2012 International Conference on Advanced Computer Science Applications and Technologies

Hybrid Intelligent Techniques for Text Categorization

Dr. Ahmed T. Sadiq
Department of Computer Science
University of Technology
Baghdad, Iraq
e-mail: drahmaed_tark@yahoo.com

Sura Mahmood Abdullah
Informatics Institute for Postgraduate Studies
Iraqi Commission for Computers and Informatics
Baghdad, Iraq
e-mail: sulovera181184@yahoo.com

***Abstract*— Text categorization is the task in which documents are classified into one or more of predefined categories based on their contents. This paper shows that the proposed system consists of three main steps: document representation, classifier construction and performance evaluation. In the first step, a set of pre-classified documents is provided. Input documents are initially pre-processed in order to be split into features and eliminate non-informative features. The remaining features are next weighted based on the frequency of each feature in that document and standardized by reducing a feature to its root using the stemming process. Due to the large number of features even after the non-informative features removal and the stemming process, the proposed system applies specific thresholds to extract distinct features which represent the input document. In the second step, the text categorization model (classifier) is built by learning the distinct features which represent all the pre-classified documents; this process can be achieved by using one of the supervised classification techniques that is called the rough set theory. The model uses a pair of precise concepts from the above theory that are called lower and upper approximations to classify any test document into one or more of main categories and sub-categories. In the final step, the performance of the proposed system is evaluated. It has achieved good results up to 96%, when applied to a number of test documents for each sub-category of main categories.**

Keywords- Text Mining; Text Categorization; Rough Set Theory.

I. INTRODUCTION

The enormous amount of information stored in unstructured texts cannot simply be used for further processing by computers, which typically handle text as simple sequences of character strings. Therefore, specific (pre-) processing methods and algorithms are required in order to extract useful patterns. Text mining refers generally to the process of extracting interesting information and knowledge from unstructured text [1].

The main goal of text mining is to enable users to extract information from textual resources and deals with the operations like, information retrieval, categorization and summarization. The most important part of text mining is the text categorization [2].

Text Categorization (TC), also known as *text classification* or *topic spotting* [3], is the automatic classification of text documents under predefined categories. Information Retrieval (IR) and Machine Learning (ML) techniques are used to assign words which are also called (features, tokens, terms or attributes) to the documents which are also called (examples or instances) and classify them into specific categories. Machine learning helps to categorize the documents automatically. Information Retrieval helps to represent the text as a feature.

Manually organizing large document bases is extremely difficult, time consuming, error prone, expensive and is often not feasible, which results are dependent on variations experts judgments [4].

There are mainly two types of approaches to text categorization. One is the *rule-based approach*. In the rule-based approach, the classification rules are manually created usually by experts in the domain of the texts. Although the rule-based approach can achieve high accuracy, it is costly in terms of labor and time. The second approach involves *machine learning techniques*, in which classification rules are automatically created using information from labeled (already-categorized) texts. Machine learning is cost-saving because it requires only labeled texts [5].

Section two of this paper shows the related works and section three explains text categorization, while rough set theory is explained in section four. Section five presents the proposed system and finally section six shows the conclusions.

II. RELATED WORKS

1) In [3], **Ruiz M.** focused on the use of hierarchical classification structures, such as the Yahoo hierarchy of topics, to build and train machine learning algorithms for text categorization. For this purpose, Hierarchical Mixtures of Experts (HME) model is adapted for text categorization. HME based on the "*divide and conquer*" principle in which a large problem is divided into many smaller, easier to solve problems whose solutions can be combined to yield a solution to the complex problem. The HME model was also evaluated using neural networks, and linear classifiers (Rocchio, Widrow-Hoff (WH) and Exponentiated-Gradient (EG)) as the nodes of the hierarchy.

2) In [6], **Nigam K.** demonstrated that supervised learning algorithms that use a small number of classified documents and many inexpensive unclassified documents can create high-accuracy text classifiers. Then an algorithm is introduced for learning from classified and unclassified documents based on the combination of Expectation-

978-0-7695-4959-0/13 $25.00 © 2013 IEEE
DOI 10.1109/ACSAT.2012.50

Maximization (EM) and a Naive Bayes probabilistic classifier.

3) In [7], **Lee K**. describes the development of supervised and semi-supervised learning approaches to similarity-based text categorization systems. Supervised approaches to text categorization usually require a large number of training documents to achieve a high level of effectiveness. His goal was to develop a text categorization system that uses fewer classified documents for training to achieve a given level of performance. A new similarity-based learning algorithm which is called Keyword Association Network (KAN) and thresholding strategies (*RinSCut* variants) were described to achieve his goal. KAN was designed to give appropriate weights to features according to their semantic content and importance by using their co-occurrence information and the discriminating power values for similarity computation. RinSCut (rank-in-score) was designed to combine the strengths of two common thresholding strategies, rank-based (RCut) and score-based (SCut). The thresholding strategies can be applied to the similarity-based learning algorithms as well as similarity-based text processing tasks.

4) In [8], **Ifrim G**. proposed a model to text categorization that concentrates on the underlying meaning of words in their context (i.e., concentrates on learning the *meaning* of words, identifying and distinguishing between different contexts of word usage). This model can be summarized in the following steps:

a) Map each word in a text document to explicit concepts.

b) Learn classification rules using the newly acquired information.

c) Interleave the two steps using a latent variable model.

The proposed model combines Natural Language Processing techniques such as word sense disambiguation, part of speech tagging, with statistical learning techniques such as Naïve Bayes in order to improve classification accuracy and to achieve robustness with respect to language variations.

5) In [9], **Radhi A.** designed a system which was achieved by the following steps:

a) Extracting concepts from text printed in natural language using machine learning approach and finding the embedded relations between concepts using Inductive Logic Programming (ILP) to have a clear schema defining the entities and hierarchal relations in the interesting domain.

b) Classifying a set of different documents based on machine learning techniques; a general inductive process automatically builds a classifier by learning from a set of pre-classified documents. The advantages of this approach are its very good effectiveness, considerable savings in terms of expert labor power and straightforward portability to different domains.

6) In [4], **Karamcheti A.** implemented two categorization engines for text categorization based on Naive Bayes and k-Nearest Neighbor methodology. Then he compared the effectiveness of these two engines by calculating standard precision and recall for a collection of documents. The compared results show that the k-Nearest Neighbor categorization engine is better than Naïve Bayes engine.

III. Text Categorization

Text Categorization is the process of assigning a given text to one or more categories. This process is considered as a supervised classification technique, since a set of pre-classified documents is provided as a training set. The goal of TC is to assign a category to a new document.

TC can play an important role in a wide variety of areas such as information retrieval, word sense disambiguation, topic detection and tracking, web pages classification, as well as any application requiring document organization [10]. The following points represent the text categorization applications [8, 11, 12, 13, 14, 15]:

- Automatic Indexing.
- Document Organization.
- Document Filtering.
- Word Sense Disambiguation.
- Hierarchical Web Page Categorization.

Text categorization has many types, the difference between these types are as follows.

A. Single-Label versus Multilabel Text Categorization

The case in which exactly one category is assigned to the input text is called *single-label* text categorization, whereas the case in which multiple categories can be assigned to the input text is called *multi-label* text categorization [15,16].

B. Document-Pivoted versus Category-Pivoted Categorization

Usually, the classifiers are used in the following way: Given a document, the classifier finds all categories to which the document belongs. This is called a *document-pivoted categorization*. Alternatively, the classifier finds all documents that should be filed under a given category. This is called a *category-pivoted categorization* [15].

C. Soft versus Hard Text Categorization

Hard categorization means a complete automated categorization system makes a binary decision on each document-category pair, while soft categorization means ranking the input documents or the output categories by the order of relevance, instead of making explicit assignment decision [15,12].

IV. Rough Set Theory

Rough set theory was developed by Zdzislaw Pawlak, in the early 1980's. It deals with the classificatory analysis of data tables. The data can be acquired from measurements or from human experts. The main goal of the rough set analysis is to synthesize approximation of concepts from the acquired data [17].

The rough set approach seems to be of fundamental importance to artificial intelligence and cognitive sciences, especially in the areas of machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, inductive reasoning and pattern recognition [18].

Rough set theory has close connections with many other theories such as fuzzy sets, statistic methods, genetic algorithms etc. Despite its connections with other theories, the rough set theory may be considered as an independent discipline [19].

The starting point of rough set theory which is based on data analysis is a data set which is represented as a table, where each row represents an object. Every column represents an attribute that can be measured for each object; this table is called an *information system*. More formally, it is a pair S = (U, A), where U is a nonempty finite set of *objects* called the *universe* and A is a nonempty finite set of *attributes* such that $a : U \rightarrow V_a$ for every $a \in A$. The set V_a is called the *value set of a*. then with any $B \subseteq A$ there is associated an equivalence relation $IND_A(B)$:

$$IND_A(B) = \{(x, y) \in U^2 \mid \forall a \in B \; a(x) = a(y)\} \quad (1)$$

$IND_A(B)$ is called the *B-indiscernibility relation*. If $(x, y) \in IND_A(B)$, then objects x and y are indiscernible from each other by attributes from B. The equivalence classes of the B-indiscernibility relation are denoted $[x]_B$ [17].

Assigning to every subset $X \subseteq U$ two sets B*(X) and B*(X) called the *B-lower* and the *B-upper approximations* of X, respectively, and defined as follows:

$$B_*(X) = \{ x \mid [x]_B \subseteq X\} \quad (2)$$

and

$$B^*(X) = \{ x \mid [x]_B \cap X \neq \phi\} \quad (3)$$

Hence, the B-lower approximation of a set is the union of all B-granules that are included in the set, whereas the B-upper approximation of a set is the union of all B-granules that have a nonempty intersection with the set.

The set

$$BN_B(X) = B^*(X) - B_*(X) \quad (4)$$

will be referred to as the B-boundary region of X. If the boundary region of X is the empty set, i.e., $BN_B(X) = \emptyset$, then the set X is *crisp* (exact) with respect to B; in the opposite case, i.e., if $BN_B(X) \neq \emptyset$, the set X is referred to as *rough* (inexact) with respect to B [20].

A rough set X can be also characterized numerically by the following coefficient:

$$\alpha_B(X) = |B_*(X)| / |B^*(X)| \quad (5)$$

Called the *accuracy of approximation*, where |X| denotes the cardinality of $X \neq \phi$. Obviously $0 \leq \alpha_B(X) \leq 1$. If $\alpha_B(X) = 1$, X is *crisp* with respect to B (X is precise with respect to B), and otherwise, if $\alpha_B(X) < 1$, X is *rough* with respect to B (X is vague with respect to B) [19].

V. The Proposed System

The proposed system can be summarized in three main steps that are integrated to give accurate results: document representation, classifier construction and performance evaluation. In the first step, after reading the input document by the proposed system which divides that document into features which are also called (tokens, words, terms or attributes), it removes the non-informative features (stop words, numbers and special characters) and represents that document in a vector space as a vector whose components are the remaining features and their weights which are computed by the frequency of each feature in that document. Those features are next standardized by reducing them to their root using the stemming process.

In spite of the non-informative features removal and the stemming process, the dimensionality of the feature space may still be too high. So the proposed system applies specific thresholds to reduce the size of the feature space for each input document based on the frequency of each feature in that document.

In the second step, the proposed system performs the learning and testing processes. In the first process, the classifier is built by observing the features of sub-categories for each main category from the training set, this process can be done using one of the supervised classification techniques that is called the rough set theory. In the second process, the classifier applies a pair of precise concepts from the rough set theory that are called the lower and upper approximations to classify the input document from the testing set into one or more of main categories and sub-categories.

In the final step, the performance of the proposed system can be measured by computing its efficiency and its effectiveness. The proposed system framework is shown in Fig.1. The details of the main steps for the proposed system framework are in the following sections:

A. Text Document

Document collection is divided into two sets: Training set and Test set. The former indicates to pre-classified set of documents which is used for training the classifier, while the latter determines the accuracy of the classifier based on the count of correct and incorrect classifications for each document in that set which is classified by the classifier into suitable main categories and sub-categories.

The training set with 280 documents was distributed in 3 main categories (Computer Science, Mathematics and Physics) and a number of sub-categories which belong to those main categories such as Computer Science includes 4 sub-categories (Artificial Intelligence, Database, Image Processing and Security), Mathematics includes 3 sub-categories (Algebra, Numerical Analysis and Statistics) and Physics includes 2 sub-categories (Laser and Materials).

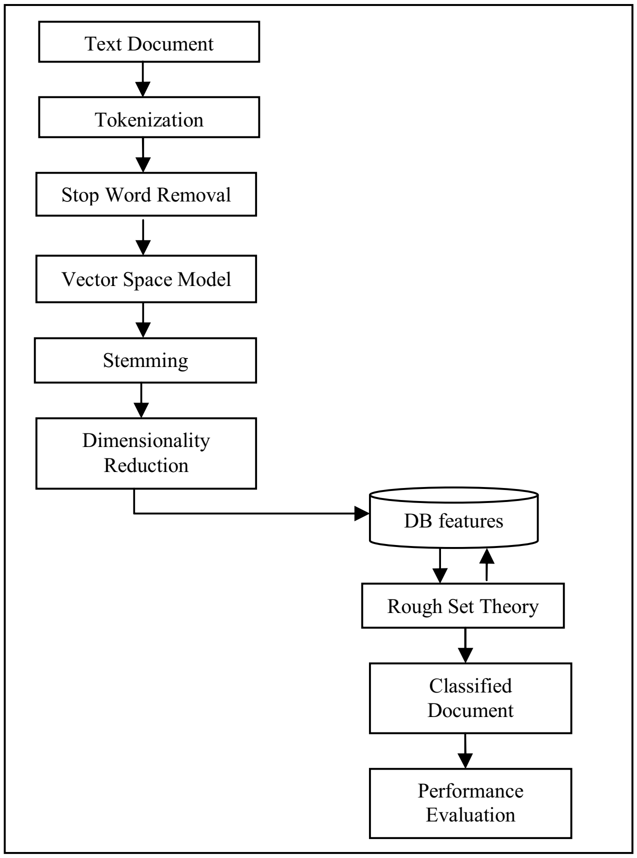

Figure 1. The Proposed Text Categorization System Framework.

B. Tokenization

Each input document is partitioned into a list of features which are also called (tokens, words, terms or attributes).

C. Stop Words Removal

A stop list is a list of commonly repeated features which appear in every document. The common features such as pronouns he, she, it and conjunctions such as and, or, but etc. need to be removed because they do not have effect on the categorization process (i.e., each feature should be removed when it matches any feature in the stop words list). For the same reason, if the feature is a special character or a number then that feature should be removed.

D. Vector space model

Each input document is represented as a vector in a vector space, each dimension of this space represents a single feature of that vector and its weight which is computed by the frequency of occurrence of each feature in that document. This representation is called vector space model. In this step, each feature is given an initial weight equal to 1.

E. Stemming

Stemming is the process of removing affixes (prefixes and suffixes) from features. This process is used to reduce the number of features in the feature space and improve the performance of the classifier when the different forms of features are stemmed into a single feature.

For example: **(convert, converts, converted, converting)**

From the above example, the set of features is conflated into a single feature by removal of the different suffixes -s, -ed, -ing to get the single feature **convert**.

There are different types of the stemming algorithms; some of them can produce incomplete stems which don't have meaning.

One of the most common stemming algorithms uses a set of rules to remove suffixes from features, this process continues until none of the rules apply. This algorithm has some drawbacks such as it is not limited to produce feature stems, for example, "revival" becomes "reviv". And it does not deal with prefixes completely, so "relevant" and "irrelevant" remain as unrelated features.

Another type of the stemming algorithms has a large set of suffixes. This type gives priority to the longest suffix which exists in the set of suffixes; the suffixes can be removed by applying a set of rules. The main drawbacks of this type are that it is time consuming, and many suffixes are not available in the set of suffixes.

The proposed system implements the stemming process by applying a set of rules in specific way. The rules of the stemming process are:

- All prefixes are removed from features, if the prefix exists in features.
- The stemming process uses a lexicon to find the root for each irregular feature. Where the lexicon has four irregular tables (irregular verb, irregular noun, irregular adjective and irregular adverb), each table has some fields which represent conjugate of each feature such as the irregular verb table has (the verb root, past, past participle, present participle and plural) fields. If the feature matches any feature in the fields of the irregular tables then that feature should be converted to its stem (root) form which exists in the first field of each irregular table.
- The similar features in size and characters are conflated under a single feature. The weight of a single feature results from summing the frequencies of the conflated features.
- When the only difference among the similar features in the first characters is (-s, -d, -es, -ed, -ly, -er, -ar, -ing, -ance, -ence, -tion, -sion or any other suffixes), then those features are conflated under the shortest one among them. The weight of the shortest feature results from summing the frequencies of the conflated features.

F. Dimensionality Reduction

Even after the non-informative features removal and the stemming process, the number of features in the feature space may still be too large. Among those features, some features may be unuseful to the categorization task and sometimes decrease accuracy. Such features can be removed without affecting the classifier performance.

The large number of features may affect the classifier learning because most machine learning algorithms which are implemented on text categorization cannot deal with this huge number of features.

Dimensionality reduction of the feature space can be done by feature selection and feature extraction.

Dimensionality reduction by feature selection deals with several methods for features selection. These methods are applied to reduce the size of the full feature set. Most feature selection methods suffer from time-consuming which is considered a critical problem in text categorization system.

Dimensionality reduction by feature extraction is to create a small set of artificial features from original feature set. The main cause for using artificial features is the problems of polysemy, homonymy and synonymy; the words may not be the optimal features.

For the above reasons, the proposed system does not use the dimensionality reduction by feature selection or feature extraction; it uses specific thresholds (10%, 8%, 6% and 4%) to reduce the number of features in the feature space (i.e., selecting features which derived from the stemming process, whose frequencies equal or larger than 10%, 8%, 6% or 4% of the number of derived features from the stemming process) and in the learning process, it stores the resulted features which represent the input document in one of the database tables which represent the sub-categories for each main category based on the sub-category for that document and stores the weights of the resulted features under any frequency field (i.e., Freq. >= 10%, 8%, 6% or 4%) of that database table based on the frequency of those features in the input document. But in the testing process, it stores the resulted features in a list which contains all features that represent the input document.

G. *The Learning \ Testing Algorithm for Text Categorization - Rough Set Theory*

A classifier can be built by learning the features which represent all the training documents that are specified for each sub-category of main categories, after that the classifier becomes ready to classify any test document into a suitable main category and sub-category based on the content of that document.

Rough set theory has been successfully applied to machine learning. This theory is a supervised classification technique, because the categories of the training documents are already known in advance.

A pair of precise concepts from the rough set theory that are called the lower and upper approximations have been used to classify the test documents into one or more of main categories and sub-categories. When the test document is given to the trained classifier; it should predict the correct main category and sub-category for that document.

The testing set with 100 documents was categorized into 4 categories and a number of sub-categories which belong to the first three categories (Computer Science, Mathematics and Physics) such as Computer Science includes 4 sub-categories (Artificial Intelligence, Database, Image Processing, Security), Mathematics includes 3 sub-categories (Algebra, Numerical Analysis, Statistics) and Physics includes 2 sub-categories (Laser, Materials). The proposed system does not only deal with documents in those categories, but it also deals with any document in any topic.

All the steps that were applied to the training documents should be applied to the test documents such as tokenization, stop words removal, vector space model, stemming and dimensionality reduction.

After applying all the previous steps to the test documents, a list of distinct features which represents the test document is obtained.

1) Upper approximation: It is the intersection between the features which represent the test document and the features in any database table, that have a frequency under any frequency field (i.e., Freq. >= 10%, 8%, 6% or 4%) of that database table which represent the sub-category of main categories. The resulted features represent a set of upper approximation features.

2) Lower approximation: It is the intersection between the resulted features from the upper approximation and the features in only one database table, that have a frequency under any frequency field (i.e., Freq. >= 10%, 8%, 6% or 4%) of that database table which represents the sub-category of main categories. The resulted features represent a set of lower approximation features.

The difference between the upper and lower approximations for the set of features is called the boundary region for that set.

The accuracy of approximation can be measured by computing the ratio between the lower and upper approximations for the set of features which represents the test document, when the accuracy value equals to 1 then the above set of features is called crisp, but when the accuracy value less than 1 then the above set of features is called rough.

H. *Classified Document*

After applying all the steps to represent the test documents (i.e., to convert the test documents into compact representation of their contents) and implementing the lower and upper approximations concepts from the rough set theory to them, the classifier should predict the correct main categories and sub-categories for those documents.

I. *Performance Evaluation for a Classifier*

The performance of the proposed system can be measured by calculating its efficiency (i.e. average time required to build a classifier from a set of the training documents and average time required to classify any test document by the classifier) and its effectiveness (i.e. the classifier ability to give the correct classification). The learning time for building the classifier is shown in Fig.2. The average of the testing time for classifying of the test documents is shown in Fig.3.

There are many metrics to evaluate the effectiveness of the proposed system. The most common are accuracy, error rate, precision and recall [21]. The results of calculating Precision and Recall are shown in Table 1.

1) Accuracy(Ac): Is the ratio between the number of documents which were correctly categorized and the total number of documents.

$$Ac_i = TP_i + TN_i / TP_i + TN_i + FP_i + FN_i \qquad (6)$$

Where TPi (true positives) is the number of documents correctly classified in category ci, TNi (true negatives) is the number of documents correctly classified as not belonging to category ci, FPi (false positives) is the number of documents incorrectly classified in category ci, and FNi (false negatives) is the number of documents incorrectly classified as not belonging to category ci [21].

2) Error rate (E): Is the ratio between the number of documents which were not correctly categorized and the total number of documents.

$$E_i = 1 - Ac_i \qquad (7)$$

3) Precision (P): Is the percentage of correctly categorized documents among all documents that were assigned to the category by the classifier.

$$P_i = TP_i / TP_i + FP_i \qquad (8)$$

4) Recall (R): Is the percentage of correctly categorized documents among all documents belonging to that category.

$$R_i = TP_i / TP_i + FN_i \qquad (9)$$

The performance evaluation for each category is shown in Fig. 4, Fig. 5 and Fig. 6.

The below Algorithm illustrates the main behavior of the proposed system.

Input: *D_1, $D_{2,}$....D_m (Different Documents), C_1, $C_{2,}$....C_n (specific categories)*
Output: *Classified Document*
Begin
For *each category C_i* ***Do***
    For *each Document D_j for C_i* ***Do***
        Split D_j into features $\Rightarrow F_j$
        Remove stop words, number and special characters from $F_j \Rightarrow T_j$
        Give frequency equal to 1 for $T_j \Rightarrow Ftr_j$, Ftr_freq_j
        Make stemming and some morphology processing for Ftr_j and increase frequency for $Ftr\_freq_j \Rightarrow Short\_Ftr_j$, $short\_Ftr\_Freq_j$
        Make Dimensionality Reduction for $Short\_Ftr_j \Rightarrow DR_j$
        Add DR_j in DB (database) for C_i
    End For
Compute Upper Approximation for C_i using the following equation
$$B^*(X) = \{ x \mid [x]_B \cap X \neq \phi\}$$
Compute Lower Approximation for C_i using the following equation
$$B_*(X)= \{ x \mid [x]_B \subseteq X\}$$
Compute ratio between Upper Approximation for C_i and DR_j for D_j, the max ratio represent the correct category for D_j
Compute accuracy for C_i using the following equation
$$\alpha_B(X) = |B_*(X)| / |B^*(X)|$$
End For
End

VI. CONCLUSION

The text categorization model (classifier) and learning algorithm cannot directly process the documents in their original form, so each input document should be converted into compact representation of its content by using the preprocessing steps which include (tokenization, stop words removal, vector space model, stemming and dimensionality reduction).

The stemming process can be implemented by applying a set of rules in specific way instead of chopping off the characters blindly and producing stems that don't have meaning. So the similar features are conflated under a single feature, but when the only difference among the similar features in the first characters is (-s, -d, -es, -ed, -ly, -er, -ar, -ing, -ance, -ence, -tion, -sion or any other suffixes) then those features are conflated under the shortest one among them. The weights of the single feature and shortest feature result from summing the frequencies of the conflated features under them.

The size of the feature space is not reduced by implementing the feature selection methods such as (mutual information, information gain, etc.) because these methods reduce the size of the full feature set and are time consuming. So the proposed system implements specific thresholds (10%, 8%, 6% and 4%) to reduce the size of the feature space for each input document based on the frequency of each feature in that document, the resulted features represent that input document.

The rough set theory is a supervised classification technique; it is used for building the text categorization model by learning the properties of a set of pre-classified documents for each sub-category of main categories. Thereafter, the model uses a pair of precise concepts from the rough set theory that are called the lower and upper approximations to classify any test document into one or more of main categories and sub-categories, because the system deals not only with the main categories, but also with a number of sub-categories for each main category.

When the rough set theory concepts are used in the proposed system, the results of the system reach to 96% when it is applied to a number of test documents for each sub-category of main categories.

The proposed system computes for each test document in the set of the test documents the testing time based on the size of each document in that set. Thereafter, it computes the average of that time for all test documents, which ranges from 5 to 14 Sec.

The proposed system computes for each sub-category from the training documents the training time based on the size of each document in that sub-category. The above time ranges from 555 to 862 Sec. for all sub-categories.

VII. REFERENCES

[1] Hotho A., Nürnberger A. and Paaß G., "A Brief Survey of Text Mining", LDV Forum - GLDV Journal for Computational Linguistics and Language Technology, Vol. 20, No. 1, pp. 19-62, May 2005.

[2] Korde V., Mahender C., "Text Classification and Classifiers: A Survey", International Journal of Artificial

Intelligence & Applications (IJAIA), Vol. 3, No. 2, pp. 85-99, March 2012.

[3] Ruiz M., "Combining Machine Learning and Hierarchical Structures for Text Categorization", PhD Thesis, Computer Science Dept., University of Iowa, Iowa City, Iowa, USA, December 2001.

[4] Karamcheti A., "A Comparative Study on Text Categorization", M.Sc Thesis, University of Nevada, Las Vegas, May 2010.

[5] Takamura H., "Clustering Approaches to Text Categorization", PhD Thesis, Information Processing Dept., Graduate School of Information Science, Nara Institute of Science and Technology, Japan, March 2003.

[6] Nigam K., "Using Unlabeled Data to Improve Text Classification", PhD Thesis, School of Computer Science, Carnegie Mellon University, USA, May 2001.

[7] Lee K., "Text Categorization with a Small Number of Labeled Training Examples", PhD Thesis, School of Information Technologies, University of Sydney, Australia, September 2003.

[8] Ifrim G., "A Bayesian Learning Approach to Concept-Based Document Classification", M.Sc Thesis, Computer Science Dept., Saarland University, Saarbrücken, Germany, February 2005.

[9] Radhi A., "Machine Learning for Text Categorization", PhD Thesis, Computer Science Dept., University of Technology, Baghdad, Iraq, June 2006.

[10] Wanas N., Said D., Hegazy N. and Darwish N., "A Study of Local and Global Thresholding Techniques in Text Categorization", proceedings of the 5th Australasian Data Mining Conference (AusDM06), Sydney, Australia, Vol. 61, pp. 91-101, December 2006.

[11] Granitzer M., "Hierarchical Text Classification Using Methods from Machine Learning", M.Sc Thesis, Institute of Theoretical Computer Science (IGI), Graz University of Technology, Austria, October 2003.

[12] Addis A., "Study and Development of Novel Techniques for Hierarchical Text Categorization", PhD Thesis, Electrical and Electronic Engineering Dept., University of Cagliari, Italy, March 2010.

[13] Sebastiani F., "A Tutorial on Automated Text Categorization", proceedings of the 1st Argentinian Symposium on Artificial Intelligence (ASAI-99), Buenos Aires, AR, pp. 7-35, 1999.

[14] Sebastiani F., "Machine Learning in Automated Text Categorization", ACM Computing Surveys, Italy, Vol. 34, No. 1, pp. 1-47, March 2002.

[15] Feldman R., Sanger J., "The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data", Cambridge University Press, USA, New York, October 2006.

[16] Pan F., "Multi-Dimensional Fragment Classification in Biomedical Text", M.Sc Thesis, Queen's University, Kingston, Ontario, Canada, September 2006.

[17] Komorowski J., Pawlak Z., Polkowski L. and Skowron A., "Rough Sets: A Tutorial", In : Pal, S.K., Skowron, A. (Eds) Rough-Fuzzy Hybridization :A New Trend in Decision Making, pp. 3-98, Springer-Verlag, Singapore 1999.

[18] Pawlak Z., "Some Issues on Rough Sets", Transactions on Rough Sets I, Lectures Notes in Computer Science (LNCS) 3100, Springer-Verlag Berlin Heidelberg, Vol. 1, pp. 1-58, 2004.

[19] Suraj Z., "An Introduction to Rough Set Theory and Its Applications: A Tutorial", proceeding of the 1st International Computer Engineering Conference (ICENCO') New Technologies for the Information Society, Cairo, Egypt, pp. 1-39, December 2004.

[20] Pawlak Z., "Rough Set Theory and Its Applications", Journal of Telecommunications and Information Technology, pp.7-10, March 2002.

[21] Kiritchenko S., "Hierarchical Text Categorization and Its Application to Bioinformatics", PhD Thesis, School of Information Technology and Engineering, Faculty of Engineering, University of Ottawa, Ottawa, Canada, 2005.

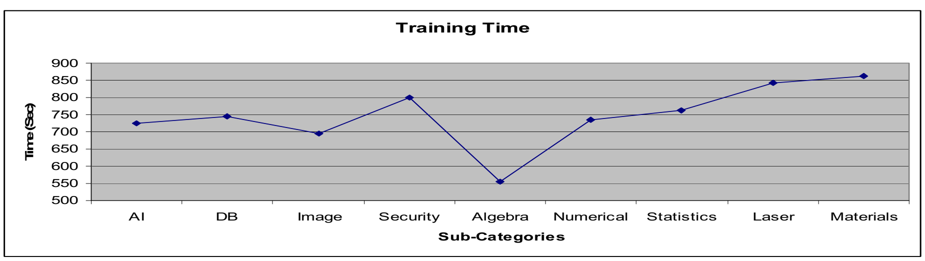

Figure 2. The Learning Time for Building the Classifier.

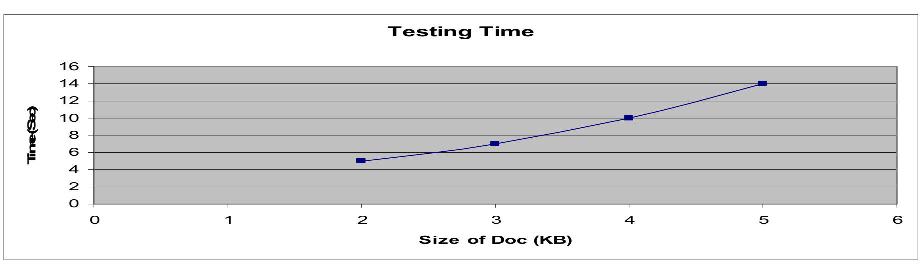

Figure 3. The Average of the Testing Time for Classifying of the Test Documents.

Table 1. The results of Calculating Precision & Recall for the Proposed System.

Category	Sub-category	Precision	Recall
Computer Science	Artificial Intelligence	100%	100%
	Database	91.66%	100%
	Image Processing	100%	94.7%
	Security	83.3%	95.2%
Mathematics	Algebra	100%	100%
	Numerical Analysis	100%	100%
	Statistics	100%	90.9%
Physics	Laser	100%	100%
	Materials	100%	100%
Unknown		100%	100%

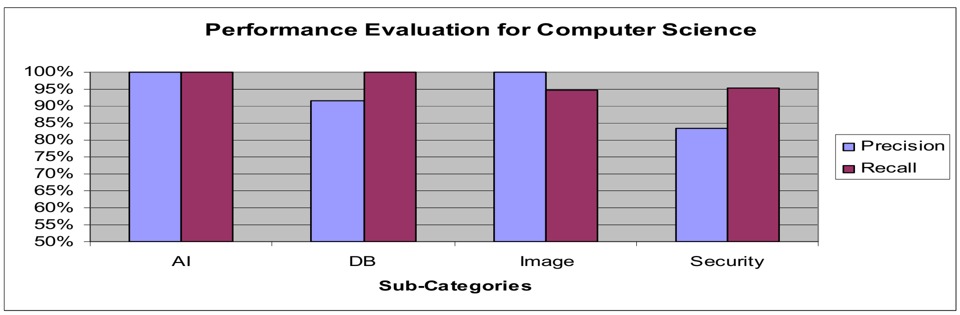

Figure 4. The performance evaluation for Computer Science Category.

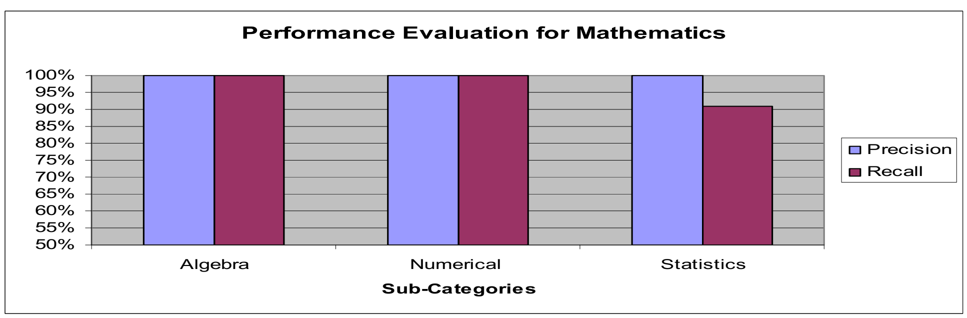

Figure 5. The performance evaluation for Mathematics Category.

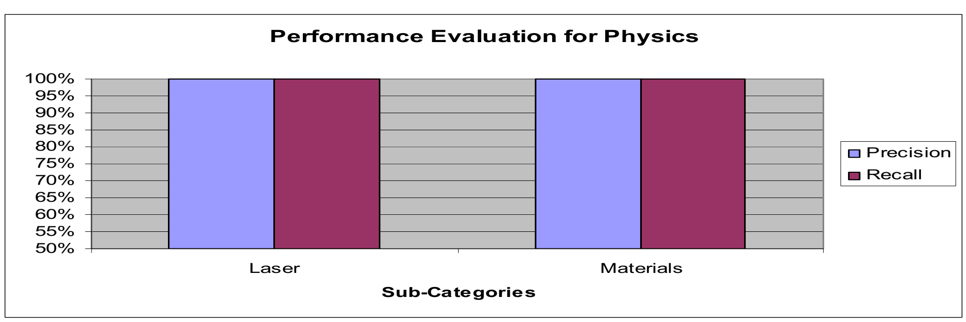

Figure 6. The performance evaluation for Physics Category.