PDF Web Documents Categorization Using Association Rules Mining

Fadhil Hannoon Abbood

Al-MustanseriahUniversity - College of education
Computer science Dep. Email: fadhil_alsaadi@yahoo.com

ABSTRACT

Documents categorization aims to mapping text documents into one or more predefined class based on its contents. This problem has recently attracted many scholars in the web mining and machine learning communities since the numbers of online documents that hold useful information for decision makers, are numerous. This paper investigates the method of classifying PDF Web documents using association rule mining. The number of PDF documents is collected and analyzed, to detect vital and essential features. Ranks values are suggested for these features. A Mutual Meaning Unify (MMU) technique is proposed for increasing the accuracy of documents representations. To reduce the document vector space, stop words are removed. To reduce the documents terms, a stemming algorithm is using. Because the large number of generated rules, a pruning process is proposed to keep on only the highly distinguishing rules. The resulting rules which construct the classifier are used for categorization process. As a result, the classifier is accurate and operates well, it has accuracy about (97%) and the error rate (3%).

الخلاصة: إن تقنية إستكشاف قواعد الإرتباط أُستخدمت لإستخلاص الخصائص وقواعد التصنيف بإستخدام مجموعة من الوثائق المعدة مسبقاً والمعروف أصنافها. لتحقيق أهداف هذا البحث في عملية تصنيف وثائق الويب، تم اعتبار المشكلة من أربعة مهام أساسية وهي، إستخلاص النصوص، إعادة معالجة و تمثيل الوثائق، تكوين المصنف وأخيراً تقييم هذا المصنف. تم جمع عدد من ملفات الوثائق المحمولة وتحليلها لإكتشاف عدد من الخصائص الأساسية والمهمة. نتيجة التحليل أدت إلى أن بعض الخصائص الظاهرية يمكن أن تؤثر بشكل كبير جداً على عملية التصنيف وتحسينه. لذلك، تم تكرارها بعدد معين ضمن النصوص. ولغرض زيادة الدقة في البيانات تم تقديم طريقة الكلمات التبادلية التي لها معنى واحد. قائمة من الكلمات غير الضرورية تم جمعها لغرض حذفها. ومن المعروف أن الكثير من كلمات اللغة الانكليزية تحوي ذيل فوضعت خوارزمية لمعالجة ذلك. تم تشذيب القواعد التي لا تحقق بعض الشروط والمتبقي منها أُستخدم في عملية التصنيف. تم استخدام مقاييس لقياس دقة المصنف، فتبين أن للمصنف دقة عالية جداً وصلت 97% ونسبة خطأ بلغت3%.

Keywords: Categorization, Web Documents, Association Rules.

1. Introduction

The World Wide Web (WWW) has become a powerful platform to store, disseminate and retrieve information as well as mine useful knowledge. Categorization of text documents have been implemented in various applications like filtering spam [1], e-mail categorization [2], e-mail monitoring and Ontology mapping [3]. Many techniques have been applied to text documents categorization, such as Bayesian Networks [4], Decision Trees [5], Neural Networks [6], Support Vector Machine (SVM) [7], K-Nearest Neighbor (K-NN) approach [4], and the some are derived from these methods such as hybrid text mining model by using Naïve Bayes and Rough Set theory [8]. There are usually two major concerns associated with the query-based Web search [9]. The first problem is low

precision, and the second is low recall. Among the proposed algorithms the best known are *Apriori* [10] and FP-tree growth [11]. Thus, this paper exploits the use of association rules mining in building the PDF Web documents categorization system from a relatively large training set.

2. Related work

In this section its outline some of the known techniques for categorizing text. Since the algorithm is based on the association rule mining idea and using PDF documents for categorization purpose, it's briefly present the concepts in order to introduce the subsequent sections.

2.1 Text Categorization

The techniques applied of building a text document classifier can be divided in some sub-classes: probabilistic, decision trees, decision rules, regression models, neural networks and SVMs, etc. The Naïve Bayesian classifiers represent an important trend in the probabilistic classifiers that proved to perform well [4]. One drawback of these classifiers is their sensitivity to term space reduction. Probabilistic classifiers is not easy to interpret or even maintain the numeric models generated by the classifier. Decision trees and decision rule-based algorithms has adopted by many researchers [5][6][7]. These classifiers belong to statistics-based class and probably the best known among them is the "linear Least Squares Fit" (LLSF), proposed in [8] in which documents are modeled in two vectors and the categorization consists of adjusting the category weights. Also in text documents categorization different types of neural networks have been applied in the attempt of making a better classifier [9].The problem in using neural networks is the training time is often excessive and the resulting network is difficult to interpret. Text categorization using support vector machines have also been proposed in the literature [10] that proved to build very powerful systems. Rocchio classifier [11] uses Rocchio's formula for relevance feedback in the document vector space model. This method is the assumption of one centroid per category, and consequently, Rocchio does not perform well when the documents belonging to a category naturally form separate clusters.
The concept of combining classifiers is proposed as a new direction for the improvement of the performance of individual classifiers [10].

2.2 Mining Frequent Patterns and Association Rules

The problem is formally stated as follows. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Let D be a set of transactions. Each transaction has a unique identifier, called its *TID*. A transaction t contains X, a set of some items in I, if $X \subseteq t$. An association rule is an implication of the form $X \Longrightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \Longrightarrow Y$ holds in D with confidence c if the fraction of transactions that also contain Y in those which contain X in D is c. The rule $X \Longrightarrow Y$ (and equivalently $X \cup Y$) has support s in D if the fraction of transactions in D that contains $X \cup Y$ is s. Given a set of transactions D, the problem of mining association rules is to generate all association rules that have support and confidence no less than the user-specified minimum support (*minsup*) and minimum confidence (called *minconf*), respectively. Once frequent itemsets are obtained, it is straightforward to generate association rules with confidence no less than *minconf*. [10]
Two essential steps are needed to mining association rules, theses are: [14][11][13][10]

2.2.1 Finding Frequent Itemsets

Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant, to find all itemsets that have support no less than *minsup*. Itemsets that satisfy *minsup* constraint are called frequent itemsets.

To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property to reduce the search space, which presented as follows: All non-empty subset of a frequent itemset must also be frequent. This property based on the following observation. If an itemset I doesn't satisfy the minimum support threshold, s, then I is not frequent, i.e., Prob{I} < s. If an item A is added to the itemset I, then the resulting itemset (i.e., $I \cup$A) cannot occur more frequently than I. Therefore, $I \cup$A is not frequent either, i.e., Prob{$I \cup$A}<s.

A two-step process is followed, consisting of *join* and *prune* actions:

1. The join step: To find F_k, a set of candidate k-itemsets is generated by joining F_{k-1} with itself. This set of candidates is denoted C_k. Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. For the (k-1)-itemset, F_i, this means that the items are sorted such that $f_i[1] < f_i[2] < : : : < f_i[k-1]$. The join, $F_{k-1} \times F_{k-1}$, is performed, where members of F_{k-1} are joinable if their first (k-2) items are in common. That is, members f_1 and f_2 of F_{k-1} are joined if ($f_1[1] = f_2[1]$) ^ ($f_1[2] = f_2[2]$) ^: : :^($f_1[k-2] = f_2[k-2]$) ^ ($f_1[k-1] < f_2[k-1]$). The condition $f_1[k-1] < f_2[k-1]$ simply ensures that no duplicates are generated.

2. The prune step: C_k is a superset of F_k. A scan of the database to determine the count of each candidate in C_k would result in the determination of F_k (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to F_k). To reduce the size of C_k, the Apriori property is used as follows: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any (k-1)-subset of a candidate k-itemset is not in F_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k. Figure (1) illustrates the Apriori algorithm.

```
Input: D: data set; minsup: minimum support;
Output: F: frequent itemsets;
1   Begin
2     F1 = {frequent 1-itemsets};
3     For (k = 2; Fk-1 ≠Ø; k++}; do
4     begin
5       Ck = apriori-gen (Fk-1 ) // New candidates
6       foreach transaction t ∈D do
7       begin
8         Ct = subset (Ck ,t); // candidates contained in t
9         Foreach candidate c ∈Ct do
10          c.count ++;
11        Fk = { c ∈Ck ׀ c.count ≥ minsup };
12      end
13    end
14  return ∪k Fk;
15  End
Procedure apriori-gen (Fk-1: frequent (k-1)-itemsets)
1   foreach itemset f1∈Fk-1 do
2     foreach itemset f2∈Fk-1 do
3     begin
4       if ( f1[1]=f2[1] ∧ f1[2]=f2[2]) ∧ ::: ∧ (f1[k-2]=f2[k-2]) ∧ (f1[k-1] <f2[k-1]) then
5         begin
6           c= f1 × f2 // join step: generate candidates
7           if hash-infrequent-subset (c; Fk-1) then
8             delet c; // prune step: remove unfruitful candidate
9           else add c to Ck;
10        end
11    end
12  return Ck;
13  end
procedure hash-infrequent-subset (c:candidate k-itemset; frequent (k-1)-itemsets);
1   foreach (k-1)-subset s of c do
2     if s ∈Fk-1 then
3       return True;
```

Figure (1) Apriori algorithm [21]

2.2.2 Generating Association Rules from Frequent Itemsets

An improving of Apriori algorithm is done by excluding the pruning step using the vertical layout concept to represent the database. The developed algorithm does not generate candidate's itemsets; it generates itemsets with their Tidlist by the operation of union and intersection respectively. The generated item will be excluded or regarded directly as a frequent itemset and simply by counting the length of its Tidlist. Figure (2) illustrates the developed Apriori algorithm for generating frequent itemsets. [15]

```
Input: set of documents Di; minsup.
Output: Frequent itemsets
Procedure Developed_Apriori( )
13  Begin
14      L1= {large 1-itemsets }
15      K=2
16      While Lk-1 ≠ Ø do
17      Begin
18          Lk= Developed_apriori_gen(Lk-1)
19          K=K+1
20      End%While
21  End%apriori
```

```
Developed_apriori_gen (Lk-1);
1   Ck= Ø
2   For all itemsets X € Lk-1 and Y € Lk-1 do
3     If X1=Y1 ^ …… ^ Xk-2=Yk-2 ^ Xk-1=Yk-1
    then
4     Begin
5       C=union (X,Y);
6       CTID=intersect (XTID,YTID);
7       If | CTID | >= minsup then
8         add C to Lk;
9       Else
10        Ignore C;
11    End;
```

Figure (2) Developed Apriori algorithm

3. The Proposed Method

In this paper, the categorization method is applied for PDF Web documents with taking the advantages of these documents feature. The categorization method proposed is an incorporation of association rule mining task with categorization problem.

First of all, the Web document files are downloaded from the known explorer Yahoo! Directory and Google search engine Websites. These documents will be stored at database files designed to be repository for the data corpus. After that, there are seven phases of proposed method these are : **Features Extraction Phase, Text Extraction Phase, Features Ranking, Preprocessing and Representation Phase, Term Weighting Phase, Classifier Construction Phase and finally Class Discovery For New Documents**

3.1 Features Extraction Phase

In this paper it is proposed to extract text features from PDF based on exploiting the visual knowledge humans make use of when reading a document. Some visual features of a PDF documents can be extracted. These features are Title, Keywords, Abstract and Conclusion. The importance of the proposed features is obtained by analyzing a collection of (100) PDF web documents.

3.2 Text Extraction Phase

Most PDFs have little or implicit structural information, making automated machine processing and data extraction a difficult task because these are not data formats but algorithmic descriptions of how the document should be rendered. In particular, text rendering is controlled by a set of show commands. Arguments to show commands are text strings and two-dimensional coordinates. However, these strings are not necessarily entire words which are typically split into fragments and separate show commands are issued for each of the fragments. Show commands do not need to appear in reading order. The graphic operators used in PDF content streams describe the appearance of pages that are to be reproduced on a raster output device. Table (1) illustrates some of graphics operators and their descriptions.

Table (1) Some of PDF graphics operators

Operator	Description
T_c	Character spacing
T_w	Word spacing
T_h	Horizontal scaling
T_r	Set the text rendering mode
T_z	Set the horizontal scaling, T_h, to (*scale* | 100)
T_d	Changing only the text location
T_D	Changing text location and leading
T*	Moving to start of next line of text, as defined by the leading
T_m	Set the text matrix, T_m, and the text line matrix, T_{LM}
T_j	Show a text string
T_J	Show one or more text strings
cm	Modify the current transformation matrix by concatenating the specified matrix
rg	Set the color space to DeviceRGB

The basic content extraction procedure consists of three phases: Content Decompression, Extract Characters or Phrases and Tokenize the Extracted Text characters or Phrases into Accepted Terms.

3.2.1 Content Decompression

PDF files have content streams that are usually created in a compressed form encoded using one or more compression algorithms. The resulting file contains commands and content in a process-able form. For content extraction, much of this is superfluous, and it is concentrate mainly on information of characters or phrases. In this paper, PDF files streams are decompressed using iTextSharp dll and then the results are used as a main source for text extraction process. Examples (1 and 2) illustrate the snippets of compressed and decompressed PDF Content Streams.

Example (1)

/Filter [/ASCII85Decode /LZWDecode]
>>
stream
J..)6T`?p&<!J9%_[umg"B7/Z7KNXbN'S+,*Q/&"OLT'FLIDK#!n`$"<Atdi`\Vn%b%)&'c
A*VnK\CJY(sF>c!Jnl@RM]WM;jjH6Gnc75idkL5]+cPZKEBPWdR>FF(kj1_R%W_d&
/jS!;iuad7h?[LF$+]]0A3Ck*$I0KZ?;<)CJtqi65XbVc3\n5ua:Q/=0$W<#N3U;H,MQKqfg
1?:lUpR;6oN[C2E4ZNr8Udn.'p+?#X+1>0Kuk$bCDF/(3fL5]Oq)^kJZ!C2H1'TO]Rl?Q:&

'<5&iP!$Rq;BXRecDN[IJB`,)o8XJOSJ9sDS]hQ;Rj@!ND)bD_q&C\g:inYC%)&u#:u,M
6Bm%IY!Kb1+":aAa'S`ViJglLb8<W9k6Yl\\0McJQkDeLWdPN?9A'jX*al>iG1p&i;eVo
K&juJHs9%;Xomop"5KatWRT"JQ#qYuL,JD?M$0QP)lKn06l1apKDC@\qJ4B!!(5m+j.
7F790m(Vj8l8Q:_CZ(Gm1%X\N1&u!FKHMB~>
Endstream

Example (2)
>>
stream
2 J
BT
/F1 12 Tf
0 Tc 0 Tw 72.5 712 TD
[(Unencoded streams can be read easily)65 (,)] TJ
0 -14 TD
[(b)20 (ut generally tak)10
(e more space than \311)] TJ
T* (encoded streams.)Tj
0 -28 TD
[(Se) 25 (v) 15 (eral encoding methods are a) 20 (v)
25 (ailable in PDF)80 (.)] TJ
0 -14 TD
(Some are used for compression and others simply) Tj
T* [(to represent binary data in an) 55
(ASCII format.)] TJ
T*
(Some of the compression encoding methods are \
suitable)
Tj
T*
(for both data and images, while others are \
suitable only) Tj
T* (for continuous-tone images.)Tj
ET
Endstream

3.2.2 Extract Characters or Phrases

The actual characters or phrases that represent the text are extracted between the so called Content Streams. Figure (3) illustrates the algorithm of extracting the Tokens that represents the text from PDF files.

Input: PDF files where PDF = {PDF1, PDF2, …, PDFn}; Table [PDF format code] which consists of {BT, ET, TD , Tj ,…, etc};Table [symbols] which consists of {;, [,],(,), :, -, /,…. , etc};

Output: PDF Text Files.

```
1  Begin
2    For each PDFn do
3      For each Token in PDFn do
4      Begin
5          If Token ∉ PDF format code or symbols then
6            Add this Token to Text File
7          Else
8            Ignore this token
9          End If
9      End
10 End
```

Figure (3) Algorithm for Extraction the Tokens from PDF files

3.3.3 Tokenize the Extracted Text Characters or Phrases into Accepted Terms

In Content Extraction phase, the result is consequence of Tokens or phrases that construct the text. In many cases, these extracted Tokens that represent the output text cannot be recognized as accepted terms. This issue can be illustrated in two examples:

Example (3)

BT
TT3 1 Tf
7.82560 0 0 7.82560 52.89826 541.31149 cm
1.00000 0 0 1.00000 0 0 Tm
w 1.2 **eb** 8.6 4.6 **d** -6.0 **o** -6.0 **m** 41.9 **a** -16.0 **i** 17.3 **n** -6.0 -10.6 **i** 17.3 **n** -6.0 4.6 **o** 9.3 **r** -19.6 **d** -6.0 **er** -5.0 4.6 **t** 17.3 **o** 9.3 4.6 **b** 9.3 **en** -6.7 **e** -16.0 **f** 10.9 **i** 1.9 **t** 1.9 4.6 **f** 10.9 **r** -4.3 **o** -21.3 **m** 26.6 4.6 **t** 1.9 **h** -6.0 **e** -16.0 19.9 **p** 9.3 **o** 9.3 **w** 1.2 **er** -20.3 TJ
ET

In example (3), after extracted the characters, the output text represents "**Web domain in order to benefit from the power**" where the Tokens represent accepted terms because there are spaces between the terms.

Example (4)

BT
TT2 1 Tf
15.53440 0 0 15.53440 220.53826 725.03149 cm
1.00000 0 0 1.00000 0 0 Tm
V 10.3 **is** -8.0 **ual** -7.3 **W** 1.5 **eb** -7.2 **M** 6.4 **i** -7.8 **ni** -7.0 **ng** 9.2 TJ
ET

In example (4), the snippet of content stream represents the output text "**VisualWebMining**", where the Tokens represent unaccepted terms (merged terms). The solution that is proposed to resolve this issue is to use a dictionary database. Figure (4) illustrates the algorithm for extracting the Tokens into accepted terms.

```
Input: PDF Text Files; Dictionary DB;
Output: PDF Text Files have Tokens Represent Accepted Terms
1  Begin
2    For each PDF Text File do
3      For each Token in PDF Text File do
4      Begin
5          i=0, j=0
6          initialize Tokens_array and Token_array2 to empty string for each entry.
7          If Token Not in Dictionary DB then
8          Begin
9             CH = characters of Token; x = no. of   characters in CH
10            Token_Char = ""
11            For y = 1 to x do
12            Begin
13               Append CH [y] to Token_Char
14               If Token_Char in Dictionary DB then
15               Begin
16                  i=i+1
17                  Token_array2[i] ← Token_Char
18               End
19            End
20            If (i<>0) then  //No term in Token_array2
21            Begin
```

Figure (4) Algorithm for Processing the Extracted Tokens into Accepted Terms.

After plain text is extracted, punctuation and other special characters need to be stripped off and the character case be transformed to lowercase to reduce the number of index terms. To increase the accuracy, (MMU) is proposed to be used which considers combination of abbreviations and synonyms of word. Table (2) illustrates that where the terms on left hand side are translated to the phrase on the right hand side.

Table (2) Sample of Mutual Meaning Uniform items

LHS	RHS
ai	artificial-intelligence
aiml	artificial-intelligence-meta-language
ann	artificial-neural-network
cgi	common-gateway-interface
csp	communication-sequential-processes
db	Database

3.3 Features Ranking

In this paper an analysis is done to detect the effect of these features with taking the text properties in Web documents categorization consideration. The useful effect of determining these features on accuracy is taken into categorization. As a result of analysis process of PDF document, the following text features are adopted: Title, Keywords, Abstract, Conclusion.

The significant features of PDF documents are extracted and added to each document vector in order to contain significant terms. Weights are suggested for each feature after many experiments. Table (3) shows the feature and related rank for PDF documents.

Table (3) PDF text document features and their ranks

PDF Feature	Related Rank
TITLE	10
Abstract	4
Keywords	8
Conclusion	3

3.4 Preprocessing and Representation Phase

Documents are given in terms of raw data which usually needs a transformation into machine process-able representation. Cleaning text from insignificant terms and stemming are prevalent part of preprocessing phase. Generally, pre-processing stage consists of the following two processes:

3.4.1 Stop Words Removal

The words such as articles, pronouns, prepositions and conjunctions that are used to provide structure in the language rather than content are removed. Also it is proposed to remove the terms or Tokens which have length shorter than (4) and greater than (15) characters because most of these terms represent PDF source code instructions or tokens are extracted incorrectly.

3.4.2 Text Conflation and Vocabulary Reduction

Often it is desirable to reduce all the morphological variants of a given word to a single index term. Stemming can be also useful to reduce the size of the index and improve the performance of knowledge discovery in database. For stemming process, it is desired to design improved stemming rules called 'Plain Stemming Method' (PSM). The results show that this method has accuracy of about (%80).

Plain Stemming Method (PSM)

The proposed (PSM) contains rules for stemming process, as follows:

***Definition*:**

Let SW= Stemmed Word.

Rule #1:
IF a word ends in "ically"
THEN "ally" -------> NULL

Rule #2:
IF a word ends in "lly"
THEN "ly" -------> NULL

Rule #3:
IF a word ends in "ily"
THEN "ily" -------> "y"

Rule #4:
IF a word ends in "bly"
THEN "y" -------> "e"

Rule #5:
IF a word ends in "ly"
THEN "ly" -------> NULL

Rule #6:
IF a word ends in "sses"
THEN "es" -------> NULL

Rule #7:
IF a word ends in "es", but not "aes", "ees", "ies" or "oes"
THEN "s" -------> NULL

Rule #8:
IF a word ends in "ness"
THEN "ness" -------> NULL

Rule #9:
IF a word ends in "ies", but not "eies" or "aies"
THEN "ies" -------> "y"

Rule #10:
IF a word ends in "s", but not "is", "us" or"ss"
THEN "s" -------> NULL

Rule #11:
IF a word ends in "ated"
THEN "d" -------> NULL

Rule #12:
IF a word ends in "ied"
THEN "ied" -------> "y"

Rule #13:
IF a word ends in "ued"
THEN "d" -------> NULL

Rule #14:
IF a word ends in "ed" but not "eed"
THEN "d" -------> NULL

Rule #15:

Rule #16:
IF a word ends in "izer"
THEN "r" -------> NULL

Rule #18:
IF a word ends in "tting"
THEN "ting" -------> NULL

Rule #20:
IF a word ends in "ating"

IF a word ends in "ization"
THEN "ation" -------> "e"
Rule #17:
IF a word ends in "ier"
THEN "ier" -------> "y"
Rule #19:
IF a word ends in "mming"
THEN "ming" -------> NULL
Rule #21:
IF a word ends in "ation"
THEN "ion" -------> "e"

To avoid errors occurrence in the terms that are stemmed incorrectly, each stemmed term is then checked with a dictionary data base.

3.5 Term Weighting Phase

To reduce the high dimensionality of the feature space. Term weighting scheme is a vital step in Web documents categorization and reduces the dimensionality space. the following common variant was proposed to be used; it is used by Cornell University [16].

$$TF.IDF = tf_{t,d} * log((1+N)/ n) \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where $tf_{t,d}$ is the frequency of word *t* in document *d*, *N* is the number of documents in the text collection and *n* is the number of documents where word *t* occurs.

3.6 Classifier Construction Phase

The next step is building the Classifier, which are one of the most important issues in this paper. The classifier construction phase consists of the following processes: **Count Frequent Itemsets, Discovering Association Rules and Association Rule Pruning.**

1.6.1 Count Frequent Itemsets

The frequent itemsets discovering is done by using developed Apriori algorithm as it is mentioned in section (2.2.1).

1.6.2 Discovering Association Rules

When the count frequent itemsets terminated, the next step is discovering association rules from these frequent itemset. This step is done as illustrated in section (2.2.2).

1.6.3 Association Rules Pruning

After rules discovering process, association rules pruning process is proposed, including the following steps:

1. Pruning the rules which do not have minimum confidence (*minconf*) threshold.
2. Pruning the rules which do not lead to given class label.

1.7 Class Discovery For New Documents

To determine the categorization of a new document in the test set by using the same phases in training set. The principle to discover class label for new document depends on a degree of similarity between new document attributes and the categorization rules attributes. A score is given to a class related to a new document. This score is counted for each category for a certain document. The highest class score provides the predicted category for a certain document. Figure (5) depicts simple algorithm for class discovery process of new document.

Input: *Web document D, class labels with categorization rules;*
Output: *document with class label;*
Begin
  Score=0;
  For each Class Label categorization rules attributes **do**
  Begin
    For each term t_i in document D **do**
      Score=Score+(No. of t_i in D)*(No. of matching rules attributes)

Figure (5) Algorithm for class discovery process

2. Experimental Results

The set of rules is built to discover the class label of new documents. The data are divided into training sets and test sets, the training sets are used to construct the classifier, the test sets are used to test categorization rules. The input of this phase is a set of categorization rules for each class label and the output is the class label of new document attributes associated with these rules. Test documents must be passed across the Text Extraction, Web Documents Features Ranking, and Document Preprocessing and Representation processes to build the new document vector space. The document results in the form $d=\{t_1, t_2, t_3, \ldots\ldots, t_n\}$. The categorization rules are collected from each class label. They are "Web-Mining", "Machine-Learning", "Neural-Networks", "Image-Processing". Table (4) illustrates the number of categorization rules extracted for each class label using the proposed ranking method which is used for discovering the class labels for new documents.

Table (4) No. of Extracted Rules using proposed ranking method

Class label	No. of training Doc.	Total rules satisfy *minconf* ≥ *60%*	Pruned Rules	Rules have class label
Web-Mining	40	24708	24651	57
Machine-Learning	40	3658	3598	60
Neural-Networks	40	2310	2230	80
Image-Processing	40	16934	16742	192
Total Rules	160	47610	47221	389

Discovering the class label for new document is done by a score given to a class related to a new document. This score is counted for each category of a certain document. The highest score value represents the class discovered for related document. An experiment is done for the prediction process of each "Web-Mining", "Machine-learning", "Neural-Networks" and "Image-Processing" class labels using the proposed algorithm. The score value is calculated from the sum of each term frequency multiplied by the rules occurrence found in the categorization rules. Table (5) illustrates the results for some new PDF documents.

Table (5) Categorization result for some new PDF documents

Doc. ID.	Class labels Score Result				Class label
	WM	ML	NN	IP	
WMT-28.pdf	11260	6010	1091	17883	Image Processing
WMT-35.pdf	13886	6512	2196	4427	Web Mining

MLT-7.pdf	2718	7328	4122	4595	Machine Learning
MLT-13.pdf	0	438	0	0	Machine Learning
NNT-24.pdf	55	962	2503	219	Neural Networks
NNT-29.pdf	1351	2982	2902	1192	Machine Learning

The precision and recall are the standard measure of Information Retrieval (IR) performance. A recall for a class is defined as the percentage of correctly classified documents among all documents belonging to that category, and precision is the percentage of correctly classified documents among all documents that were assigned to the class by the classifier [12]. Table (6) illustrates a confusion matrix that contains information about actual and predicted results given by a classifier.

Table (6) Confusion matrix of a classifier

	Classified positive	Classified negative
Actual Positive	TA	FA
Actual negative	FR	TR

where:
TA: the number of correct categorizations that are **true accepted.**
FA: the number of incorrect categorizations that are **false accepted**.
FR: the number of incorrect categorizations that are **false rejected**.
TR: the number of correct categorizations that are **true rejected**.
Based on the confusion matrix, the precision (*p*) and recall (*r*) of the positive class are defined as follows: [5]

$$precision = \frac{TA}{TA+FA} \quad \text{..... (2)}$$

$$recall = \frac{TA}{TA+FR} \quad \text{..... (3)}$$

Alternatively, there is the *F*1 measure, equal to 2/(1/recall + 1/precision), which combines the two measures in an ad hoc way. The value of *F1* is high only when both precision and recall are high, and is low for design options that trivially obtain high precision by sacrificing recall or vice versa. [12][5]
Table (7) shows the various document retrieval statistics. Documents are either retrieved or omitted based on their value to the classifier prediction. Documents are either relevant or irrelevant.

Table (7) Relevant and non-Relevant retrieved documents

	Relevant	Non- Relevant	Total
Retrieved	TA	FA	TA+FA
Not Retrieved	FR	TR	FR+TR
Total	TA+FR	FA+TR	TA+FA+FR+TR

where:
(TA+FA) : all accepted documents.
(FR+TR) : all Rejected documents.
(TA+FR) : all relevant documents in collection.
(FA+TR) : all Non-Relevant documents in collection.
The accuracy and error rates are calculated as follows: [5] [17]

$$Accuracy = \frac{(TA+TR)}{(TA+FA+FR+TR)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

$$Error = 1 - \ Accuracy \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5)$$

Using results of classifier, the experiment shows that the proposed methods for categorization web documents gives high quality results. After applying precision and recall equations to a number of PDF test documents. Table (8) shows the categorization results using (40) test documents for "Web-Mining" class label.

Table (8) Categorization result for "Web-Mining" class label

Documents categorization results	Relevant categorizatio	Non-Relevant categorization	Total categorization
classified	(39) TA	(3) FA	(42) all accepted doc.
Unclassified	(1) FR	(117) TR	(118) all rejected doc.
Total	(40)TA+FR	(120)FA+TR	

As illustrated in table (8), the results show that (39) of (40) test documents are classified correctly to relate class label and (1) are classified incorrectly. When applying the precision, recall, accuracy and error measures, the results show that the precision is (93%) and recall is (98%), which are high. The accuracy of the performance is (98%), the error is (2%). Figure (6) illustrate the categorization results for all class labels.

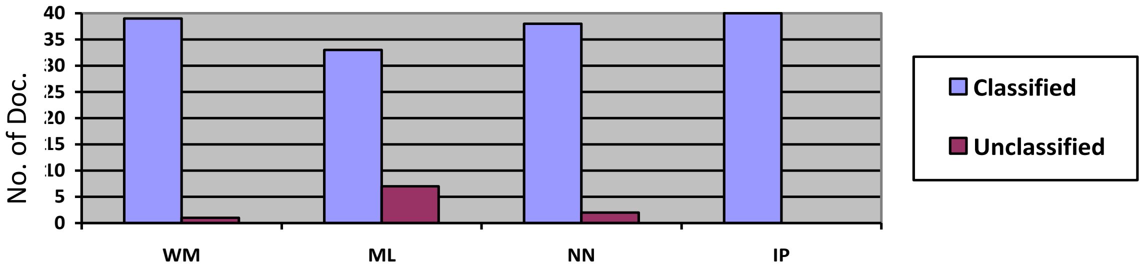

Figure (6) Classified and Unclassified Documents

By using the same experiment categorization results for four class labels, the precision, recall, accuracy and error can be calculated. Table (9) illustrates these results.

Table (9) Categorization measure results for four class labels

Class Label	Precision %	Recall %	Accuracy %	Error %
Web-Mining	93	98	98	2
Machine-Learning	97	83	95	5
Neural-Networks	100	95	99	1
Image-Processing	87	100	96	4
Average	94	94	97	3

To condense precision and recall into one number, the F_1-Measure is used. Table (10) illustrate the F_1-Measure equation results for four class labels.

Table (10) F_1-Measure values of the four class labels

Class Label	F_1-Measure %
Web-Mining	95
Machine-Learning	89

Neural-Networks	97
Image-Processing	93
Average	94

5. Conclusions

There are several conclusions drawn from this paper:

1- A new method is developed to automatic web documents categorization. Once the categorization model is learned, it can be used for classifying future PDF documents.

2- The experiments clearly indicate that the categorization process is effective. It improves the retrieval performance compared with that of no categorization. It also achieves the retrieval performance equivalent to the results using manual categorization.

3- Experimental results show that the classifiers built in this way are common and accurate and perform well. This approach enables categorization with overall accuracy of about (97%) while the error measure is (3%).

4- The categorization using some features ranking such as PDF visual text features improves the categorization results by giving a discriminative feature weights among classes.

5- One of the main advantages of categorization using association rule-based classifiers is its relatively fast. But the training phase takes considerable amount of time because the size of training set.

6- It represents the natural way of representation therefore the rules are easy for implementation. Also the rules are reusable, transparent, easy to modify and update.

7- As it was elucidated, the experiments show that the proposed algorithms used to implement these sub-modules exhibit very high scalability because the algorithms are independent of the hugeness of the number of documents.

8- It is robust, i.e. shows a similar behavior on all data sets.

References:

[1] Delany S. , Cunningham P., Tsymbal A. and Coyle L., ***A Case-Based Technique for Tracking Concept Drift in Spam Filtering***, J. Knowledge Based Systems, vol. 18, nos. 4-5, pp. 187-195, 2004.

[2] kim S., Han K., Rim H. and Myaeng S., ***Some Effective Techniques for Naïve Bayes Text Classification,*** IEEE Transactions on Knowledge and Data Engineering, 2006.

[3] Mao M., Peng Y., Spring M., ***Ontology Mapping: As a Binary Classification Problem***, IEEE Fourth international conference on Semantics, Knowledge and grid, 2008.

[4] Han J. and Kamber M., ***Data Mining: Concepts and Techniques***, Morgan Kaufmann Publishers, 2006.

[5] Liu B., ***Web Data Mining***, Springer-Verlag, 2007.

[6] Saha D., ***Web Text Classification Using a Neural Network,*** Second International Conference on Emerging Applications of Information Technology, IEEE publisher, 2011.

[7] Guyon I., Weston J., Barnhill S. and Vapnik V., ***Gene selection for Cancer Classification Using Support Vector Machines,*** *Machine Learning*, 46(1-3):389–422, 2002.

[8] Vidhya K. and Aghila G., ***Hybrid Text Mining Model for Document Classification,*** *2nd International Conference on Computer and Automation Engineering (ICCAE)*,IEEE publisher, 2010.
[9] Kosala R. and Blockeel H., ***Web Mining Research: A Survey. SIGKDD Explorations***, 2(1): pp. 1-15., 2000.
[10] Agrawal R. and Srikant R., ***Fast Algorithms for Mining Association Rules***. In *Proc. 1994 Int. Conf. Very Large Data Bases*, Pages 487-499, Santiago, Chile, September 1994.
[11] Han J., Pei J., and Yin Y., ***Mining Frequent Patterns Without Candidate Generation,*** In *ACM-SIGMOD*, Dallas, 2000.
[12] Feldman R. and Sanger J., ***The Text Mining Handbook: Advanced approaches in analyzing unstructured data***, Cambridge University press, 2007.
[13] Mitsa T., ***Temporal Data Mining***, Chapman & Hall/CRC, 2010.
[14] Margaret H., ***Data Mining, Introductory and Advanced Topics,*** Department of Computer Science and Engineering Southern Methodist University, Prentice Hall, 2002.
[15] Al-Kafagi H., ***Pruning of Apriori's Algorithm Pruning Steps***, journal of Al-Rafidain University College,2004.
[16] Cornell University, "***SMART IR System***", available at (*ftp://ftp.cs.cornell.edu/pub/smart*), 1990.
[17] Berry M. and Kogan J., ***Text Mining: Applications and Theory***, John Wiley & Sons, Ltd, 2010.