PERGAMON

Information Processing and Management xxx (2002) xxx–xxx

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

A feature mining based approach for the classification of text documents into disjoint classes

Salvador Nieto Sánchez [a], Evangelos Triantaphyllou [a,*], Donald Kraft [b]

[a] *Department of Industrial and Manufacturing Systems Engineering, 3128 CEBA Building,
Louisiana State University, Baton Rouge, LA 70803, USA*
[b] *Department of Computer Science, 286 Coates Hall, Louisiana State University, Baton Rouge,
LA 70803, USA*

Received 10 July 2000; accepted 7 September 2001

Abstract

This paper proposes a new approach for classifying text documents into two disjoint classes. The new approach is based on extracting patterns, in the form of two logical expressions, which are defined on various features (indexing terms) of the documents. The pattern extraction is aimed at providing descriptions (in the form of two logical expressions) of the two classes of positive and negative examples. This is achieved by means of a data mining approach, called One Clause At a Time (OCAT), which is based on mathematical logic. The application of a logic-based approach to text document classification is critical when one wishes to be able to justify why a particular document has been assigned to one class versus the other class. This situation occurs, for instance, in declassifying documents that have been previously considered important to national security and thus are currently being kept as secret. Some computational experiments have investigated the effectiveness of the OCAT-based approach and compared it to the well-known vector space model (VSM). These tests also have investigated finding the best indexing terms that could be used in making these classification decisions. The results of these computational experiments on a sample of 2897 text documents from the TIPSTER collection indicate that the first approach has many advantages over the VSM approach for solving this type of text document classification problem. Moreover, a guided strategy for the OCAT-based approach is presented for deciding which document one needs to consider next while building the training example sets. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Document classification; Document indexing; Vector space model; Data mining; One Clause At a Time (OCAT) algorithm; Machine learning

[*] Corresponding author. Tel.: +1-255-388-5372; fax: +1-255-388-5990.
E-mail addresses: snieto@gi.com (S. Nieto Sánchez), trianta@lsu.eduhttp://www.imse.lsu.edu/vangelis/ (E. Triantaphyllou), kraft@bit.csc.lsu.edu (D. Kraft).

PII: S0306-4573(01)00049-8

1. Introduction and background information

This paper investigates the problem of classifying text documents into two disjoint classes. Two sample sets of training examples (text documents) are assumed to be available. An approach is developed that uses indexing terms to form logical expressions (patterns) that next are used to classify unseen text documents. This is a typical case of supervised "crisp" classification.

A typical application of this type of classification problem occurs in the declassification process of vast amounts of documents originally produced by the US Federal Government. For reasons of national security, today there are huge numbers of documents that remain classified as secret. These documents are being kept in secured places because once they were considered to be important to national security. However, high maintenance costs and new laws dictate that these documents should be re-evaluated, and the ones that are not critical any more should become public. Thus, such documents need to be (roughly speaking) classified into the two disjoint categories of "*secret*" and "*non-secret*". In this context, when a document becomes "*non-secret*" after being "*secret*", it is termed as "*declassified*". In reality, documents have been classified into multiple levels of criticality to the national security, but in this study we will consider only two classes as described above. It should also be stated here that once a document becomes public (i.e., it has been declassified), then there is no way to make it secret again (especially now with the proliferation of the Internet).

In order to highlight the complexity of this kind of classification problem, consider the evaluation of the following three illustrative sentences: "*An atomic test is to be conducted at Site X*", "*An atomic test is to be conducted at 1:00 p.m.*", and "*An atomic test is to be conducted at Site X at 1:00 p.m.*" which come from three hypothetical documents, *A*, *B*, and *C*, respectively. According to (DynMeridian, 1996) and (DOE, 1995), only document *C* is both specific and sensitive and should not be declassified and instead should continue to be kept secret. The reason for this DOE (US Department of Energy) classification rule is because document *C* includes a sentence with specific reference to the "*place and time*" of an "*atomic test*". On the other hand, documents *A* and *B* can be declassified (assuming that the rest of their contents is not critical) and become available to the general public. In this illustrative example some key text features that can be used to characterize the two classes are references to "*place*", "*time*" and "*atomic test*".

Traditionally, this declassification process is carried out by employing vast numbers of human experts. However, the sheer amount of documents under consideration can make this process extremely ineffective and inefficient. Although there are guidelines of how to declassify secret documents, directly computerizing the human effort would require developing sophisticated parsers. Such parsers would have to analyze syntactically a document and then determine which, if any, guideline is applicable. The poor quality of the documents (many of which are decades old) and the complexity of the declassification guidelines could make such an approach too risky to national security. Thus, a reasonable alternative is to seek to employ machine-learning techniques. More specifically, techniques that use "learning from examples" approaches might be appropriate. Thus, this study is centered on the following three inter-related research problems:

1. Employ data mining techniques for extracting (mining) from two sets of examples text features that could be used to correctly group documents into two disjoint classes.
2. Use such features to form logical expressions (patterns) that could explain how the training examples are grouped together, and accurately classify unseen documents.

3. When considering a guided learning strategy for extracting the logical expressions, determine the next training document to include in the sets with the training examples so that accurate logical expressions are extracted as quickly as possible.

Since being able to justify this kind of classification decisions is important (given the severity of wrongly releasing a sensitive document to the public), methods that do not clearly allow for an explanation of the decision making process are not appealing here. Therefore, an impetus for this research is to seek to develop an approach that is based on mathematical logic, versus approaches that do not provide satisfactory explanation capabilities.

Traditional text classification and information retrieval (IR) techniques may have some limitations in solving this problem because they group documents that share a similar content. The prime example of such techniques is the vector space model (VSM) (Salton, 1989), which according to the literature (Buckley & Salton, 1995; Shaw, 1995) is the most effective methodology for this type of classification. The limitation of this technique is that it is based on similarity measures and thus it may not be able to distinguish between the previous three illustrative sentences in terms of the critical classification issues despite their all having similar contents. Other techniques, such as fuzzy set approaches (FSAs), neural networks (NNs), nearest neighbor, and computational semantic analysis (SA), have limitations in addressing these research problems, either because of their time complexity or because the resulting sizes of their outputs are still unacceptable and do not possess satisfactory explanatory capabilities (Chen, 1996; Scholtes, 1993).

An alternative approach to address the present research problems is the *One Clause At a Time* (*or OCAT*) algorithm (Triantaphyllou, 1994; Triantaphyllou, Soyster, & Kumara, 1994). This is a new data mining approach based on mathematical logic. This approach extracts (mines) key features from the training examples and next uses them to construct logical expressions (patterns) that can be used in classifying the training examples into the two original disjoint classes. These logical expressions can also easily be transformed into the IF-THEN type of decision rules. The OCAT approach applies to examples that can be represented by binary vectors (although attributes with continuous values can be transformed into ones with binary values (Triantaphyllou, 2001). However, this is not a limitation because it is the mere presence or absence of certain key words that can cause a document to be grouped in one class or another. On the other hand, the typical document classification done by traditional IR systems uses term frequencies (which are continuous numbers usually normalized in the interval $[0, 1]$) of keywords to group together documents of seemingly similar context.

This paper is organized as follows. Section 2 presents an overview of the document indexing process. Section 3 introduces the OCAT algorithm. Section 4 presents an overview of the VSM algorithm. Section 5 presents an overview of the guided learning approach (GLA). Section 6 describes the methodology of this investigation. Section 7 presents and summarizes the results. Finally the paper ends with some concluding remarks.

2. A brief overview of the document clustering process

The traditional process for automatic clustering of text documents results in a grouping of documents with similar content into meaningful groups in order to facilitate their storage and retrieval (see, for example, Salton, 1989). This is a four-step process as follows. In the first step a computerized system compiles a list of the unique words that co-occur in a sample of the documents from various classes (see, for example, Cleveland & Cleveland, 1983; Salton, 1989). In the second step, the co-occurring frequency of these words is analyzed and the best set of indexing terms is extracted. Usually, indexing terms (also known as *keywords* or *content descriptors*) are selected among the words with moderate frequency. The most *common* and the most *rare* words (i.e., the most frequent and infrequent words, respectively) are discarded as keywords because they convey little lexical meaning (see, for example, Cleveland & Cleveland, 1983; Fox, 1990; Luhn, 1957, 1958; Meadow, 1992; Salton, 1968; Zipff, 1949). Some examples of common words are: "a", "an", "and", and "the" (Fox, 1990); rare words depend on a document's subject domain (Meadow, 1992).

In the third step, a document is indexed by affixing it with the set of keywords that only occur in its text. According to Cleveland & Cleveland (1983), "this assignment is correct because authors usually repeat words that conform with the document's subject". A list (vector) of keywords represents the content of a document and usually it is referred to as a *document representation* or *surrogate*. An example of such a surrogate is the list of the seven words or phrases: {"*Document classification*", "*document indexing*", "*vector space model*", "*data mining*", "*OCAT algorithm*", and "*machine learning*"}. This surrogate could be used to represent the content of this paper, which is composed of thousands of words, symbols, and numbers. Hence, the goal of the third step is to construct a surrogate for representing the content of each document.

An advantage of using such surrogates is that they can be further simplified by expressing them as numerical vectors (Salton, 1989). One way to construct such vectors is by expressing their elements as binary values to indicate the presence (denoted by 1) or absence (denoted by 0) of certain keywords in a document. For instance, the vector's element $w_{ij} = 1$ (or 0) expresses the presence (or absence) of keyword T_i $(i = 1, 2, 3, \dots, t)$ in document D_j $(j = 1, 2, 3, \dots, N)$. Thus, the surrogate $D_j = [0\,1\,1\,1\,1\,0]$ of six binary elements indicates the presence of keywords (terms) T_2, T_3, T_4, and T_5 and the absence of keywords (terms) T_1 and T_6 in D_j.

Another way to construct these numerical vectors is by expressing (i.e., weighting) their elements using real values from the range $[0, 1]$. In this case, the value of an element w_{ij} indicates the relative occurrence frequency of a keyword within a document. For instance, a hypothetical surrogate such as $D_j = [\,0.00 \quad 1.00 \quad 0.10 \quad 0.75 \quad 0.90 \quad 1.00\,]$ may indicate that term T_3 occurs little, terms T_4 and T_5 occur moderately, and terms T_2 and T_6 occur with high frequency in D_j. In the remainder of this paper, however, only binary surrogates will be considered. As stated above the reason for considering binary vectors as surrogates is because the mere presence or absence of a keyword (or some pattern of keywords) may be detrimental in assigning a document to one of the two disjoint classes considered in this paper.

In the last step of the (traditional) classification process, documents sharing similar keywords (i.e., content) are grouped together. This classification follows from the pairwise comparison of all the surrogates (Salton, 1989).

3. An overview of the OCAT algorithm

The OCAT algorithm (Triantaphyllou, 1994; Triantaphyllou et al., 1994) is an inductive learning (data mining) approach for the classification of examples (documents in this study) into two disjoint classes. Each example is represented by a binary vector, although it can be generalized into vectors defined on continuous variables (Triantaphyllou, 2001). The *i*th element of such a vector represents the presence (1) or absence (0) of a key characteristic (also called a feature, variable, atom, or attribute) pertinent to the phenomenon under study. For this study, such binary features represent the presence or absence of the index terms (keywords) in the document surrogates. It is a good idea for the analyst to first try to define these examples by using as many features as possible.

The OCAT algorithm systematically identifies (“mines”) a small set of features while simultaneously constructing a logical expression of small size defined on these features that can be used to group together the training examples into the two original disjoint classes. This logical expression is a Boolean function that evaluates to true when it is given training examples from one of the two disjoint classes (the “positive” class) and false when it is given training examples from the other class (the “negative” class). The assignment of the terms “positive” and “negative” is arbitrary. Hopefully, if the training examples are representative enough, then this logical expression can be used to accurately classify unseen examples. Moreover, this logical expression can be used to extract new knowledge pertinent to the system under study. The extracted Boolean function is in conjunctive normal form (CNF) or in disjunctive normal form (DNF). Fig. 1 illustrates this algorithm for the CNF case.

From Fig. 1 it becomes evident that the OCAT algorithm is greedy in nature. This allows it to return logical expressions of small size. That is, it attempts to return a logical expression that is comprised of a small number of CNF (or DNF) clauses (Triantaphyllou & Soyster, 1996a). This algorithm is greedy in the sense that in the first iteration, it forms a clause that for the CNF case accepts (i.e., it evaluates to 1) all the examples in one of the classes (E^+) and rejects (i.e., it evaluates to 0) as many examples in the other class (E^-) as possible. Similarly, in the second iteration it forms another clause that again accepts all the examples in E^+ and rejects as many

```
Input:      $E^+$ and $E^-$ are the two disjoint sets with the “positive” and “negative” training
            examples, respectively.
Output:     A Boolean function in CNF (for this implementation) form.

      Begin
        $i = 0$;  $C = \varnothing$;   /* initializations */
        do while ($E^- \neq \varnothing$ )
Step 1:    Set $i \leftarrow i + 1$;  /* where $i$ 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 the CNF clause $C_i$;
Step 4:    Let $C \leftarrow C \chi C_i$;
Step 5:    Let $E^- \leftarrow E^- - E^-(C_i)$;
        repeat;
      End;
```

Fig. 1. The OCAT algorithm for CNF expressions.

examples as possible in E^- that were accepted by the previous clause(s). The algorithm repeats this process until all the examples in E^- are rejected by the generated sequence of clauses.

Triantaphyllou et al. (1994) and Triantaphyllou (1994) have implemented a branch-and-bound (B&B) approach to construct a set of clauses of minimal cardinality (i.e., a logical expression comprised by a minimum number of clauses) to solve the problem in *Step 2* of the OCAT algorithm in Fig. 1. This approach, however, is limited to problems of small to medium size because of the extensive CPU times it takes to find an optimal or next to optimal set of clauses (i.e., a set of clauses of minimal cardinality). In Triantaphyllou & Soyster (1995) the authors present a simple transformation approach that can be used to extract DNF functions from algorithms that infer CNF functions and vice versa.

More recently, Deshpande & Triantaphyllou (1998) have implemented a faster heuristic of quadratic time complexity as it is highlighted in Fig. 2. The version of the OCAT approach depicted in Fig. 2 returns a CNF expression (Boolean function) which accepts all of the positive examples while rejecting all of the negative examples. This algorithm can also be enhanced with a randomized approach which essentially solves this problem many times (in a randomized fashion) and at the end it selects the smallest (in terms of the number of clauses in the Boolean expression) of the extracted functions. This solution allows for larger classification problems and delivers a set of (CNF or DNF) clauses of small size (as opposed to the minimal or near to minimal size sets delivered by the B&B approach).

The notation $\mathrm{POS}(A_j)$ and $\mathrm{NEG}(A_j)$ in Fig. 2 denotes the number of positive and negative examples that will be accepted, respectively, if feature (atom) A_j is introduced in the current CNF clause. That is, $\mathrm{POS}(A_j) = |E^+(A_j)|$ and $\mathrm{NEG}(A_j) = |E^-(A_j)|$. Please note that Steps 1 and 2 in Fig. 2 also consider the negations (denoted as $\bar{A}_j$) of the features A_j. The ratio $\mathrm{POS}(A_j)/\mathrm{NEG}(A_j)$ in Steps 1 and 2 is used to quickly identify features that can form a clause that would accept all the

```
Input:    $E^+$ and $E^-$ are the two training sets as before defined on the binary atoms $A_j$
          (for $j = 1, 2, 3, ..., t$).
Output:   A Boolean function in CNF which accepts all positive and rejects all negative training
          examples.

Set $k = 1$ ;  Set $C = \varnothing$ ; /* initializations */
do while ($E^-$   $\varnothing$)
        Let $E^+$ be the original set of positive training examples ;
        Set $C_k = \varnothing$;   /* initialize the current clause */
        do while ($E^+$   $\varnothing$)
          Step 1:  Calculate the $POS(A_j)$ / $NEG(A_j)$ ratio for all features $A_j$ (and their negations);
          Step 2:  Choose a new feature $A_j$ according to $max[POS(A_j)$ / $NEG(A_j)]$ value.
          Step 3:  Let $C_k \leftarrow C_k \vee A_j$ ;
          Step 4:  Let $E^+(A_j)$ be the set of members of $E^+$ which are accepted when $A_j$ is
                   included in the current clause $C_k$ ;
          Step 5:  Let $E^+ \leftarrow E^+ - E^+(A_j)$ ;
        repeat;
        Step 6.  Let $E^-(C_k)$ be the set of members of $E^-$ which are rejected by $C_k$ ;
        Step 7:  Let $E^- \leftarrow E^- - E^-(C_k)$ ; Let $k \leftarrow k + 1$ ; Let $C \leftarrow C \wedge C_k$ ;
repeat;
```

Fig. 2. A fast heuristic for forming CNF clauses for the OCAT approach.

positive examples while rejecting many of the remaining negative ones (for the CNF case). By selecting a feature that maximizes this ratio, it is likely to have a feature that has a large $\mathrm{POS}(A_j)$ value and a low $\mathrm{NEG}(A_j)$ value. If the value of $\mathrm{NEG}(A_j)$ is equal to zero, then a large value is assigned to the ratio in Step 1 (in Fig. 2) in order to make the choice of including the feature (atom) A_j very appealing.

In order to help illustrate the previous issues consider the two sets of training examples depicted in Fig. 3. The set with the positive examples is comprised of four examples, while the set of the negative examples is comprised of six examples. These examples (document surrogates in our context) are defined on four binary features (i.e., index terms or atoms) or their negations. A value of 1 indicates the presence of the corresponding index term, while a value of 0 indicates the absence of the index term.

In the experiments described in this paper, the OCAT approach (as depicted in Fig. 1) employs the fast heuristic shown in Fig. 2. However, in other implementations a B&B approach (Triantaphyllou, 1994) may be used in Step 2. When the fast heuristic in Fig. 2 is employed, the first CNF clause that is derived is $(A_2 \vee A_4)$. To follow this, observe that the ratio with the maximum value is given by $\mathrm{POS}(A_2)/\mathrm{NEG}(A_2) = 2/2 = 1.00$ (this is a tie with the ratio that corresponds to A_4). When feature A_2 (which is arbitrarily selected over A_4) is included in the first clause (which initially is nil), then the first and the second positive examples will be accepted by that clause. The next iteration of the fast heuristic will consider the same negative examples, but the updated set of positive examples consists of the remaining positive examples (i.e., the third and fourth). This will result in the next feature to be selected being A_4. Now, the CNF clause that is comprised of these two features (i.e., A_2 and A_4) will accept all the positive examples in E^+ while rejecting the first, fourth and fifth examples from the set of the negative examples in E^-.

The next iteration of the OCAT approach (in Fig. 1) will consider the original four positive examples, but now the set of the negative examples consist of the ones not rejected so far. That is, the second, third, and sixth negative examples, in terms of the original set E^-. Working as above, it can be seen that the next application of the fast heuristic will return the clause : $(\bar{A}_2 \vee \bar{A}_3)$. The loop in Fig. 1 needs to be repeated once more, and a third (and final) clause is derived. That clause is $(A_1 \vee A_3 \vee \bar{A}_4)$. In other words, the logical expression (in CNF) which is derived from the training examples depicted in Fig. 3 is as follows:

$$(A_2 \vee A_4) \wedge (\bar{A}_2 \vee \bar{A}_3) \wedge (A_1 \vee A_3 \vee \bar{A}_4). \tag{1}$$

A fundamental property of expression (1) is that it accepts (i.e., evaluates to 1) all the examples in E^+, while it rejects (i.e., evaluates to 0) all the examples in E^-. However, since such an ex-

$$E^+ = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \qquad E^- = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Fig. 3. A set of four positive and a set of six negative examples.

pression is usually constructed from a relatively small collection of training examples, it is possible that the expression to be inaccurate when it classifies unseen examples. The error may occur if either the unseen example is positive, and the expression rejects it, or if the example is negative and the expression accepts it. More details on the OCAT approach can be found at: http://www.imse.lsu.edu/vangelis.

4. Improving the classification accuracy of the OCAT algorithm by using two logical expressions

Let us consider generating a second logical expression for classifying unseen examples. Please recall that the first expression is derived by treating the first set of training examples as positive and the second as the negative examples. However, the second expression is derived by treating the second set of training examples as positive and the first as the negative training examples. For instance, Fig. 4 depicts the same training examples as the ones in Fig. 3, but now they have reverse roles.

When the OCAT approach is applied on the new inference problem, the following expression (2) is derived:

$$(A_3 \vee \bar{A}_2) \wedge (\bar{A}_4 \vee A_2 \vee \bar{A}_1) \wedge (A_1 \vee \bar{A}_3). \tag{2}$$

As with expression (1), a property of the corresponding Boolean function $f(x) = (A_3 \vee \bar{A}_2) \wedge (\bar{A}_4 \vee A_2 \vee \bar{A}_1) \wedge (A_1 \vee \bar{A}_3)$ is to accept (i.e., to evaluate to 1) the former negative examples and to reject (i.e., to evaluate to 0) the former positive examples. For convenience, following the setting of the examples in Figs. 3 and 4, expression (1) will be called the *positive rule* (denoted as R^+) while expression (2) the *negative rule* (denoted as R^-).

The disadvantage of using only one rule (logical expression) can be overcome by considering the combined decisions of R^+ and R^- when classifying an unseen example e. If e is a positive example it will be denoted as e^+, while if it is a negative example it will be denoted as e^-. Under this setting, the classification of e can only be:

1. *Correct* if and only if:
   (a) $R^+(e^+) = 1$ and $R^-(e^+) = 0$;
   (b) $R^+(e^-) = 0$ and $R^-(e^-) = 1$.
2. *Incorrect* if and only if:
   (c) $R^+(e^+) = 0$ and $R^-(e^+) = 1$;
   (d) $R^+(e^-) = 1$ and $R^-(e^-) = 0$.

$$E^+ = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad E^- = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 4. The training example sets in reverse roles.

3. *Undecided* if and only if:
   (e) $R^+(e^+) = 1$ and $R^-(e^+) = 1$;
   (f) $R^+(e^-) = 1$ and $R^-(e^-) = 1$;
   (g) $R^+(e^+) = 0$ and $R^-(e^+) = 0$;
   (h) $R^+(e^-) = 0$ and $R^-(e^-) = 0$.

Cases (a) and (b) are called "correct" classifications because both rules perform according to the desired properties described above. However, as indicated above it is possible that the rules could incorrectly classify an example (cases (c) and (d)). Or the rules could simultaneously accept (cases (e) and (f)) or reject (cases (g) and (h)) the example. Cases (e)–(h) are called "undecided" because one of the rules does not possess enough classification knowledge, and thus such a rule must be reconstructed. Therefore, "undecided" situations open the path to improve the accuracy of a classification system. This paper exploits the presence of "undecided" situations in order to guide the reconstruction of the rule that triggered an erroneous classification decision.

5. An overview of the vector space model

The VSM is a mathematical model of an IR system that can also be used for the classification of text documents (Salton, 1989; Salton & Wong, 1975). It is often used as a benchmarking method when dealing with document retrieval and classification related problems. Fig. 5 illustrates a typical three-step strategy of the VSM approach to clustering.

To address *Step 1* Salton (1989) indicates that a suitable measure for pairwise comparing any two surrogates X and Y is the cosine coefficient (CC) as defined in Eq. (3) (other similarity measures are listed in Salton (1989, Chapter 10):

$$\mathrm{CC} = \frac{|X \cap Y|}{|X|^{1/2} \cdot |Y|^{1/2}}. \tag{3}$$

In this formula, $X = (x_1, x_2, x_3, \ldots, x_t)$ and $Y = (y_1, y_2, y_3, \ldots, y_t)$, where x_i indicates the presence (1) or absence (0) of the ith indexing term in X and similarly for y_i in respect to Y. Moreover, $|X| = |Y| = t$ is the number of indexing terms, and $|X \cap Y|$ is the number of indexing terms appearing simultaneously in X and Y. To be consistent with the utilization of binary surrogates, formula (3) provides the CC expression for the case of Boolean vectors. This coefficient measures the angle between two surrogates (Boolean vectors) in a Cartesian plane. Salton (1989) indicates that "the magnitude of this angle can be used to measure the *similarity* between any two docu-

Input: A sample of surrogates
Output: Clusters of documents and clusters' centroids

Step 1: Compute the pairwise similarity coefficients among all surrogates in the sample
Step 2 Cluster documents with sufficiently large pairwise similarities
Step 3 Compute the centroids of the clusters

Fig. 5. The VSM approach.

ments". This statement is based on the observation that two surrogates are identical, if and only if the angle between them is equal to 0°.

In *Step 2* the VSM clusters together documents that share a similar content based on their surrogates. According to Salton (1989), any clustering technique can be used to group documents with similar surrogates. A collection of clustering techniques is given in Anderberg (1973), Aldenderfer (1984), Späth (1985), and Van Rijsbergen (1979). However, it is important to mention here that with any of these techniques, the number of generated classes is always a function of some predefined parameters. This is in contrast with the requirements of our problem here in which the number of classes is exactly equal to two. When the VSM works under a predefined number of classes, it is said to perform a pseudo-classification. In this study the training examples are already grouped into two (disjoint) classes. Thus, the VSM is applied on the examples (documents) in each class and the corresponding centroids are derived. Hence, we will continue to use this kind of pseudo-classification.

To address *Step 3* Salton (1989), Salton & Wong (1975), and Van Rijsbergen (1979) suggest the computation of a class centroid to be done as follows. Let w_{rj} $(j = 1, 2, 3, \dots, t)$ be the jth element of the centroid for class C_r which contains q documents. Also, the surrogate for document D_i is defined as $\{D_{ij}\}$. Then, w_{rj} is computed as follows:

$$w_{rj} = (1/q)\sum_{i=1}^{q} D_{ij} \quad \text{for } j = 1, 2, 3, \dots, t. \tag{4}$$

That is, the centroid for class C_r is also a surrogate (also known as the "average" document) defined on t keywords.

Finally, the VSM classifies a new document by comparing (i.e., computing the CC) its surrogate against the centroids that were created in *Step 3*. A new document will be placed in the class for which the CC value is maximum.

In the tests to be described later in this paper, the VSM is applied on the documents (training examples) available for each class. In this way, the centroid of each one of the two classes is derived. For instance, consider the training examples depicted in Figs. 3 and 4. The VSM is now applied on these data. The centroids in expression (5) have been constructed from the data in Fig. 3 and the centroids in expression (6) from the data in Fig. 4. Obviously, the centroids for the second set are in reverse order of those for the first set of data.

$$\begin{aligned} C_+ &= [1/2, 1/2, 1/4, 1/2], \\ C_- &= [2/3, 1/3, 1/2, 1/3], \end{aligned} \tag{5}$$

$$\begin{aligned} \mathbb{C}_+ &= [2/3, 1/3, 1/2, 1/3], \\ \mathbb{C}_- &= [1/2, 1/2, 1/4, 1/2]. \end{aligned} \tag{6}$$

The notation C_+ and C_- stands for the centroids for the data in Fig. 3, and $\mathbb{C}_+$ and $\mathbb{C}_-$ stand for the centroids for the data in Fig. 4, respectively. In order to match the names of the positive and negative rules described for the OCAT algorithm, the two centroids for the data in Fig. 3 will be called the *positive centroids* while the centroids for the data in Fig. 4 will be called the *negative centroids*. As with the OCAT algorithm, the utilization of two sets of centroids has been investigated in order to tackle the new classification problem by using the VSM as new examples become available.

6. Guided learning for the classification of text documents

The central idea of the GLA can be illustrated as follows. Suppose that the collection to be classified contains millions of documents. Also, suppose that an oracle (i.e., an expert classifier) is queried in order to classify a small sample of examples (documents) into classes E^+ and E^-. Next, suppose that the OCAT algorithm is used to construct the positive and negative rules, such as was the case with expressions (1) and (2). As indicated earlier, these rules may be inaccurate when classifying examples not included in the training set, and therefore they will result in one of the classification outputs provided in cases (a)–(g), as described earlier. One way to improve the classification accuracy of these rules is to add one more document to the training set (either in E^+ or E^-) and have them reconstructed. Therefore, the question GLA attempts to answer is: *What is the next document to be inspected by the expert classifier so that the classification performance of the derived 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 call this the RANDOM input learning strategy. A drawback of this strategy may occur if the oracle and *incumbent* rules frequently classify a document in the same class. If this occurs frequently, then the utilization of the oracle and the addition of the example to the training set is of no benefit. An alternative and more efficient way to provide the expert with a document is to select one in an "undecided" situation. This strategy (in a general form) was first introduced in Triantaphyllou & Soyster (1996b). This approach appears to be a more efficient way of selecting the document because an "undecided" situation implies that one of the rules misclassified the document. Therefore, the expert's verdict will not only guide the reconstruction of the rule that triggered the misclassification, but it may also improve the learning rate of the two rules. We call this the GUIDED input learning strategy. An incremental learning version of the OCAT approach is described in Nieto Sanchez, Triantaphyllou, Chen, & Liao (2001).

7. Experimental data

In order to determine the classification performance of the OCAT approach in addressing this new problem, the OCAT's classification accuracy was compared with that of the VSM. Both approaches were tested under three experimental settings:

1. a Leave-One-Out Cross-Validation (or CV) (also known as the Round-Robin test);
2. a 30/30 Cross-Validation (or 30CV), where 30 stands for the number of training documents in each class; and
3. in an experimental setting in which the OCAT algorithm was studied under a random and a guided learning strategy.

These will be defined below. This multiple testing strategy was selected in order to gain a more comprehensive assessment of the effectiveness of the various methods.

For these tests, a sample of 2897 documents was randomly selected from four document classes of the TIPSTER collection (Harman, 1995; Voorhees, 1998). The previous numbers of documents in each class were determined based on memory limitations on the computing platform used (an IBM Pentium II PC running Windows 95). The TIPSTER collection is a standard data set for experimentation with IR systems. The four document classes were as follows:

1. Department of Energy (DOE) documents,
2. Wall Street Journal (WSJ) documents,
3. Associated Press (AP) documents, and
4. ZIPFF class documents.

We chose documents from this collection because for security reasons we did not have access to actual secret DOE documents.

Table 1 shows the number of documents that were used in the experimentation. These documents were randomly extracted from the four classes of the TIPSTER collection.

We simulated two mutually exclusive classes by forming the following five class-pairs: (DOE vs. AP), (DOE vs. WSJ), (DOE vs. ZIPFF), (AP vs. WSJ), and (WSJ vs. ZIPFF). These five class-pairs were randomly selected from all possible class-pairs combinations. To comply with the notation presented in the previous sections, the first class of each class-pair was denoted as E^+, while the second class was denoted as E^-. Thus, we try to find a Boolean expression to classify a document surrogate into the proper TIPSTER class.

Table 2 shows the average number of keywords that were extracted from the five class-pairs mentioned above. The data in this table can be interpreted as follows. For the class-pair (DOE vs. AP), the average number of keywords used in all the experiments was 511 under the CV validation and 803 under the 30CV validation. A similar interpretation applies to the data in the other columns.

It should be stated at this point that we used a number of alternative indexing terms. Besides single words, we also used sequences of two words at a time, sequences of three words at a time, and sequences of four words at a time. However, some pilot studies indicated that the best results

Table 1
Number of documents randomly extracted from each class

Class:	DOE	AP	WSJ	ZIPFF	Total
Number of documents:	1407	336	624	530	2897

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

Table 2
Average number of indexing words used in each experiment

Type of experiment	DOE vs. AP	DOE vs. WSJ	DOE vs. ZIPFF	AP vs. WSJ	WSJ vs. ZIPFF
CV	511	605	479	448	501
30CV	803	911	890	814	811

In order to keep the size reasonable for our computing environment, only the first hundred words from each document were considered. Stop words were always removed.

would be derivable by using as indexing terms single words only. Thus, the atoms (binary variables) in the derived logical expressions are single keywords and not sequences of them.

8. Testing methodology

This section first summarizes the methodology for the Leave-One-Out Cross-Validation and the 30/30 Cross-Validation. These two alternative testing methods have been employed in order to gain a better understanding of the various procedures used to classify text documents. The same section also presents the statistical tests employed to determine the relative performance of the VSM and the OCAT algorithm. This section ends with the methodology for the GLA.

8.1. The Leave-One-Out Cross-Validation

The cross-validation (CV) testing was implemented on samples of 60 documents as follows. First, 30 documents from each class were randomly selected. Please note that the size 60 was used due to storage limitations in our computing environment. Then, one document was removed from these sets of documents with its class noted. After that, the *positive* and *negative rules* under the OCAT approach and the *positive* and *negative centroids* under the VSM were constructed using the remaining 59 documents. In the third step the class of the document left out was inferred by both algorithms. Then, the correctness of the classification was determined according to the cases (a)–(h), as defined in Section 4. The previous second and third steps were repeated with different sets of training examples until all 60 documents had their class inferred one at a time. This experimental setting was replicated 10 times with different subsets of the training data, at which point the results of the two algorithms were tested for statistical differences.

8.2. The 30/30 Cross-Validation

The 30/30 Cross-Validation (30CV) was implemented on samples of 254 documents as follows. The number of 254 documents was used to avoid excessive computational time. Initially, the *positive* and *negative rules* under the OCAT approach and *positive* and *negative centroids* under the VSM were constructed by using only 30 documents (randomly selected) from each class. Then, the classification of the remaining 194 documents was inferred. As before, the correctness of this classification was determined according to the cases (a)–(g), as defined earlier. As with the first experimental setting, the 30CV validation was replicated 10 times, at which point the results of the two algorithms were tested for statistical difference.

8.3. Statistical performance of both algorithms

To determine the statistical performance of both algorithms, the following hypotheses were tested. The first test was needed to determine the relative dominance of the algorithms. In the second test we implemented a sign test in order to determine the consistency of the dominance of the algorithms.

1. $H_0 : \bar{P}_{\mathrm{OCAT}} \leqslant \bar{P}_{\mathrm{VSM}}$
   $H_1 : \bar{P}_{\mathrm{OCAT}} \leqslant \bar{P}_{\mathrm{VSM}}$
2. $H_0 : p = 0.50$
   $H_1 : p \neq 0.50$

where P_{OCAT} and P_{VSM} are the numbers of documents with “correct” classification under the two algorithms divided by the total number of documents in the experiment. In addition, p is the probability of finding an equal number of positive and negative differences in a set of outcomes. More on how these tests were performed is provided in the following sub-sections which present the computational results.

8.4. Experimental setting for the GLA

Consider the question: *What is the best next document to be given to the oracle in order to improve the performance of the classification rule*? Three samples of 510 documents (255 from each class) from the three class-pairs: (DOE vs. ZIPFF), (AP. vs. DOE), and (WSJ vs. ZIPFF) were used. The number of 510 documents was determined by the available RAM memory on the Windows PC we used. The previous three class-pairs were processed by the OCAT algorithm (only) under the RANDOM and the GUIDED learning approaches.

These two learning approaches were implemented as follows. At first, 30 documents from each class in the experiment were randomly selected, and the positive and negative rules (logical expressions) were constructed. Next, the class membership of all 510 documents in the experiment was inferred based on the two classification rules. The criteria expressed as cases (a)–(g) in Section 4 were used to determine the number of “correct”, “incorrect”, and “undecided” classifications. Next, a document was added to the initial training sample as follows. For the case of the RANDOM approach, this document was selected at random from among the documents not included in the training sets yet (i.e., neither in E^+ nor in E^-).

In contrast, under the GUIDED approach this document was selected from the set of documents which the positive and negative rules had already termed as “undecided”. However, if the two rules did not detect an “undecided” case, then the GUIDED approach was replaced by the RANDOM approach until a new “undecided” case was identified. This process for the RANDOM and GUIDED approaches was repeated until all 510 documents were included in the two training sets E^+ and E^-. The results of this experimentation are presented next.

9. Results for the Leave-One-Out and the 30/30 Cross-Validations

Table 3 summarizes the experimental results for the CV validation, while Table 4 summarizes the results for the 30CV validation. The abbreviations “C:”, “I:”, and “U:” in the first column of both tables correspond to the “correct”, “incorrect”, and “undecided” classification outcomes which can be obtained by using the positive and the negative rules (for the OCAT case) or the positive and the negative centroids (for the VSM case). For instance, the data in Table 3, column 2

Table 3
Summary of the first experimental setting: Leave-One-Out Cross-Validation

	DOE vs. AP		DOE vs. WSJ		DOE vs. ZIPFF		AP vs. WSJ		WSJ vs. ZIPPF		TOTAL	
	VSM	OCAT	VSM	OCAT	VSM	OCAT	VSM	OCAT	VSM	OCAT	VSM	OCAT
C:	334	410	286	296	280	358	316	365	286	303	1502	1732
I:	261	5	314	66	320	25	284	47	314	76	1493	219
U:	5	185	0	238	0	217	0	188	0	221	5	1049

Table 4
Summary of the second experimental setting: 30/30 Cross-Validation

	DOE vs. AP		DOE vs. WSJ		DOE vs. ZIPFF		AP vs. WSJ		WSJ vs. ZIPPF		TOTAL	
	VSM	OCAT	VSM	OCAT	VSM	OCAT	VSM	OCAT	VSM	OCAT	VSM	OCAT
C:	975	1406	975	1266	1035	1320	1088	1283	1088	1145	5161	6420
I:	975	70	975	134	915	124	846	140	837	176	4548	644
U:	0	474	0	550	0	506	16	527	25	41	41	2686

(i.e., class-pair (DOE vs. AP)) indicate that the VSM identified 334 "correct", 261 "incorrect", and 5 "undecided" cases.

Similarly, the data in Table 3, column 3 (i.e., class-pair (DOE vs. AP)) indicate that the OCAT algorithm identified 410 "correct", 5 "incorrect", and 185 "undecided" classifications. The data in the other columns can be interpreted in a similar manner. The last two columns of these two tables summarize the results across all five class-pairs. Fig. 6 compares the proportions of these results for both algorithms.

Two key observations can be derived from the size of the dark areas (or areas of "undecided" classifications) in Fig. 6 which was derived from Tables 3 and 4. First, it can be observed that the proportion of "undecided" cases detected by the VSM algorithm is almost 0%. These have occurred when the two positive and the two negative centroids accepted the same document and, therefore, the classes predicted by both sets of centroids have to be selected randomly. More specifically, these "undecided" instances occurred even when these randomly predicted classes were identical. The VSM was implemented using the CC coefficient, following the suggestions in Salton (1989).

In contrast, as the second observation, we have large proportions of "undecided" classifications with the OCAT algorithm. Please recall that this type of classification decision is desired because such cases demonstrate that either the positive or the negative rules are unable to classify correctly new documents. Therefore, in these results the large dark areas in the above figure show that both rules were unable to classify correctly a large proportion of the documents in the experiments. More importantly, the size of these areas indicates that positive or negative rules may be improved if they are modified when an "undecided" situation is detected.

Consider the proportion of the "incorrect" classifications (i.e., the white areas in Fig. 6). One can derive two conclusions. First, the number of "incorrect" classifications the VSM made amounts to 49.77% (or 1493/3000) with the CV validation and to 46.65% (or 4548/9750) with the 30CV validation. These large proportions of "incorrect" classifications can be attributed to the

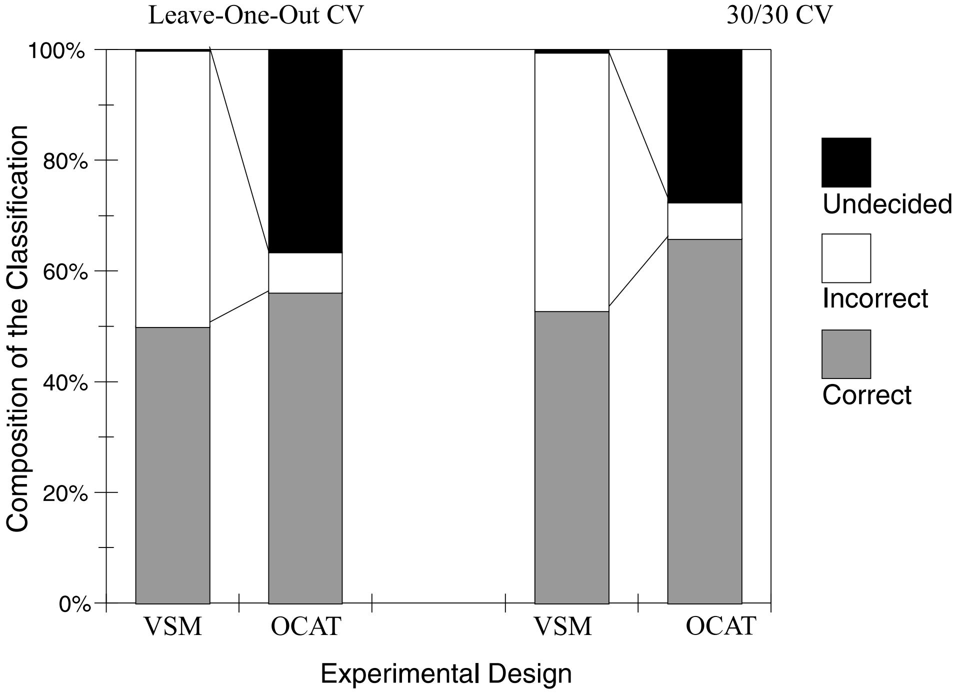

Fig. 6. Comparison of the classification decisions under the OCAT and VSM approaches.

inability of the positive and negative centroids to distinguish between "incorrect" and "undecided" classifications. Second we have 7.30% (or 219/3000) and 6.61% (or 644/9750) of "incorrect" classifications the OCAT algorithm made with the two test settings. In this case, these small error rates can be attributed to the utilization of positive and negative rules of small size that enabled the OCAT algorithm to distinguish between the "incorrect" and "undecided" classifications.

Despite the disparate proportions of the "inaccurate" and "undecided" classifications for both of these algorithms, their performances were statistically compared using *only* the proportions with the "correct" classifications. That is, the undecided cases were not considered here. In this way the VSM approach was not placed in an unfair setting when it was compared with the OCAT approach. This comparison was needed in order to determine which algorithm better addressed

Table 5
Statistical difference in the classification accuracy of the VSM and OCAT approaches

Type of experiment	P_{OCAT}[a]	P_{VSM}[b]	$P_{\mathrm{VSM}} - P_{\mathrm{OCAT}}$	Binomial test	
				Half-length[c]	Interval
CV	0.577	0.501	−0.0760	0.025	(−0.035, −0.085)
30CV	0.658	0.529	−0.1287	0.014	(−11.47, −14.27)

[a] 1732/*n* and 6420/*n*; where *n* is 3000 for CV and 9750 for 30CV.
[b] 1502/*n* and 5161/*n*; where *n* is 3000 for CV and 9750 for 30CV.
[c] Denotes that both approaches performed statistically differently.

Table 6
Data for sign test to determine the consistency in the ranking of the OCAT and VSM approaches

	Type of experiment	
	CV	30CV
Number of "+" signs	4	7
Number of "−" signs	46	43
$p\text{-value} = \sum_{i=0}^{m} \binom{50}{i} \cdot p^i \cdot (1-p)^{50-i}$,	p-value $= 2.23 \times 10^{-10}$	p-value $= 1.04 \times 10^{-7}$
where $m = 4$ for CV and 7 for 30CV and $p = 0.50$		

the classification problem studied in this paper. For this comparison, it was assumed that no additional improvement of the two algorithms was possible under the CV and 30CV cross-validations. Furthermore, the "incorrect" and "undecided" outcomes were considered as incorrect classifications.

The results of these tests (as shown in Table 5) indicate that the OCAT approach is more accurate in both types of computational experiments than the VSM. Furthermore, the very low p-

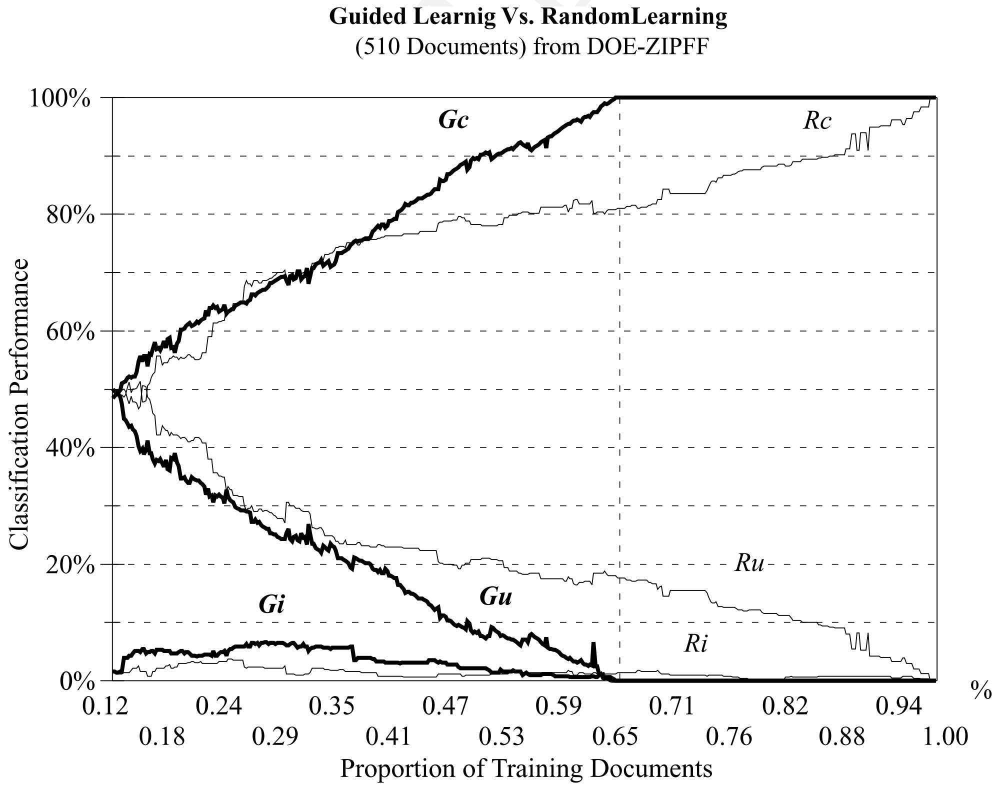

Fig. 7. Results when the GUIDED and RANDOM approaches were used on the (DOE vs. ZIPFF) class-pair.

values in Table 6 indicate that it is extremely unlikely to find a similar number of positive and negative differences in the proportions of the "correct" classifications under the two approaches (Barnes, 1994). Therefore, the results of these two statistical tests indicate that the OCAT approach is better suited to address the document classification problem studied in this paper.

10. Results for the GLA

Figs. 7–9 show the results of the OCAT algorithm under the RANDOM and GUIDED input learning approaches. The horizontal axis indicates the percentage of training documents used during the experiment. For example, at the beginning of the experiment there were 60 training documents or 11.76% of the 510 documents in the experiment. Next, when one more document was added to the training set, following the recommendation of the GUIDED and RANDOM approaches, there were 12.16% of the documents in the experiment.

The vertical axis shows the proportions of "correct", "incorrect", and "undecided" classification for the various percentages of training documents used in the experiment. The abbreviations Rc, Ri, Ru and Gc, Gi, Gu stand for the proportions of "correct", "incorrect", and "undecided" outcomes for the RANDOM and GUIDED approaches, respectively. For instance,

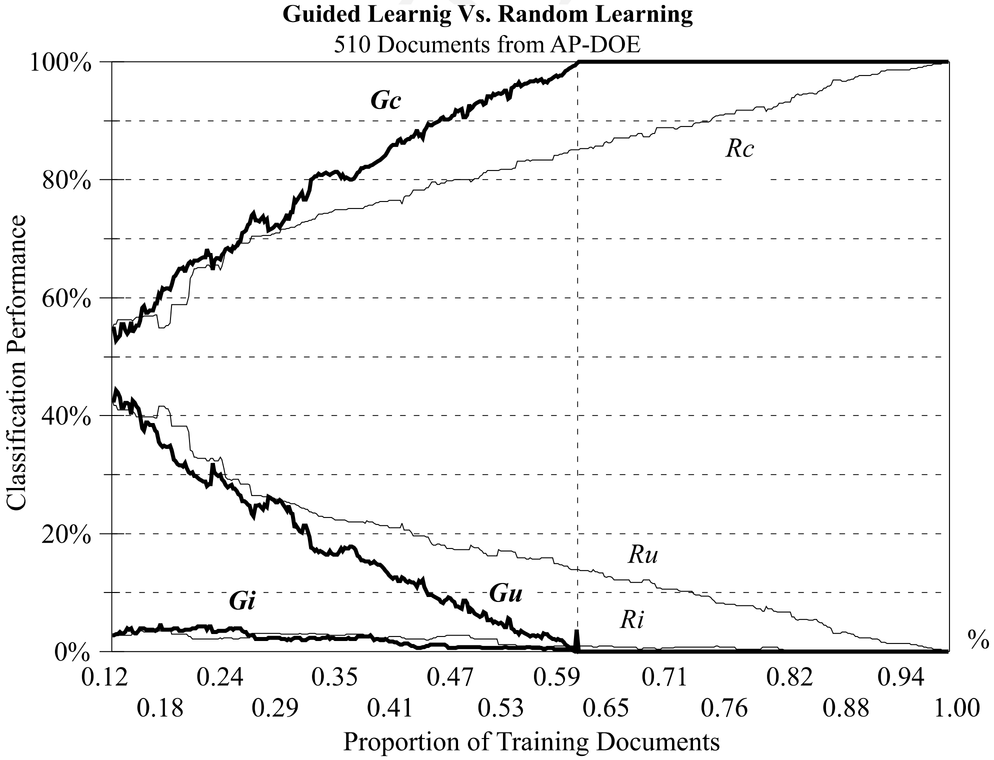

Fig. 8. Results when the GUIDED and RANDOM approaches were used on the (AP vs. DOE) class-pair.

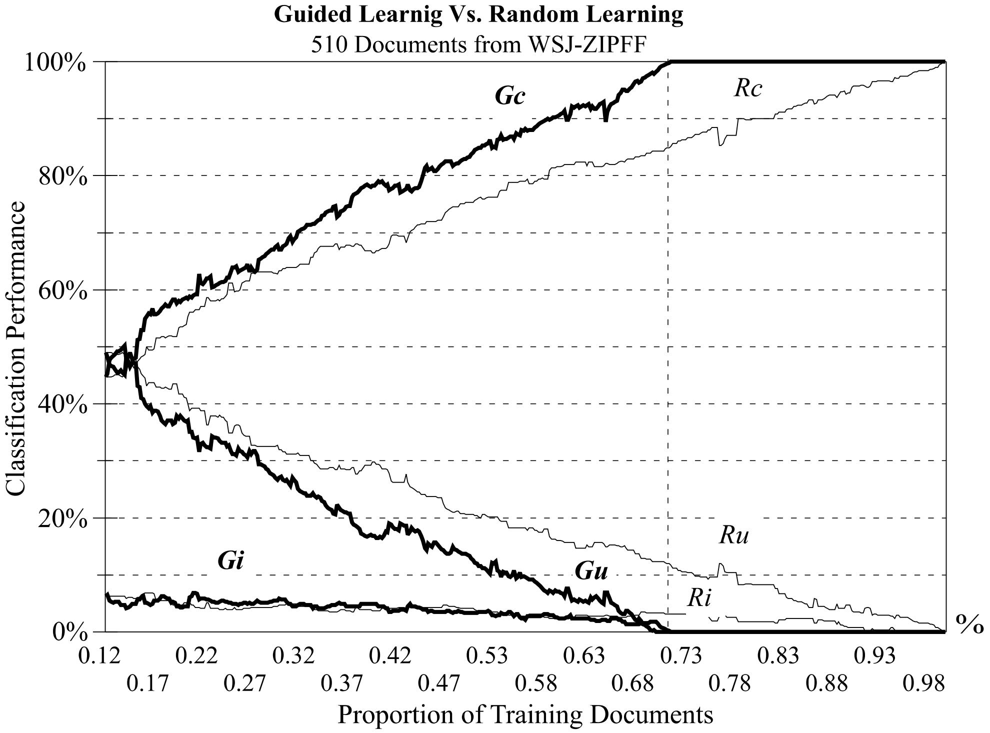

Fig. 9. Results when the GUIDED and RANDOM approaches were used on the (WSJ vs. ZIPFF) class-pair.

Rc is the proportion of “Correct” classifications under the RANDOM approach, and Gu is the proportion of “Undecided” classification under the GUIDED approach.

Table 7 shows the percentage of training documents the OCAT algorithm needed before it classified all 510 documents in each class-pair correctly (i.e., it became 100% accurate). The position of the dotted line in the above three figures corresponds to the percentages shown in this table. For instance, in Fig. 7 (class-pair (DOE vs. ZIPFF)) this line is at 65.69% on the horizontal axis.

Some important observations can be made regarding the proportions of “correct”, “incorrect”, and “undecided” classifications in Figs. 7–9. First, the rate of “correct” classifications under the GUIDED approach, Gc, was higher than the rate Rc under the RANDOM approach. Actually,

Table 7
Percentage of documents from the population that were inspected by the oracle before an accuracy of 100% was reached

Class-pairs	% Under GUIDED	% Under RANDOM
(DOE vs. ZIPFF)	65.69	100.00
(AP vs. DOE)	60.98	99.80
(WSJ vs. ZIPFF)	71.18	99.80
Average	65.95	99.87

100% accuracy was achieved when the number of “Incorrect” and “Undecided” classifications were 0%.

the last row in Table 7 indicates that the OCAT algorithm needed on the average about 34% less training documents to classify correctly all 510 document under the GUIDED approach than under the RANDOM approach.

These results are very interesting for a number of reasons. They confirm the assumption stated in Section 5 which indicated that the utilization of documents with the "undecided" classification could increase the accuracy of the OCAT algorithm. These results are also encouraging because they help to answer the second question stated in the introduction of this section. That is, queries to the oracle should stop when about 66% of the 510 documents from the three class-pairs of the TIPSTER collection had been inspected and were included in the training sets. More importantly, these results are important because they suggest that the OCAT algorithm can be employed for the classification of large collections of text documents.

The other two observations are related to the rates at which the "incorrect" and "undecided" classifications were eliminated. It can be observed from the previous three figures that these rates were a direct consequence of improving the classification rules. The figures show that the rates Gi and Gu reach 0% when about 66% (336 documents) of the 510 documents in the experiment have been included in E^+ and E^-. On the other hand, it can be seen that under the RANDOM learning approach, the rates Ri and Ru reached 0% when 99.8% (509 documents) of the documents are processed.

11. Concluding remarks

This paper has examined a classification problem in which a document must be classified into one of two disjoint classes. As an example of the importance of this type of classification, one can consider the possible release to the public of documents that may affect national security. The method proposed in this paper (being an automatic method) is not infallible. This is also true because its performance depends on how representative the training examples (documents) are. The application of such an approach to a problem of critical importance (such as the one highlighted in the introduction) can be seen as an important and useful automatic tool for a preliminary selection of the documents to be classified.

We considered an approach to this problem based on the VSM algorithm and compared it with an algorithm which is based on mathematical logic, called the OCAT algorithm. We tested these two approaches on almost 3000 documents from the four document classes of the TIPSTER collection: Department of Energy (DOE), Wall Street Journal (WSJ), Associated Press (AP), and the ZIPFF class. Furthermore, these documents were analyzed under two types of experimental settings: (i) 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). The experimental results suggest that the OCAT algorithm performed significantly better in classifying documents into two disjoint classes than the VSM.

Moreover, the results of a third experiment suggested that the classification efficiency of the OCAT algorithm can be improved substantially if a GLA is implemented. Actually, experiments on samples of 510 documents from the previous four classes of the TIPSTER collection indicated that the OCAT algorithm needed only about 336 (i.e., 66% of the) training documents before it correctly classified all of the documents.

The results presented here, although limited to a relatively small collection of almost 3000 documents, are encouraging because they suggest that the OCAT algorithm can be used in the classification of larger collections of documents.

Acknowledgements

The authors are very appreciative for the very thoughtful and constructive comments made by two anonymous referees of an earlier version of the paper. Those comments played an important role in improving the context and presentation of this paper. The first and second authors wish to thank the US Department of Energy for providing the financial funds that supported this research. The first author is pleased to recognize the partial financial support from Consejo Nacional de Ciencia y Technologia-Mexico. The second author is also very appreciative to the US Navy, Office of Naval Research, for the support from grants N00014-95-1-0639 and N00014-97-1-0632.

References

Aldenderfer, M. S. (1984). *Cluster analysis*. Beverly-Hill, CA: Sage.

Anderberg, M. R. (1973). *Cluster analysis for applications*. New York: Academic Publishers.

Barnes, J. W. (1994). *Statistical analysis for engineers and scientists, a computer-based approach*. New York: McGraw-Hill.

Buckley, C., & Salton, G. (1995). Optimization of relevance feedback weights. In *Proceedings of SIGIR 1995* (pp. 351–357).

Chen, H. (1996). *Machine learning approach to document retrieval: An overview and an experiment*. Technical Report, University of Arizona, MIS Department, Tucson, AZ, USA.

Cleveland, D., & Cleveland, A. D. (1983). *Introduction to indexing and abstracting*. Littleton, CO: Libraries Unlimited.

Deshpande, A. S., & Triantaphyllou, E. (1998). A greedy randomized adaptive search procedure (GRASP) for inferring logical clauses from examples in polynomial time and some extensions. *Mathematical and Computer Modelling*, *27*(1), 75–99.

DOE (1995). *General Course on Classification/Declassification, Student Syllabus, Handouts, and Practical Exercises*. US Department of Energy, Germantown, MD, USA.

DynMeridian (1996). *Declassification Productivity Initiative Study Report*. DynCorp Company, Report Prepared for the US Department of Energy, Germantown, MD, USA.

Fox, C. (1990). A stop list for general text. *ACM Special Interest Group on Information Retrieval*, *24*(1–2), 19–35.

Harman, D. (1995). Overview of the second text retrieval conference (TREC-2). *Information Processing and Management*, *31*(3), 271–289.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, *5*(3), 155–165.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, *4*(4), 600–605.

Meadow, C. T. (1992). *Text information retrieval systems*. San Diego, CA: Academic Press.

Nieto Sanchez, S., Triantaphyllou, E., Chen, J., & Liao, T. W. (2001). An incremental learning algorithm for constructing Boolean functions from positive and negative examples. *Computers and Operations Research* (in press).

Salton, G. (1968). *Automatic information organization and retrieval*. New York: McGraw-Hill.

Salton, G., & Wong, A. (1975). A vector space model for automatic indexing. *Information retrieval and language processing*, *18*(11), 613–620.

Salton, G. (1989). *Automatic text processing. The transformation analysis, and retrieval of information by computer*. Reading, MA: Addison-Wesley.

Scholtes, J. C. (1993). *Neural networks in natural language processing and information retrieval*. The Netherlands: North-Holland.

Späth, H. (1985). *Cluster dissection and analysis: theory, Fortran programs, and examples*. Chichester, UK: Ellis Harwood.

Shaw, W. M. (1995). Term-relevance computations and perfect retrieval performance. *Information Processing and Management*, *31*(4), 312–321.

Triantaphyllou, E. (2001). *The OCAT approach for data mining and knowledge discovery*, Working Paper, IMSE Department, Louisiana State University, Baton Rouge, LA 70803-6409, USA.

Triantaphyllou, E., & Soyster, A. L. (1996a). On the minimum number of logical clauses inferred from examples. *Computers and Operations Research*, *23*(8), 783–799.

Triantaphyllou, E., & Soyster, A. L. (1996b). An approach to guided learning of Boolean functions. *Mathematical Computing Modeling*, *23*(3), 69–86.

Triantaphyllou, E., & Soyster, A. L. (1995). A relationship between CNF and DNF systems which are derived from the same positive and negative examples. *ORSA Journal on Computing*, *7*(3), 283–285.

Triantaphyllou, E., Soyster, A. L., & Kumara, S. R. T. (1994). Generating logical expressions from positive and negative examples via a branch-and-bound approach. *Computers and Operations Research*, *21*(2), 185–197.

Triantaphyllou, E. (1994). Inference of a minimum size Boolean function from examples by using a new efficient branch-and-bound approach. *Journal of Global Optimization*, *5*, 64–94.

Van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). London: Butterworths.

Voorhees, E. (1998). Overview of the sixth text retrieval conference (TREC-6). In *Proceedings of the sixth text retrieval conference (TREC-6), Gaithersburg, MD, USA* (pp. 1–27).

Zipff, H. P. (1949). *Human behavior and the principle of least effort*. Menlo Park, CA: Addison-Wesley.