced in this function in order to avoid a division by zero. This was a high reward value because an atom that does not accept any element from the $E^-$ set is a good candidate to be included in the clause being formed. On the other hand, an atom $a$, that made $\text{POS}(a) = 0$ and $\text{NEG}(a) = 0$ was associated with a punishment value by making $\frac{\text{POS}(a)}{\text{NEG}(a)} = -1$ because such an atom does not accept examples from the $E^+$ set nor from the $E^-$ set.

4.2.2 Construction of a Centroid Using the VSM Model

The centroid of a document class was computed according to Equation (3.3) on page 14. To illustrate the computation of centroid $C_+$, for the set $E^+$ consider the following examples.

$$E^+ = \begin{bmatrix} 0100 \\ 1100 \\ 0011 \\ 1001 \end{bmatrix}$$

$$E^- = \begin{bmatrix} 1010 \\ 0001 \\ 1111 \\ 0000 \\ 1000 \\ 1110 \end{bmatrix}$$

According to Equation (3.3), the first element of $C_+$ is $w_{+,1} = \frac{1}{4} \times (0 + 1 + 0 + 1) = 1/2$. Notice that the four binary numbers in the parenthesis correspond to the first binary element.
of the four examples in $E^+$. The remaining three weights $w_{+,3}$, $w_{+,4}$, and $w_{+,4}$ of $C_+$ are computed similarly. Hence, all four weights of centroid $C_+$ are:

\[
\begin{align*}
    w_{+,1} &= \frac{1}{4} \times (0 + 1 + 0 + 1) = \frac{1}{2}, \\
    w_{+,2} &= \frac{1}{4} \times (1 + 1 + 0 + 0) = \frac{1}{2}, \\
    w_{+,3} &= \frac{1}{4} \times (0 + 0 + 1 + 0) = \frac{1}{4}, \\
    w_{+,4} &= \frac{1}{4} \times (0 + 0 + 1 + 1) = \frac{1}{2}.
\end{align*}
\]

Therefore, the centroid of $E^+$ can be defined by $C_+ = [1/2, 1/2, 1/4, 1/2]$. When these computations are extended to the examples in the $E^-$ set, then the following centroid $C_-$ is obtained.

**4.2.3 Experimental Conditions for Problem 2**

In Section 2 the second problem was defined as how to construct a set of rules for classifying text documents. As indicated earlier, this problem was tackled by investigating the classification performance of the OCAT and VSM algorithms under the following two experimental conditions: (i) The leave-one-out cross validation (CV) and (ii) a 30/30 cross validation (30/30 CV), where 30 indicates the initial number of documents in the training classes $E^+$ and $E^-$. (Similar experimental conditions can be seen in [Deshpande and Triantaphyllou, 1998] and [Utgoff, 1997], respectively). We implemented the CV validation on the five class-pairs of the TIPSTER collection that were defined in Section 4.1.1, as follows:

**Step 1.** 30 documents from each class were randomly selected and were assigned to $E^+$ and $E^-$, respectively.

**Step 2.** One document was then removed from either $E^+$ or $E^-$, and the *positive and negative rules and positive and the negative centroids* (for the OCAT and
VSM cases, respectively) were then constructed using the remaining 59 documents. A different document was removed every time.

Step 3. The class of the document left out was then inferred by both algorithms.

Step 4. The correctness of the classification of both algorithms was determined according to the criteria in Figure 3.7 (on page 34).

Steps 2 through 4 were repeated until all 60 documents had their class inferred. Finally, this test was replicated ten times at which point the results of the two algorithms were tested for statistical difference.

The 30/30 cross validation (30CV) was implemented on larger samples of 254 documents. This number of documents was used in this experiment in order to avoid excessive computational times, but it still yielded statistically significant results. We also implemented this experimental condition on the five class-pairs indicated in Section 4.1.1., as follows:

Step 1. 30 documents from each class were randomly selected, and they were assigned to $E^+$ and $E^-$, respectively.

Step 2. The Positive and negative rules and the positive and negative centroids (for the OCAT and VSM cases, respectively) were then constructed using the data in $E^+$ and $E^-$. 

Step 3. Next, the classification of all 254 documents was inferred.

Step 4. The correctness of the classification of both algorithms was determined according to the criteria in Figure 3.7 (on page 34).
As with the first experimental condition, the 30CV validation was replicated ten times at which point the results of the two algorithms were tested for statistical difference.

4.2.4 Statistical Performance of the Two Algorithms

In order to compare the relative performance of both algorithms, the following two sets of hypotheses were tested.

a) **One tail test:**

\[
\begin{align*}
H_0 : \overline{P_{\text{OCAT}}} &= \overline{P_{\text{VSM}}} \\
H_1 : \overline{P_{\text{OCAT}}} &> \overline{P_{\text{VSM}}}
\end{align*}
\]

Where, \(\overline{P_{\text{OCAT}}}\) and \(\overline{P_{\text{VSM}}}\) are the proportions of documents that were inferred correctly by both algorithms.

b) **Sign Test:**

The sign test was implemented by comparing the individual difference \(P_{\text{VSM}} - P_{\text{OCAT}}\) of all fifty replications in the experiments (i.e., 10 replications \(\times\) the five class-pairs). In this test, \(P_x\) was the proportion of documents with a "Correct" classification at the end of replication \(j\) (for \(j = 1, 2, 3, \ldots, 50\)) that was obtained with algorithm \(x\) (where \(x = \text{OCAT}\) or \(\text{VSM}\)) is the divided by the number of documents in replication \(j\). Thus, if the above difference was positive(negative), a "+"("-") sign was scored. To implement this test, the following hypotheses were used:

\[
\begin{align*}
H_0 : p &= 0.50 \\
H_1 : p &< 0.50
\end{align*}
\]

Where \(p\) is the probability of finding an equal number of positive and negative differences in a set of outcomes.
4.3 Methodology for Problem 3: A Guided Learning Approach

In order to answer the question *what is the next document to be inspected by the expert classifier in order to improve the performance of the current classification rule*, only the OCAT algorithm was investigated under the RANDOM and the GUIDED learning approaches. The experimental condition for these two approaches was as follows:

Step 1. Three samples of 510 documents each (255 from each class) from the three class-pairs: (DOE vs. ZIPF), (AP. vs. DOE), and (WSJ vs. ZIPF) were used. One sample was extracted from each class-pair. The number of documents (i.e., 510) was determined by the maximum size of the ASCII files the Turbo Pascal 1.5 programming language for Windows could open simultaneously.

Step 2. Positive and negative rules were constructed using only 30 documents that were randomly selected from each class-pair. As indicated in an earlier section, the first and second classes were identified as $E^+$ and $E^-$, respectively.

Step 3. The class of all 510 documents in the experiment was then inferred. As with the previous two problems, the criteria illustrated in Figure 3.6 were used to determine the number of "Correct," "Incorrect," and "Undecided" classifications.

Step 4. One more document was selected and added to a class. For the RANDOM approach this document was selected randomly from the set of unclassified documents. While for the GUIDED approach this document was selected from the set of documents that the current rules had already identified with an "Undecided" classification. The latter approach was temporarily substituted with
a RANDOM approach if "Incorrect" (as opposed to "Undecided") classifications were only detected by the two rules.

Step 5. An experiment with a class-pair was terminated when one of the following two conditions was met: (i) the OCAT algorithm had constructed a rule that classified all 510 documents accurately or (ii) all 510 documents had been included in $E^+$ and $E^-$.  

4.4 Methodology for Problem 4: An Incremental Learning Approach

The methodology for repairing only the portion(s) of the knowledge that cause a misclassification followed the experimentation described in [Michalski and Larson, 1978]. In the methodology presented next, however, this repair was divided into the following two subproblems: (i) repair of a Boolean function that rejects a positive example $e^+$ and (ii) repair of a Boolean function that accepts a negative example $e^-$. All functions were expressed in DNF form. The decision for using DNF expressions (as opposed to the CNF expressions that were used in the previous three sections of this methodology) was needed in order to compare the results of this experimentation with the ones in the existing literature. The methodology for the two subproblems is presented next.

4.4.1 Repair of a Boolean Function that Rejects Erroneously a Positive Example

As indicated in Section 3.2.4, a Boolean function $\mathcal{F}$ accepts a positive example $e^+$ if and only if it evaluates to true (denoted by 1). Consequently, when an example $e^+$ is rejected erroneously by function $\mathcal{F}$, the following condition is satisfied:

$$\mathcal{F}(e^+) = c_1 \lor c_2 \lor c_3 \lor \ldots \lor c_n = 0. \quad (4.2)$$
Where \( c_i \) (for \( i = 1, 2, 3, \ldots, n \)) is the \( i \)th clause of the DNF function. Equation (4.2) indicates that when this condition is met, then all \( n \) clauses have rejected the example \( e^+ \).

Hence, the methodology described next addressed the problem: **how to select the clause to be repaired such that when it is updated it will make the function \( \mathcal{F} \) to accept the example \( e^- \).** Figure 4.1 illustrates the methodology that addresses this problem.

**Input:** Let \( e^- \) be the positive example rejected by function \( \mathcal{F} = c_1 \lor c_2 \lor \ldots \lor c_n \).

The sets \( E^- \) and \( E^+ \).

**Output:** A Boolean function \( \mathcal{F}' \) in DNF form that accepts all examples in \( E^- \lor e^- \) and rejects all examples in \( E^+ \).

```
begin
Step 1: let \( \mathcal{E}^- (c_i) \) (for \( i = 1, 2, 3, \ldots, n \)) be the set of the members of \( E^- \) which are accepted by clause \( c_i \);
Step 2: let \( A(c_i) \) be the number of atoms in \( c_i \);
Step 3: select \( c_k \) (for \( 1 \leq k \leq n \)) according to a predetermined strategy\(^1\);
Step 4: \( \mathcal{F} = \mathcal{F} - c_k \);
Step 5: let \( \mathcal{E}^- (c_k) = \mathcal{E}^- (c_k) \lor e^- \);
Step 6: let \( f \) be the DNF Boolean function that solves the subproblem \( \text{OCAT}(\mathcal{E}^- (c_k), E^-) \);
Step 7: let \( \mathcal{F}' = \mathcal{F} \lor f \);
end;
```

\(^1\) predetermined strategy:

- **Most generalizing clause:** \( \max \{ |\mathcal{E}(c_i)| / A(c_i) \} \);
- **Least generalizing clause:** \( \min \{ |\mathcal{E}(c_i)| / A(c_i) \} \);

**Figure 4.1.** Algorithm for updating a Boolean function that rejects erroneously a positive example.

To tackle this problem, the following two criteria for selecting the clause to be repaired were investigated: Selection of **the most generalizing clause** and selection of **the least generalizing clause**. The goal of investigating these criteria was to determine which one produced a Boolean function with fewer number of clauses.
The key concept for achieving an incremental learning solution was the size of the subproblem $OCAT(\mathcal{E}^+(c_j), E^-)$ in Step 6 of the algorithm in Figure 4.1. This was the key concept because if it is assumed that $|\mathcal{E}^+(c_j)| < |E^-|$ (where, $|x|$ is the size of the set $x$), then the CPU time for solving the subproblem $OCAT(\mathcal{E}^+(c_j), E^-)$ will be shorter than the time required to solve the complete problem $OCAT(E^+ \cup e^+, E^-)$.

At this moment, it is important to notice that by using either of the two criteria, it may happen that one or more atoms or clauses will be added to the function $\mathcal{F}$. According to [Michalski and Larson, 1978], this situation can be anticipated “because the utilization of ILE approaches always renders more complex systems.” This is an important situation because if more atoms or clauses are added to the function $\mathcal{F}$, then it can be also anticipated that this function will not only accept the example $e^+$, but it may also accept other unclassified examples. This is a desired situation because it suggests that ILE approaches will not only require shorter processing times than NILE approaches, but also because it suggests that the former approach may become more accurate at a faster rate than the latter one.

4.4.2 Repair of a Boolean Function that Accepts Erroneously a Negative Example

The algorithm in Figure 4.2 addresses the second subproblem for repairing a Boolean function $\mathcal{F}$ that accepts a negative example $e^-$. In this case, if this situation arises, then function $\mathcal{F}(e^-)$ satisfies the condition:

$$\mathcal{F}(e^-) = c_1 \lor c_2 \lor c_3 \lor \ldots \lor c_n = 1. \quad (4.3)$$

An inspection of Equation (4.3) suggests that when this condition is met at least one of the $n$ clauses erroneously accepts example $e^-$. Thus, the main concern in this second subproblem
was how to select the clause(s) to be repaired such that the updated function $\mathcal{F}''$ also rejects the example $e$. 

The algorithm in Figure 4.2 solves the problem posed in Equation (4.3) by first identifying the set of clauses $C$ that accept the example $e$, and then by forming a subset of positive examples $E^+(C)$, which is composed of the examples in $E^+$ that are accepted by the clauses $C$. The set of negative examples is formed by $E^- \cup e^-$. As with the first subproblem, the key concept for achieving this incremental learning was also the size of the expression $\text{OCAT}(E^+(C), (E^- \cup e^-))$ at Step 4 in Figure 4.2. An inspection of this expression indicates that if $|E^+(C)| < |E^+|$ is true (where, $|x|$ is the size of the set $x$), then the CPU time for solving this subproblem is most likely shorter than the time for solving the problem $\text{OCAT}(E^-, E^- \cup e^-)$.

It is interesting to notice that the CPU time for this second subproblem is bounded by the size of the subset $E^-(C)$. From Equation (4.3) it follows that if all $n$ clauses accept the

| Input: | Let $e^-$ be the negative example that is accepted by function $\mathcal{F} = c_1 \lor c_2 \lor \ldots \lor c_n$. The sets $E^-$ and $E^+$. |
| Output: | A new classification rule $\mathcal{F}''$ that accepts all the examples in $E^+$ and rejects the example in $E^- \cup e^-$. |
| begin | |
| Step 1: | let $C$ be the set of clauses $c_i$ (i = 1, 2, 3, ..., n) that accept $e^-$. |
| Step 2: | let $\mathcal{F} = \mathcal{F} - C$; |
| Step 3: | let $E^+(C)$ be set of the members of $E^+$ which are accepted by $C$; |
| Step 4: | let $f$ be the Boolean function in DNF form that solves the subproblem $\text{OCAT}(E^+(C), (E^- \cup e^-))$; |
| Step 5: | let $\mathcal{F}'' = \mathcal{F} \cup f$; |
| end; |

Figure 4.2. Algorithm for updating a Boolean function that accepts erroneously a negative example.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
example $e^-$, then $E^+(C) = E^+$. Moreover, if this equality condition occurs, it would indicate that a NILE process to solve the problem $OCAT(E^+, E^- \cup e^-)$ will be executed, and therefore it would indicate that an expensive repair (i.e., a long CPU time) will occur. This is also an interesting situation because it can also be anticipated that occasional NILE processes may reduce the size of the Boolean function. In fact, [Reinke and Michalski, 1986] have already indicated "that although NILE processes are expensive, they produce 'less complex' (small) systems."

4.4.3 Computational Complexity of the algorithms for the ILE Approach

An inspection of the algorithms in Figure 4.1 and Figure 4.2 indicates that in the worst case scenario, NILE approaches are executed to solve the problems $OCAT(E^+ \cup e^+, E^-)$ and $OCAT(E^+, E^- \cup e^-)$. Furthermore, in [Triantaphyllou and Deshpande, 1998] the computational complexity of the NILE approaches to be used was defined as $O(n^2)$. Therefore, the complexity of the algorithms introduced in Sections 4.4.1 and 4.4.2 is also $n^2$ because in the worst case scenario they reduce to a NILE approach.

To conclude the methodology for the ILE approach, the experimental conditions for the GUIDED learning approach (Section 4.3) were implemented to test the performance of the incremental algorithms. In this case, however, only 336 documents were considered in the experiment. This number of 336 documents was selected based on the observation that the non-incremental OCAT algorithm became 100% accurate with these many documents when it was implemented using the GUIDED learning approach. Additionally, the performance measures to test the two versions of ILE were: the classification accuracy, the total number of clauses in the positive and negative rules, and the CPU time. These
performance measurements were statistically compared with those derived under the NILE approach by using a Sign Test (as described in Section 4.2.4).

4.5 Experimentation Tools

The computer programs for this methodology were written in the computer language Turbo PASCAL 1.5 for Windows [Borland, 1991]. This computer language was selected for portability reasons; although, C++ could have been a much better selection. Moreover, the computational experiments for the first, second, and third problems investigated in this research were run on a Pentium 166MHz PC. The experiments for the ILE problem were run on a Pentium II 400MHz PC. Finally, all computational results were analyzed with the statistical package MINITAB 6.1 [Ryan et al., 1992].
5 AN ILLUSTRATIVE EXAMPLE

The example in this section illustrates the construction of a Boolean function in the CNF form and the construction of a pair of centroids by using the OCAT and VSM algorithms, respectively. This illustrative example considers the ten surrogates $D_1, D_2, D_3, \ldots, D_{10}$ shown in Figure 5.1. These documents are indexed by the presence or absence of keywords $T_1, T_2, T_3,$ and $T_4$. Additionally, to comply with the requirements of the new classification problem, the surrogates $D_1, D_2, D_3,$ and $D_4$ are assigned to the set $E^+$, while the remaining surrogates $D_5, D_6, D_7, D_8, D_9,$ and $D_{10}$ are assigned to the set $E^-$.

![Figure 5.1. Set of documents to be analyzed.](image)

5.1 Construction of a Boolean Function by Using the OCAT Algorithm

The construction of a Boolean function in CNF form that can classify the ten documents in Figure 5.1 can be done as follows (the heuristic to construct this function is described in Figure 4.1.). The first clause is constructed such that it accepts all four surrogates in $E^+$ while rejects many surrogates from $E^-$. (A function in a DNF form can be easily obtained by making some simple transformation in the input data as described in [Triantaphyllou and Soyster, 1995].) The next clause accepts again all four surrogates in $E^+$.
and rejects more surrogates from $E^-$ that were not rejected by the previous clause(s). This process is repeated until the CNF expression rejects all the examples in $E^-$.

The set of all keywords for this example is defined as \{$T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8$\}. Hence, according to Step 1 of the heuristic describe in Figure 4.1, the various $POS(T_j)$ and $NEG(T_j)$ values and $POS(T_j)/NEG(T_j)$ ratios for this example are as follows:

- $POS(T_1) = 2$  $NEG(T_1) = 4$  $POS(T_1) / NEG(T_1) = 1/2$
- $POS(T_2) = 2$  $NEG(T_2) = 2$  $POS(T_2) / NEG(T_2) = 1$
- $POS(T_3) = 1$  $NEG(T_3) = 3$  $POS(T_3) / NEG(T_3) = 1/3$
- $POS(T_4) = 2$  $NEG(T_4) = 2$  $POS(T_4) / NEG(T_4) = 1$
- $POS(T_5) = 2$  $NEG(T_5) = 2$  $POS(T_5) / NEG(T_5) = 1$
- $POS(T_6) = 2$  $NEG(T_6) = 4$  $POS(T_6) / NEG(T_6) = 1/2$
- $POS(T_7) = 3$  $NEG(T_7) = 3$  $POS(T_7) / NEG(T_7) = 1$
- $POS(T_8) = 2$  $NEG(T_8) = 4$  $POS(T_8) / NEG(T_8) = 1/2$

From the evaluative function of Step 2, $\max(POS(T_j) / NEG(T_j)) = 1$. However, an inspection of the above $POS(T_j)/NEG(T_j)$ ratio indicates that a tie exists between keywords $T_5$, $T_7$, and $T_8$. For this numerical example, $T_5$ was selected as the first keyword in clause $K_1$. That is, $K_1 = (T_5)$. It is interesting to notice that at this moment, $K_1$ only accepts the surrogates $D_1$ and $D_2$ from the set $E^+$ and rejects only the surrogate $D_7$ from $E^-$. Furthermore, notice that because $E^+ \neq \emptyset$ and $E^- \neq \emptyset$, at least one more keyword must be added to $K_1$.

The second keyword is obtained in a similar manner. That is, this new keyword must accept documents from the updated set $E^+ = \{D_3, D_4\}$ and must reject more documents from the updated $E^- = \{D_3, D_4, D_3, D_3\}$. From these updated sets, the newer $POS(T_j)$ and $NEG(T_j)$ values and $POS(T_j)/NEG(T_j)$ ratios are computed as follows:

- $POS(T_1) = 1$  $NEG(T_1) = 2$  $POS(T_1) / NEG(T_1) = 1/2$
- $POS(T_2) = 1$  $NEG(T_2) = 1$  $POS(T_2) / NEG(T_2) = 1$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\[ \begin{align*}
PO\left(T_2\right) &= 2 \\
NEG\left(T_2\right) &= 1 \\
POS\left(T_2\right) / NEG\left(T_2\right) &= 2 \\
PO\left(T_3\right) &= 1 \\
NEG\left(T_3\right) &= 2 \\
POS\left(T_3\right) / NEG\left(T_3\right) &= 1/2 \\
PO\left(T_4\right) &= 1 \\
NEG\left(T_4\right) &= 3 \\
POS\left(T_4\right) / NEG\left(T_4\right) &= 1/3 \\
PO\left(T_5\right) &= 0 \\
NEG\left(T_5\right) &= 3 \\
POS\left(T_5\right) / NEG\left(T_5\right) &= 0.
\end{align*} \]

(Please observe that in these computations keyword \(T_2\) and its negation \(T_3\) are omitted for further consideration. This is done because \(T_2\) is already included in \(K_i\)).

In this case, the evaluative function \(\max(POS(T_j) / NEG(T_j)) = 2\), which corresponds to keyword \(T_2\). Consequently, \(K_i = (T_2 \lor T_4)\). Now \(K_i\) already accepts all the documents in \(E^-\), while it still accepts documents \(D_6, D_7,\) and \(D_{10}\) from \(E^+\). That is now, we have: \(E^+ = \emptyset\) and \(E^- \neq \emptyset\). The first equality condition indicates that clause \(K_i\) has been completed, whereas the second condition indicates that at least one more clause needs to be constructed.

The second clause \(K_2\) must also accept all the surrogates in \(E^+ = \{D_3, D_5, D_6, D_4\}\) and must reject more documents from the revised set \(E^- = \{D_6, D_7, D_{10}\}\) (please note that now \(E^-\) is reset to the original set). After two more iterations of the OCAT algorithm, we get: \(K_2 = (T_7 \lor T_5)\). The third clause \(K_3 = (T_5 \lor T_2 \lor T_7)\) was constructed in a similar manner.

When all these three clauses are taken together, the following CNF expression is formed:

\[ (T_2 \lor T_4) \land (T_7 \lor T_5) \land (T_5 \lor T_2 \lor T_7). \tag{5.1} \]

Expression (5.1) corresponds to the positive rule (Boolean function). The negative rule can be constructed in a similar fashion. To construct the negative rule, documents in the set \(E^-\) are now treated as the positive examples, while documents in the set \(E^+\) are treated as the negative examples. For the data in Figure 5.1, the negative rules turns out to be:

\[ (T_3 \lor T_2) \land (T_7 \lor T_2 \lor T_5) \land (T_5 \lor T_3). \tag{5.2} \]
5.2 Construction of a Pair of Centroids Using the VSM Algorithm

The construction of a pair of centroids can be illustrated by considering again the ten documents in Figure 5.1. Recall that it was assumed that the first class is $E^+ = \{D_1, D_2, D_3, D_4\}$ and that the second class is $E^- = \{D_5, D_6, D_7, D_8, D_9, D_{10}\}$. For convenience, the centroid of each class is denoted as $C_+$ and $C_-$, respectively. Then, according to Equation (3.3), the weights of the keywords of these two centroids are computed as follows. For centroid $C_+$, these weights are

$$w_{+,1} = \frac{1}{4} \times (0 + 1 + 0 + 1) = \frac{1}{2},$$
$$w_{+,2} = \frac{1}{4} \times (1 + 1 + 0 + 0) = \frac{1}{2},$$
$$w_{+,3} = \frac{1}{4} \times (0 + 0 + 1 + 0) = \frac{1}{4},$$
$$w_{+,4} = \frac{1}{4} \times (0 + 0 + 1 + 1) = \frac{1}{2}.$$

And, for centroid $C_-$ these weights are

$$w_{-,1} = \frac{1}{6} \times (1 + 0 + 1 + 0 + 1 + 1) = \frac{2}{3},$$
$$w_{-,2} = \frac{1}{6} \times (0 + 0 + 1 + 0 + 0 + 1) = \frac{1}{3},$$
$$w_{-,3} = \frac{1}{6} \times (1 + 0 + 1 + 0 + 0 + 1) = \frac{1}{2},$$
$$w_{-,4} = \frac{1}{6} \times (0 + 1 + 1 + 0 + 0 + 0) = \frac{1}{3}.$$

Finally, the centroids of these two classes are defined as $C_+ = [\frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}]$ and $C_- = [\frac{2}{3}, \frac{1}{3}, \frac{1}{2}, \frac{1}{3}]$, respectively.

As with the Boolean function (5.1), this pair of centroids corresponds to the positive centroids. The negative pair of centroids can be constructed by considering the documents in the set $E^-$ as the positive examples while documents in the set $E^+$ as the negative examples. Under this new setting, the negative pair of centroids for the data in Figure 5.1 is: $C_+ = [\frac{2}{3}, \frac{1}{3}, \frac{1}{2}, \frac{1}{3}]$ and $C_- = [\frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}]$, respectively.
6 COMPUTATIONAL RESULTS

This section presents the results of some computational experiments related to the four problems investigated in this research. As indicated in Section 2, the first problem regards the selection of a set of text features (i.e., keywords) to index the documents in the experiments. The second problem addresses a methodology for constructing patterns of keywords in the form of Boolean functions. The third problem regards a methodology to guide the learning of the Boolean functions as more text documents become available. Finally, the forth problem addresses the development of an incremental learning algorithm to improve the efficiency of the OCAT approach for constructing Boolean functions.

6.1 Results for Problem 1: Extraction of Text Features

The bars in Figure 6.1 summarize the average classification accuracy of the OCAT and VSM algorithms after completing experiments (as described in Section 4.1.4) with patterns of 1, 2, 3, 4, and 5 consecutive words. The numbers above the bars in this figure show the average classification accuracy and the variance under each algorithm for each set of experiments.

An inspection of this figure reveals that the classification accuracy of the OCAT algorithm was not affected by the various sizes of the patterns of consecutive words. Actually, an investigation of this situation revealed that this accuracy was obtained because the ratio $\frac{POS(i)}{NEG(i)}$ was always larger for patterns of single words than it was for patterns of multiple words. This was an unexpected but encouraging result because it allowed us to abate the number of keywords to solve large classification problems without
reducing the accuracy. The following example with surrogates $d_1, d_2, d_3, \ldots, d_8$ attempts to illustrate the importance of the above finding.

![Graph showing accuracy of patterns of 1, 2, 3, 4, and 5 consecutive words.](image)

**Figure 6.1.** Average classification accuracy using patterns of 1, 2, 3, 4, and 5 consecutive words.

\[
\begin{align*}
T_1 & \quad \cdots \quad T_{13} \\
E^+ \quad d_1 & = [111000000000000] \\
E^- \quad d_2 & = [001000000000000] \\
\quad d_3 & = [010110000000000] \\
\quad d_4 & = [100010100000000] \\
\quad d_5 & = [000100110000000] \\
\quad d_6 & = [001000001100000] \\
\quad d_7 & = [000000000011110] \\
\quad d_8 & = [000000000000001] \\
\end{align*}
\]

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
These eight surrogates are defined by the presence or absence of the patterns (i.e., sequence of keywords) $T_1$, $T_2$, $T_3$, ..., $T_{15}$. The underlined elements are assumed to correspond to patterns of two consecutive words, whereas the non-underlined elements are single-word patterns. These eight surrogates were processed using two strategies. In the first strategy, a subproblem with surrogates consisting of one word was considered. That is, in this case the surrogates consisted of the non-underlined elements $T_1$, $T_2$, $T_4$, $T_5$, $T_8$, $T_{10}$, $T_{12}$, $T_{13}$ and $T_{15}$. In the second strategy the surrogates consisted of all fifteen elements. The results of this brief illustrative example are presented in Table 6.1.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Rule</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Words</td>
<td>$(T_1 \lor T_2 \lor T_3) \land (T_{10})$</td>
<td>5.4 Seconds</td>
</tr>
<tr>
<td>Two Consecutive Words</td>
<td>$(T_1 \lor T_2 \lor T_3) \land (T_{10})$</td>
<td>7.2 Seconds</td>
</tr>
</tbody>
</table>

It can be easily verified that both rules classify accurately these examples.

The second column of the above table shows that the two classification rules are identical despite the different size of the patterns. These identical rules indicate that the OCAT algorithm was insensitive to the size of the word patterns. Nonetheless, the CPU times on the last column show that the OCAT algorithm was faster by 1.8 seconds (30%) when it processed surrogates defined on single-word patterns than when it processed surrogates of two consecutive words. Similar results were also observed during the pilot runs with the various pseudo-documents that were extracted from the DOE test documents. A pseudo-document can be constructed by using a section of an actual document [Jacobs and Rau, 1990]. In the pilot runs, a paragraph was considered as a document. Therefore, there were as many pseudo-documents as paragraphs in a document.
This was an important finding because it made possible to design more efficient experiments. For instance, in experiments in which there were sixty training documents and about eight hundred keywords, the CPU time to produce classification rules was abated from about four minutes to approximately 1.5 minutes.

The results for VSM were also encouraging because the small variances in its classification suggested that it maintained its accuracy for the various sizes of the patterns. Therefore, in the remaining of this work only a single-word keywords strategy was implemented to address Problems 2 through 4.

6.2 Results for Problem 2: Construction of Classification Rules

The following Table 6.2 summarizes the average number of unique keywords that were extracted from the documents during the two experimental conditions: CV and 30CV. In the experimental conditions described in Section 4.2.3, it was indicated that ten replications were executed for both experimental conditions. Therefore, the number of indexing words in this table were computed by averaging the number of keywords in the ten dictionaires of unique content descriptors of the ten replications.

<table>
<thead>
<tr>
<th>Experimental Condition</th>
<th>DOE vs. AP</th>
<th>DOE vs. WSJ</th>
<th>DOE vs. ZIPF</th>
<th>AP vs. WSJ</th>
<th>WSJ vs. ZIPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV</td>
<td>511</td>
<td>605</td>
<td>479</td>
<td>448</td>
<td>501</td>
</tr>
<tr>
<td>30CV</td>
<td>803</td>
<td>911</td>
<td>890</td>
<td>814</td>
<td>811</td>
</tr>
</tbody>
</table>

- To avoid excessive CPU times, only documents with approximately 150 words were considered.
- Stop words were always removed.

The following Tables 6.3 and 6.4 summarize the computational results for the CV validation and the 30CV validation, respectively. The abbreviations “C:”, “I:”, and “U:” in the first column of both tables correspond to the “Correct,” “Incorrect,” and “Undecided”
classifications as they were defined in Figure 3.6. For instance, the data in Table 6.3 for the class-pair DOE vs. AP indicate that the VSM identified 334 “Correct,” 261 “Incorrect,” and 5 “Undecided” classifications. Similarly, the data for the same class-pair indicate that the OCAT algorithm identified 410 “Correct,” 5 “Incorrect,” and 185 “Undecided” classifications. In addition, the last two columns of these two tables summarize the results across all five class-pairs. For example, the number of correct classifications of the VSM in Table 6.3 is 334+286+280+316+286 = 1,502, whereas this number for the OCAT algorithm is 410+296+358+365+303 = 1,732. Figure 6.2 compares graphically the data in these last two columns.

Table 6.3. Summary of First Experimental Condition: Leave-One-Out Cross Validation

<table>
<thead>
<tr>
<th></th>
<th>DOE vs. AP</th>
<th>DOE vs. WSJ</th>
<th>DOE vs. ZIPF</th>
<th>AP vs. WSJ</th>
<th>WSJ vs. ZIPF</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:</td>
<td>334</td>
<td>286</td>
<td>280</td>
<td>316</td>
<td>286</td>
<td>1,502</td>
</tr>
<tr>
<td>I:</td>
<td>261</td>
<td>314</td>
<td>320</td>
<td>284</td>
<td>314</td>
<td>1,493</td>
</tr>
<tr>
<td>U:</td>
<td>5</td>
<td>66</td>
<td>25</td>
<td>47</td>
<td>76</td>
<td>219</td>
</tr>
</tbody>
</table>

The VSM was implemented using the CC coefficient, following the suggestions in [Salton, 1989].

Table 6.4. Summary of Second Experimental Condition: 30/30 Cross Validation

<table>
<thead>
<tr>
<th></th>
<th>DOE vs. AP</th>
<th>DOE vs. WSJ</th>
<th>DOE vs. ZIPF</th>
<th>AP vs. WSJ</th>
<th>WSJ vs. ZIPF</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:</td>
<td>975</td>
<td>1,266</td>
<td>1,035</td>
<td>1,088</td>
<td>1,088</td>
<td>5,161</td>
</tr>
<tr>
<td>I:</td>
<td>975</td>
<td>134</td>
<td>915</td>
<td>846</td>
<td>837</td>
<td>4,548</td>
</tr>
<tr>
<td>U:</td>
<td>0</td>
<td>550</td>
<td>0</td>
<td>16</td>
<td>25</td>
<td>41</td>
</tr>
</tbody>
</table>

The VSM was implemented using the CC coefficient, following the suggestions in [Salton, 1989].
Two facts can be derived from the size of the dark (i.e., the "Undecided" classifications) areas in Figure 6.2. First, it can be observed that the VSM detected only 5 "Undecided" situations under the CV validation and 41 under the 30CV validation. An investigation of these situations revealed that these instances (i.e., 5 and 41) were consequences of pure chance. Actually, it was found that they occurred when the following two conditions were satisfied:

1. First, the class of a document was randomly selected because the two positive centroids had simultaneously classified it in the same class. That is, because Equation 3.3 had computed two identical centroids.
2. Second, the two negative centroids also classified the document in the same class as the positive centroids did, and therefore the class of the document had to be also randomly assigned. More precisely, it was found that the root cause of these "Undecided" situations occurred when Equation 3.3 had computed four identical centroids. This finding suggested that in its present form the VSM had no capabilities for accurately identifying "Undecided" situations.

The second fact regards the contrasting 1,049 and 2,686 "Undecided" classifications the OCAT algorithm identified under the two experimental conditions. In Sections 3.2.4 and 3.3 it was indicated that "Undecided" situations occur when the positive and negative rules simultaneously classify a document in the same class because they do not possess enough knowledge to classify it correctly. Moreover, it was also indicated that these types of situations can be desirable because they show that either the positive or the negative rule needs to be updated as more examples become available. These results are encouraging because they suggest that the classification accuracy of the OCAT algorithm can be improved if "Undecided" situations are used to guide its learning.

Two additional facts can also be observed from the proportion of "Incorrect" classifications (the white areas) in Figure 6.2. First, it can be seen that the VSM incurred 1,493 "Incorrect" classifications under the CV validation, and that it incurred 4,548 cases under the 30CV validation. These large proportions of "Incorrect" classifications were unexpected, however, they suggested that the VSM did not distinguish "Incorrect" from "Undecided" situations. These large numbers of "Incorrect" classifications can be attributed to the VSM performance under pseudo-classification. As indicated in Section 3.2.5, when
the number of classes is prefixed (two in this research), the VSM is said to perform a pseudo-classification.

More importantly, the large proportions of "Incorrect" situations suggest that in its present form the VSM cannot further improve its classification accuracy. The second fact regards the 219 and 644 "Incorrect" classifications the OCAT algorithm made under the two experimental settings, respectively. These small numbers of "Incorrect" classifications were attributed to the utilization of positive and negative rules of small size that enabled the OCAT algorithm to distinguish between "Incorrect" and "Undecided" classifications.

Finally, the statistical performance of the two algorithms was determined by using only the "Correct" classification proportions that are indicated in the last column of Tables 6.3 and 6.4. It is important to mention here that "Incorrect" and "Undecided" outcomes were both considered as incorrect classifications. As it was indicated in Section 4.2.4, the comparison of these two algorithms was needed in order to determine which one better addressed the classification of documents into two disjoint classes. The results of the two statistical tests (i.e., One Tail Test and the Sign Test) are summarized in Tables 6.5 and 6.6.

<table>
<thead>
<tr>
<th>Experimental Condition</th>
<th>( P_{\text{OCAT}} )</th>
<th>( P_{\text{VSM}} )</th>
<th>( P_{\text{VSM}} - P_{\text{OCAT}} )</th>
<th>Binomial Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV</td>
<td>0.577</td>
<td>0.501</td>
<td>-0.076</td>
<td>0.025</td>
</tr>
<tr>
<td>30CV</td>
<td>0.658</td>
<td>0.529</td>
<td>-0.1287</td>
<td>0.014</td>
</tr>
</tbody>
</table>

\[ 1.732 / n \text{ and } 6.420 / n, \text{ where } n = 3.000 \text{ under CV and } 9.750 \text{ under 30CV}. \]

\[ 1.502 / n \text{ and } 5.161 / n, \text{ where } n = 3.000 \text{ under CV and } 9.750 \text{ under 30CV}. \]

\[ \sqrt{\frac{P_{\text{OCAT}}(1-P_{\text{OCAT}})}{n} - \frac{P_{\text{VSM}}(1-P_{\text{VSM}})}{n}} \]. While \( ^* \) denotes that both algorithms perform differently.
64

Table 6.6. Sign Test for the Accuracy of the OCAT and VSM Algorithms.

<table>
<thead>
<tr>
<th>Type of Experiment</th>
<th>CV</th>
<th>30CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of &quot;+&quot; signs</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Number of &quot;-&quot; signs</td>
<td>46</td>
<td>43</td>
</tr>
<tr>
<td>p-value(^5)</td>
<td>2.23×10(^{-10})</td>
<td>1.04×10(^{-7})</td>
</tr>
</tbody>
</table>

\[ \sum_{i=0}^{50} \binom{50}{i} p^i (1-p)^{50-i} \] Where \( m \) was 4 for CV and 7 for 30CV.

The two confidence intervals (C.I.) in Table 6.5 indicate that the OCAT algorithm was statistically more accurate in both experimental conditions than the VSM was. Furthermore, the small \( p\)-values shown in Table 6.6 indicate that it is extremely unlikely to find an equal number of positive and negative signs. Therefore, the results of these two statistical tests led us to conclude with high confidence that the OCAT algorithm better addressed the problem of classifying documents into two disjoint classes.

6.3 Results for Problem 3: Determining which Document to Consider Next

Figures 6.3 through 6.5 show the results of the OCAT algorithm under the RANDOM and GUIDED learning approaches that were described in Section 4.3. The horizontal axis in these figures shows the percentage of training documents in the experiments. For example, at the beginning of the experiments there were only 60 (i.e., 11.76\% of 510) training documents. The vertical axis in these three figures shows the classification levels the OCAT algorithm achieved under various percentages of training documents.

In these two figures the abbreviations \( R_c \), \( R_i \), \( R_u \), \( G_c \), \( G_i \), and \( G_u \) stand for the proportions of "Correct," "Incorrect," and "Undecided" classifications for these two approaches. The "R" and "G" letters stand for RANDOM and GUIDED, respectively. Thus, \( R_c \) is the proportion of "Correct" classifications under the RANDOM approach, and \( G_u \) is
the proportion of "Undecided" classification under the GUIDED approach. A similar interpretation can be extended for the other abbreviations.

Table 6.7 shows the percentage of the training documents at which the OCAT algorithm classified all 510 documents in an experiment. The percentages shown in this table correspond to the position of the vertical (dotted) line in the above three figures. For instance, the datum 65.99% for the class-pair DOE vs. ZIPF under the GUIDED approach corresponds to the value of the vertical line on the horizontal axis in Figure 6.3.

Three observations can be derived from these three figures regarding the proportions of "Correct," "Incorrect," and "Undecided" classifications. First, it can be seen that the rates of the proportions of "Correct" classifications $Gc$ were faster than the rates $Rc$. In addition, Table 6.7 indicates that the OCAT algorithm needed about 34% less training documents to classify all 510 documents correctly by using a GUIDED learning approach than by using the RANDOM approach. These results confirmed the expectation that the learning rate of the OCAT algorithm is faster with the GUIDED than with the RANDOM approach. These results are encouraging because they also agreed with the findings reported in [Triantaphyllou and Soyster, 1996-2] under a different experimental setting.

Table 6.7. Percentage of Training Documents at which the OCAT Algorithm became 100% Accurate.

<table>
<thead>
<tr>
<th>Class-pairs</th>
<th>% Under GUIDED</th>
<th>% Under RANDOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(DOE vs. ZIPF)</td>
<td>65.69</td>
<td>100.00</td>
</tr>
<tr>
<td>(AP vs. DOE)</td>
<td>60.98</td>
<td>99.80</td>
</tr>
<tr>
<td>(WSJ vs. ZIPF)</td>
<td>71.18</td>
<td>99.80</td>
</tr>
<tr>
<td>Average</td>
<td>65.95</td>
<td>99.87</td>
</tr>
</tbody>
</table>

100% accuracy was achieved when the OCAT algorithm classified all 510 documents correctly.
Figure 6.3. Results when the GUIDED and RANDOM approaches were used on the (DOE vs. ZIPF) class-pair.

Figure 6.4. Results when the GUIDED and RANDOM approaches were used on the (WSJ vs. ZIPF) class-pair.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 6.5. Results when the GUIDED and RANDOM approaches were used on the (AP vs. DOE) class-pair.

More importantly these results confirmed the assumption stated in Sections 2.2.4 and 2.3, which indicates that the utilization of training documents with "Undecided" classifications can be used to improve the learning rate of the OCAT algorithm. These results also indicate that GUIDED approaches may use fewer documents and therefore shorter CPU times to address the new document classification problem than the RANDOM approach would. Nonetheless, it is important to mention here that although the previous results are based on a relatively small sample of documents from the TIPSTER collection, they suggest that the OCAT algorithm can be used in the classification of large collections of documents.

The other two main observations that can be inferred from Figures 6.3 through 6.5 regard the rates of the proportions of the "Incorrect" and "Undecided" classifications. First,
these figures show that the rates $G_i$ and $G_u$ reached 0% when about 66% (about 336 documents; Table 6.7) of the 510 documents in the experiments had been included in $E^+$ and $E^-$. The other observation regards the number of documents at which the rates $R_i$ and $R_u$ reached 0%. From these figures it can be seen that $R_i$ and $R_u$ reached 0% when 509 (or 99.8%) documents had been included in the training sets. These results are encouraging because the elimination of "Incorrect" and "Undecided" was much faster by using the GUIDED learning approach than by using the RANDOM approach. Therefore, the results of this third problem led us to conclude that the utilization of the OCAT algorithm under a GUIDED learning approach may also produce shorter CPU times when classifying larger collections of documents into two disjoint classes.

6.4 Results for Problem 4: Incremental Learning with the OCAT Algorithm

The computational results of investigating the performance of the incremental OCAT (the algorithms are described in Figures 4.2 and 4.3) are reported in three separate sections. The first section compares the results of the classification accuracy of NILE and ILE learning algorithms as more training documents were added to the experiments. The second section reports the number of clauses constructed under these two algorithms. Finally, the CPU times used by both algorithms to construct the rules are discussed. As indicated in Section 4.4, the incremental learning problem was investigated by using a GUIDED learning approach on documents from the following three class-pairs of the TIPSTER collection: (DOE vs. ZIPF), (AP vs. DOE), and (WSJ vs. ZIPF). It is important to mention that in the following figures, the thickest lines correspond to the results of the NILE approach, while
the thick and the thin lines represent the results of the ILE approach for the most
generalizing clause criterion and the least generalizing clause criterion, respectively.

6.4.1 Classification Accuracy of the Incremental OCAT

As indicated in the Methodology Section, the results of the non-incremental OCAT
under a GUIDED learning approach were used to benchmark the classification performance
of the incremental OCAT. Table 6.8 summarizes the number of training documents these
two algorithms needed to extract rules that could classify all 510 documents correctly in the
experiments with the three class-pairs from the TIPSTER collection.

The data in this table suggest that the rules constructed by ILE became more accurate
than NILE as more training examples became available. Furthermore, an inspection of
Figures 6.6 through 6.9 indicate that the classification “speed” of ILE (thick and thin lines)
was faster (steeper) than that for the NILE approach (thickest line). This can be verified by
inspecting the “slope” of the two versions of ILE. Such a slope appears to be higher for the
two versions of ILE. Actually, in Section 4.4 it was hypothesized that if an atom (or a
clause) was added to the function, then it was possible that the function could also accept
other unseen examples. This is an interesting situation that suggests that ILE cannot become
more accurate much faster, but also that it may need fewer examples to achieve a high
accuracy.

Table 6.9 in the next page summarizes the number of positive signs of the comparison
between the classification accuracy of the two algorithms as they were trained with more
documents. An interpretation of the data in this table is as follows. The 0 (zero) positive
signs from the comparison NILE - ILE(MGC) for class-pair DOE vs. ZIPF indicates that
ILE(MGC) was always more accurate than the NILE approach. Contrary, the datum for ILE(MGC) - ILE(LGC) for class pair AP vs. DOE indicates that 210 positive signs were obtained. In this case the high number of positive signs shows that the most generalizing criterion was a better performer than the least generalizing criterion.

Table 6.8. Training Documents Needed to Extract Rules that Could Classify Correctly all Documents.

<table>
<thead>
<tr>
<th>Class-pair</th>
<th>DOE vs. ZIPF</th>
<th>AP vs. DOE</th>
<th>WSJ vs. ZIPF</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>NILE(^1)</td>
<td>336</td>
<td>336</td>
<td>336</td>
<td>336</td>
</tr>
<tr>
<td>ILE (MGC)</td>
<td>244</td>
<td>311</td>
<td>288</td>
<td>281</td>
</tr>
<tr>
<td>ILE (LGC)</td>
<td>277</td>
<td>319</td>
<td>303</td>
<td>299</td>
</tr>
</tbody>
</table>

\(^1\)Numbers taken from Table 6.7 for the case of the Guided approach. That is, \([65.95\% \times 510] = 336.\) MGC and LGC stand for Most Generalizing Clause and Least Generalizing Clause, respectively.

Table 6.9. Data for the Statistical Comparison of the Classification Accuracy Between NILE and ILE approaches.

<table>
<thead>
<tr>
<th>Number of Positive Signs</th>
<th>DOE vs. ZIPF</th>
<th>AP vs. DOE</th>
<th>WSJ vs. ZIPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>NILE - ILE(MGC)</td>
<td>0(^1)</td>
<td>1(^1)</td>
<td>5(^1)</td>
</tr>
<tr>
<td>NILE - ILE(LGC)</td>
<td>1(^1)</td>
<td>2(^1)</td>
<td>0(^1)</td>
</tr>
<tr>
<td>ILE(MGC) - ILE(LGC)</td>
<td>92(^E)</td>
<td>210(^E)</td>
<td>163(^E)</td>
</tr>
</tbody>
</table>

MGC and LGC stand for the Most Generalizing Clause and the Least Generalizing Clause, respectively.

All p-values were approximated using \(N(np, np(1-p))\). For all \(^1\) instances, \(n = 250\); while for all \(^E\) instances, \(n\) was 237, 240, and 242, respectively. These different sizes of \(n\) were needed because the all comparisons yielding zero in the above differences were discarded [Barnes, 1994]. (The total number of observations were 274)

\(^1\)denotes a p-value close to 0 and \(^E\) denotes a p-value close to 1.

An inspection of the data in the above table led to the following conclusions. First, the small p-value of the comparison between the classification accuracy of the NILE and ILE algorithms indicate that the two versions of incremental OCAT were much better performers than the non-incremental OCAT. Second, because the comparison between the two versions of ILE rendered opposite p-values, it is not clear which of the two criteria performs better. Further experimentation may be needed to determine the best performer criterion.
Figure 6.6. Accuracy of the NILE and ILE approaches for the class-pair (DOE vs. ZIPF).

Figure 6.7. Accuracy of the NILE and ILE approaches for the class-pair (WSJ vs. ZIPF).
6.4.2 Number of Clauses Under the Incremental OCAT

Figures 6.9 through 6.11 illustrate the sum of the number of clauses in the positive and in the negative rules. Please note that the repair of the Boolean function that accepts a negative example was implemented as described in Figure 4.2. Table 6.10 summarizes the number of clauses constructed by these two approaches to classify correctly all 510 documents in the experiments. A comparison of these number of clauses indicate that ILE always created more clauses than the NILE approach did as more training examples became available. This conclusion can be supported by the zero number of positive signs shown in Table 6.11.
Table 6.10. Number of Clauses at the End of the Experiment.

<table>
<thead>
<tr>
<th>Class-pair</th>
<th>DOE-ZIPF</th>
<th>AP-DOE</th>
<th>WSJ-ZIPF</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>NILE(^1)</td>
<td>23</td>
<td>10</td>
<td>25</td>
<td>19</td>
</tr>
<tr>
<td>ILE (MGC)</td>
<td>47</td>
<td>36</td>
<td>36</td>
<td>40</td>
</tr>
<tr>
<td>ILE (LGC)</td>
<td>37</td>
<td>60</td>
<td>74</td>
<td>57</td>
</tr>
</tbody>
</table>

\(^1\)Numbers taken from the results of the Guided approach.
MGC and LGC stand for the Most Generalizing Clause and the Least Generalizing Clause, respectively.

Regarding the comparison of the two versions of the ILE approach, the small \textit{p-value} and the two zero values shown in Table 6.11 indicate that the most generalizing clause criterion constructed fewer clauses than the least generalizing clause did. These results also confirmed the hypothesis stated in Section 4.4. According to that hypothesis newer clauses could be added to the Boolean function every time the subproblems in \textit{Steps 6} and \textit{4} in Figures 4.2 and 4.3, respectively, are executed. The results are encouraging because they also agree with the results in [Michalski and Larson, 1978] and [Reinke and Michalski, 1986], which indicate that "ILE approaches always render more complex concept descriptions (i.e., more clauses) at reduced CPU times."

Table 6.11. Data for the Statistical Comparison of the Number of Clauses Constructed by the NILE and ILE approaches.

<table>
<thead>
<tr>
<th>Number of Positive Signs for a Sign Test</th>
<th>DOE vs. ZIPF</th>
<th>AP vs. DOE</th>
<th>WSJ vs. ZIPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>NILE - ILE(MGC)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NILE - ILE(LGC)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ILE(MGC) - ILE(LGC)</td>
<td>44(^1)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\(^1\)Numbers taken from the results of the Guided approach.
MGC and LGC stand for the Most Generalizing Clause and the Least Generalizing Clause, respectively.
The following \textit{p-value} was obtained using the normal approximation \(N(np, np(1-p))\), \(n = 243\) since all comparisons yielding zero in the above differences were discarded [Bernoulli, 1994]. (The total number of observations were 274)
\(^1\)Denotes a \textit{p-value} close to 0.

An additional situation can be secured from the above figures. It can be observed that thick and thin lines occasionally reached the thickest line at various points. An investigation...
of this situation indicated that the ILE and the NILE approaches occasionally constructed the same number of clauses. More specifically, it was found that this situation occurred when either the positive or negative rules were reconstructed because they erroneously had accepted negative example. That is, this sudden reduction in the number clauses occurred when the subproblem \( \text{OCAT}(E^+, E^- \cup \neg e^-) \) was executed. This situation was discussed in Section 4.4.2. As the figures show this is an encouraging result because these occasional reconstructions also reduced the complexity of the rule.

---

**Figure 6.9.** Number of clauses under the NILE and ILE approaches for the class-pair (DOE vs. ZIPF).
Figure 6.10. Number of clauses under the NILE and ILE approaches for the class-pair (WSJ vs. ZIPF).

Figure 6.11. Number of clauses under the NILE and ILE approaches for the class-pair (AP vs. DOE).
6.4.3 CPU Times under the Incremental OCAT Approach

Figures 6.12 through 6.14 present the CPU times the NILE and ILE approaches required to update either the positive or the negative Boolean functions. Table 6.12 shows the CPU times at the end of the experiments. These CPU times correspond to the solution of the subproblems in Step 6 or Step 4 in Figures 4.2 and 4.3, respectively.

An inspection of these figures indicates that ILE employed shorter CPU times than the NILE did. Furthermore, the high numbers of positive signs between the comparisons of the CPU times of the NILE and the two versions of ILE in Table 6.13 confirmed that the NILE algorithm always took longer times to reconstruct a Boolean function. These results are encouraging because they also agree with the findings in [Michalski and Larson, 1978] and [Reinke and Michalski, 1986] who have indicated that ILE approaches always deliver shorter CPU times than NILE approaches do. On the other hand, the small $p$-values shown in Table 6.13 for the comparison between the two versions of ILE reveals that the most generalizing clause criterion always required shorter CPU times than the least generalizing clause criterion.

<table>
<thead>
<tr>
<th>Class-pair</th>
<th>DOE-ZIPF</th>
<th>AP - DOE</th>
<th>WSJ-ZIPF</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>NILE$^1$</td>
<td>468.532</td>
<td>514.584</td>
<td>729.430</td>
<td>570.848</td>
</tr>
<tr>
<td>ILE (MGC)</td>
<td>150.896</td>
<td>156.375</td>
<td>142.966</td>
<td>150.088</td>
</tr>
<tr>
<td>ILE (LGC)</td>
<td>178.885</td>
<td>168.371</td>
<td>215.932</td>
<td>187.729</td>
</tr>
</tbody>
</table>

$^1$Numbers taken from the results of the Guided approach.

MGC and LGC stand for the Most Generalizing Clause and the Least Generalizing Clause, respectively.
Table 6.13. Data for the Statistical Comparison of the CPU Times Employed by the NILE and ILE approaches to Construct a Boolean Function.

<table>
<thead>
<tr>
<th></th>
<th>Number of Positive Signs for a Sign Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DOE vs. ZIPF</td>
</tr>
<tr>
<td>NILE - ILE(MGC)</td>
<td>269</td>
</tr>
<tr>
<td>NILE - ILE(LGC)</td>
<td>273</td>
</tr>
<tr>
<td>ILE(MGC) - ILE(LGC)</td>
<td>29†</td>
</tr>
</tbody>
</table>

MGC and LGC stand for the Most Generalizing Clause and the Least Generalizing Clause, respectively. The following p-value was obtained using the normal approximation $N(np, np(1-p))$ for all instances. $n = 260, 217, \text{ and } 247$ for all comparisons yielding zero in the above differences were discarded [Barnes, 1994]. (The total number of observations were 274)

*denotes a p-value close to 0.

It is important to notice here that although the two versions of ILE were faster than NILE, they incurred in some occasional expensive updates. These expensive processing times correspond to the “spikes” that run from the thick and thin lines to the thickest line (i.e., to the NILE approach). As indicated in Section 6.4.2, these long times correspond to the reconstruction of all the clauses of either Boolean function. More precisely, it was found that they occurred because all the clauses of the Boolean function had accepted a negative example. That is, this occurred when the condition $E^+(C) = E^-$ in Step 4 of Figure 4.2 was met. It is important to mention here that these expensive CPU times also agree with the literature regarding the reconstruction of decision trees (a tree may be represented in a DNF form), which indicates that “on the average the cost of an incremental update will be much lower than the cost of rebuilding the tree, despite an occasional expensive update” [Utgoff, 1997].
Figure 6.12. CPU times under the NILE and ILE approaches for the class-pair (DOE vs. ZIPF).

Figure 6.13. CPU times under the NILE and ILE approaches for the class-pair (WSJ vs. ZIPF).
6.4.4 Summary of the Computational Results Under the Incremental OCAT

The results of investigating the two versions of incremental OCAT are encouraging. First, from the computational results of the three experiments with the three class-pairs from the TIPSTER collection, it can be seen that the two versions of ILE outperformed the NILE approach as far as classification accuracy and CPU times are concerned. Regarding classification accuracy, the last column in Table 6.8 shows that the two versions of ILE required 281 and 299 training documents, respectively, to become more accurate, while the NILE approach required 336 observations. This represents a significant reduction in the number of training documents of 16.40% and 11.01%, respectively. Furthermore, the statistical test in Table 6.9 also confirm the superiority of ILE over the NILE approach.
The data on CPU times reported in Table 6.12 and Table 6.13 also confirm the dominance of the two versions of ILE over the NILE approach. For instance, the data in Table 6.12 show that ILE required only 26.3% and 32.9% of the time needed by non-incremental OCAT to classify all 510 documents in the experiments. This is an important result that suggests that the utilization of ILE approaches can deliver results in about 25% of the time the non-incremental OCAT will take.

On the other hand, the data in Table 6.10 and Table 6.11 proved that NILE always produced fewer clauses than ILE did. Despite these results, the computational experimentations conducted in this research suggest that occasional reconstructions of the classification rules reduced the number of clauses. This is an interesting situation that led us to believe that ILE will also render fewer number clauses if the “complexity” of the rules can be “guided” in order to promote their reconstruction. Of course, such reconstructions will be accompanied by some expensive CPU times that in the worst case scenario are bounded by a polynomial time.

Regarding the comparison of the two versions of ILE, the statistical test on the data on the last row of Table 6.9 suggests that it is not clear which of the two algorithms is the preferred one. Furthermore, the results in Tables 6.11 and 6.13 indicate that the two algorithms performed identically. These results were unexpected. However, they suggest that further investigation in the selection of the clause to be repaired is needed. One suggestion is, for example, to further investigate the most generalizing clause criterion because it showed slightly better results than the least generalizing criterion.
In conclusion, the results presented in this research suggest with high confidence that the utilization the incremental OCAT algorithm can outperform the VSM for the classification of large collections of text documents. This situation was evident during the investigation of the second problem because the results indicate that the VSM does not posses a capability to distinguish “Undecided” from “Incorrect” situations. Furthermore, the results of the second and third research problem indicate that the non-incremental OCAT approach can lead to an excellent classification accuracy by detecting “Undecided” situations. Moreover, the results of the fourth problem suggest that the incremental OCAT algorithm cannot only outperform the VSM, but also that it can deliver better and faster results than the non-incremental OCAT.
7 CONTRIBUTIONS OF THIS RESEARCH

The results presented in this dissertation are expected to make some substantial contributions in the classification of documents into two disjoint classes and in the area of inductive inference in general. In the classification of text documents, the statistical results presented here suggest that the OCAT algorithm outperformed the classification accuracy of the VSM, which is often used as a benchmarking methodology for these types of classification processes. More importantly, the OCAT algorithm delivers an output that matches the deterministic requirements stated in [DynMeridian, 1996] and [DOE, 1996] for this type of classification process. Furthermore, the OCAT algorithm may be an attractive solution to experts classifiers because its output can be easily converted into English words.

In the area of inductive inference, the methodologies presented in this dissertation are novel because this is the first time the OCAT algorithm is used for the classification of examples (i.e., documents in our case) by employing positive and negative rules. The experimentation with two rules demonstrated that the classification of examples can be done more confidently because now it is possible to determine whether or not the existing Boolean functions have enough knowledge to classify correctly unseen documents. This concept of positive and negative rules was also extended for the first time to the VSM, for which pairs of positive and negative centroids were constructed.

The computational results in Figures 6.3 through 6.5 also emphasize the importance of using two Boolean functions to guide their learning at a much faster rate. Actually, the results in these figures indicate that the utilization of "Undecided" situations to guide this learning was the main reason why the OCAT algorithm only analyzed 336 (66%) documents
in order to classify accurately all the 510 documents in the experiments. The utilization of these “Undecided” situations is also another contribution of this research because this is the first time that such situations are used to improve the existing classification knowledge of the two Boolean functions.

The most significant contribution of this research is the development of the incremental learning algorithms presented in Figures 4.2 and 4.3 that modified the OCAT approach (or IOCAT from now on when referring to incremental OCAT). IOCAT was designed to repair only one or more clauses of the Boolean function that causes an inconsistent classification. The computational results in Figures 6.12 through 6.14 indicate that by using IOCAT, the computational times were reduced to about 1/4 of the times required by the non-incremental OCAT algorithm. The implications of these results on the problem raised by [DynMeridian, 1996] and [DOE, 1996] are significant because they suggest that the IOCAT algorithm cannot only be capable of delivering accurate classifications, but it can also deliver such classifications at a much faster CPU times.

Another implication of this new incremental algorithm is that it can be extended to the B&B methodologies presented in [Triantaphyllou, 1994]. Such an extension will allow those B&B approaches to solve larger problems than the ones presented in the above reference.

Finally, it should be stated that the research conducted in this research has led to some presentations at the following professional and scientific conferences:


Furthermore, these results have lead to the following publications and working papers:


8 FUTURE RESEARCH

Although the results presented in this research are limited to a sample of almost 3,000 documents from the TIPSTER collection, they are encouraging because they suggest that a logical analysis approach via the IOCAT algorithm can be used to solve document classification problems (and potentially other classification problems) of larger dimensions. Therefore, a motivation to continue advancing this research may be focused on improving the incremental learning capabilities of the IOCAT algorithm. Some ideas for future investigation are presented next.

From Figures 6.6 through 6.8, it can be observed that the number of clauses constructed by the ILE can be huge. At the same time, it can be observed in Figures 6.9 to 6.11 that the Boolean functions became 100% accurate at a much faster rate. As indicated in Section 4, these results were obtained by incorporating only one example at a time into the training set. Hence, future investigation on this area may be focused on how to select and include more than one example with an "Undecided" classification.

A solution to this problem can be attractive by assuming that it may be economically sound to query an expert classifier with more than one document (i.e., training examples) per visit. Such a solution may be also attractive if documents are selected in such a way that an acceptable computational cost is still maintained. The selection of these documents can be guided by focusing more attention on:

1. The clause that accepts the most examples.
2. The clause with the least number of atoms.
3. The clause that rejects the most examples.
Another problem that requires a future investigation is as follows. Suppose that the expert classifier decides that the two Boolean functions have become very complex. That is, that they become too large. Hence, future research could focus in developing a methodology to determine when this complex situation arises in order to temporarily execute NILE approaches, which always produce a less complex set of classification rules.
9 SUMMARY

This research investigated a classification problem in which a document must be classified in one and only one of two classes. This type of classification is new in the sense that a document mis-classification could have severe consequences. It is also a new problem because now the classification of a document may be triggered by the presence of few sensitive words contained in the document’s text. As an example of the importance of this type of classification, one can consider the accidental release of documents to the public that may affect national security.

This new classification problem was tackled by investigating four closely related problems. The first problem was how to identify the set of text features (such as keywords) that best reflect the type of information contained in each document. The literature suggested that the “All Words Approach” was the recommended strategy to address this first problem. Some computations experiments with the “All Words Approach” indicated that the OCAT algorithm solved satisfactorily this first problem by using single-word patterns without compromising its classification accuracy.

The second problem addressed the question of how a computerized system can correctly infer the source class of text documents by constructing patterns of text keywords. This problem was tackled by using the One Clause At a Time (OCAT) algorithm on almost 3,000 documents from four document classes of the TIPSTER collection. These classes were documents by the Department of Energy (DOE), the Wall Street Journal (WSJ), the Associated Press (AP), and documents from the ZIPF class. Moreover, the OCAT algorithm was analyzed under the following two experimental conditions: (i) a leave-one-out cross
validation and (ii) a 30/30 cross validation (where 30 indicates the initial number of training documents from each document class). Furthermore, the performance of the OCAT algorithm was compared with the results by the VSM. The Vector Space Model (VSM) algorithm is often used to benchmark this type of classification processes. Some experimental results with the two experimental conditions suggest that the OCAT algorithm statistically outperformed the VSM when classifying documents into two disjoint classes.

The third problem was how to provide a guided learning methodology with the OCAT algorithm in order to improve its classification performance as more examples become available. Some computational results suggest that the classification efficiency of the OCAT algorithm can be improved if a guided learning approach is implemented. Actually, experiments on samples of 510 documents from the four classes from the TIPSTER collection indicate that the OCAT algorithm needed about 336 (i.e., 66%) training documents before it accurately classified all documents in the experiments.

The fourth problem addressed the problem of how to construct a Boolean function at a much faster rate. This problem was tackled by implementing an incremental learning approach. Computational results with the same samples of documents used for the third problem indicated that the average CPU times to construct the Boolean function was reduced to approximately 25% of the time needed by the non-incremental algorithm. However, these results indicated that the classification rules could also became more complex.

The literature showed that some existing methodologies, such as semantic analysis and neural networks, have been used for the classification of documents that share similar
content. However, the literature also suggested that these methodologies have been used in classification processes in which a misplacement of a document may not have grave consequences (e.g., it does not affect national security). Moreover, it showed that some of these methodologies are still limited to problems of small size either because of the time complexity of their algorithms or because they possess limited learning capabilities (see, for example, [Chen, 1996] and [Macleod, 1991]).
REFERENCES


49. MICHALSKI, R.S. and Larson, J.B., (1978), “Selection of the most representative training examples and incremental generation of VL1 hypotheses: The Underlying methodology and the description of the programs ESEL and AQ11 (Tech. Rep. No. UIUCDCS-R-78-867). University of Illinois at Urbana, Department of Computer Science, Urbana, IL, USA.


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


Salvador Nieto Sanchez was born in March 11, 1957 in Mexico City. In 1978, he received his first B.S. degree in Professional Teaching and in 1980 his second B.S. degree in Industrial Engineering from Centro Nacional de Enseñanza Técnica Industrial. In 1981, the British Council awarded him an eighteen-months scholarship to attend courses in Computerized Numerical Control Machine-tools in various schools and universities in the United Kingdom. In 1982, he returned to Mexico and he worked for Alcatel-Mexico occupying different managerial positions. From 1991 to 1993, he worked for Motorola-Mexico as Logistics and later as Materials Manager. From 1993 to 1998, he completed an M.S. in Industrial Engineering and an M.S. in Engineering Science in Louisiana State University. Currently, he works for General Instrument-Mexico as the Logistics and Inventory Planning Manager.
DOCTORAL EXAMINATION AND DISSERTATION REPORT

Candidate: Salvador Nieto Sanchez

Major Field: Engineering Science

Title of Dissertation: Classification of Text Documents Using a Logical Analysis Approach

Approved:

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

October 6, 1999