Intelligent Documents Classification System

Hasanen S. Abdullah *,Ph.D.(Asst. Prof.) **Hala Dhiaa Hasan***

Abstract: There are a huge number of documents that available in many various sources in unorganized format, therefore these unstructured documents needs to be classified. In this paper, a proposed system called "Intelligent Documents Classification System" which represents the system for classifying the documents to the correct class based on its textual information. This system contain through four steps which are preprocessing, features extraction, proposed method for features selection, and finally, modify model of naïve bays. Two datasets are used to evaluate the proposed system, the first dataset its name as "bbc from ucd repository" is standard that contains technical research documents distributed over five classes which available on the internet and the second dataset is collected dataset contains books documents distributed over six classes which collected during this work. The IDC system achieved the powerful results. For the standard dataset the accuracy is 95.1%, precision is 95%, recall is 95.8%, and f1-measure is 95.39% while the accuracy for the collected dataset is 95.3%, precision is 95.16%, recall is 95.83%, and f1-measure is 95.49%.

Keywords: Documents Classification, Naïve Bays, Features Extraction, Features Selection, TF-IDF.

*University of Technology

1. Introduction

The large number of electronic documents that contains unstructured and semi structured information became available from different resources like the governmental electronic repositories ,world wide web, news articles, chat rooms, online forums, blog repositories, biological databases, electronic mail and digital libraries [1].

Documents classification known as the task of assigning unlabeled documents into one or more predetermined classes. In order to execute classification task the documents must be represented which is convert the textural information of the documents into a compact format [2]. To solve the task of classification there are two kinds of machine learning approaches supervised and unsupervised approach. The predetermined classes are given for training set of documents in supervised, and documents clustering which also unsupervised documents classification where perform the classification entirely without return to outer information [3]. Machine learning algorithms are used to categorize the documents and their process and architecture are various. This machine learning algorithm is Support Vector Machines (SVM), rule induction, naïve-bayes, K-Nearest Neighbors (KNN), decision trees, Neural Networks (NN) [4].

2. Related Works

There are several works that are considered as related works to the documents classification system using different techniques, some of these works are described below:

In 2017, Shugufta Fatima and B. Srinivasu, addressed a work entitled "Text Document Categorization Using Support Vector Machine" presented the system for classifying Reuter corpus by using a support vector machine technique. This system perform many step such as prepossessing the documents by performing tokenization, delete stop word, stemming algorithm and then vector space model is performed for features extraction by using term frequency technique. SVM is used to classify the document into predetermined classes. The better accuracy is obtained when partitioned the documents into 80 training and 20 testing so that when the number of classified document is 146 and unclassified document is 23 the accuracy is 86.39% [5].

In 2015, Suresh Kumar, shows a work entitled "Optimization of Text Classification Using Supervised and Unsupervised Learning Approach". Their work presented various supervised and unsupervised approach such as KNN, Linear SVC, SGD, K-Means, Multinomial NB, Bernoulli NB, and SVM to perform documents classification by using 20,000 newsgroup documents divided over 20 various newsgroups. The accuracy of KNN is 78.27%, Linear SVC is 86.24%, SGD is 86.27%, K-Means is 87.6%, Multinomial NB is 84.23%, Bernoulli NB is 83.4%, and SVM is 84.23% respectively [6].

In 2013, Khushbu Khamar, presented a work entitled "Short Text Classification Using KNN Based on Distance Function", that shows document classification system based on short text. This kind of categorization does not require long time as opposed to the categorization of big text because in classification of the short text, a few words are used. The KNN algorithm is used and gives better result than SVM and Naïve Bayes algorithms by depending on the distance function (Euclidean distance and Pearson Correlation distance) and the value of KNN which is either K=3 or K=5. The f1-measure of KNN is 0.78, SVM is 0.50, and NB is 0.59 respectively [7].

In 2013, Anuradha Patra and Divakar Singh, addressed a work entitled "Neural Network Approach for Text Classification Using Relevance Factor as Term Weighting Method", that presented system for document classification by using Back Propagation Neural Network (BPNN) algorithm. The dataset contain150 text document belonging to three classes (computer science, sports, and medicine). This system first perform preprocessing by deleting stop words and execute stemming algorithms then two term weighting methods are used such as Term Frequency (TF) and proposed method which is named as Relevance Factor (RF) then the classifier is constructed by using BPNN. The experiment results show that the better results are obtained when the two methods (TF-RF) are combined. Without using Relevance Factor the value of f1-measure is 0.72 while using TF-RF the value of f1-measure is 0.92 [8].

3. Theoretical Background

There are many steps can be used for documents classification which illustrates as follow:

1) **Documents Collection:** This is the first step of categorization process. The documents from different sources are gathered in various formats like .html, .doc, .pdf, and web content, etc. These documents are used during the training and testing the classifier [9].

2) **Text Preprocessing:** In the step of data preprocessing, many steps are performed in order to prepare the documents to the following step. There are many steps of preprocessing are performed to obtain the result as (n-gram or single-words) from the input which is plain text of document. Documents preprocessing includes three steps which are tokenization, remove stop word, and stemming [10].

3) **Features Extraction:** One of very common techniques of feature extraction from text is the Bag of Words model which is also called Vector Space model. Since the text is a sequence of characters converted into vector of numerical attributes with a fixed size taken by machine learning algorithms rather than the text documents with different length [11]. In this model the vocabulary (list of words) such as W = <w1,...;wd> can be constructed from all the words that appears in the training set after deleting the stop words, then each document is represented as vector of features like D = <t1,...td> consist from the vocabulary (W) [12]. Term weighting are methods used to allocate appropriate weights to the feature of the vector that represent the document such as Term Frequency (TF), Inverse Document Frequency (IDF), Boolean weight, entropy, and TF-IDF**,** etc [13].

   • **Term Frequency-Inverse Document Frequency [14] :**In this method, the weight of word i in a document d is allocated by counting the number of times the word appears in the document, and in inverse proportion to the number of documents in the document collection in which the word appears. Term Frequency × Inverse Document Frequency (TF×IDF) is most popular approach that take into account the number of word throughout all the documents in the

corpus as opposite to the other methods that neglected this property like Boolean and term frequency.

$$\mathrm{wi} = TF * \log \frac{N}{Ni} \quad (1)$$

where:

TF refer to the number of times the word i appear in document d.
N refer to the total number of documents in the training set.
Ni refer to the number of documents where word i appears.

4) **Features Selection:** The aim of feature selection method is to reduce the dimensions of data set by deleting the irrelevant features in order to decrease over fitting and enhance the accuracy of classification [15]. The features selection methods are divided into two type; filters and wrappers. Document frequency, mutual information, chi-square, and information gain are methods of filters also called scoring schemes [16].

5) **Naïve Bayes Classifier:** The Naïve Bayes (NB) classifier is a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and collections of values in a given data set. The assumption of NB is known as category conditional independence because NB is constructed on the theory of Bayes with independence presumption between the features. The classifier NB assumes the appears of attribute in a category does not rely on the appears of any else attribute. The NB classifier is described as a quick learning algorithm under different problems of supervised classification and it performs well [17]. The NB classifier is a probabilistic classifier depend on NB rule, the posterior probability of a document d being in class c is given as:

$$\mathrm{P(c|d)} = \frac{P(d|c)\,P(c)}{P(d)} \quad (2)$$

$$\mathrm{P(c|d)} = \frac{P(w1,w2,\ldots wn|c)\,P(c)}{P(d)} \quad (3)$$

Where P(wi|c) is the conditional probability of term wi appears in a document d of class c. (w1, w2, .., wn) are the tokens in the document

d and are a part of the vocabulary used for classification. n is the number of such tokens in the document d.

The parameter P(c) is prior probability of class and it is estimated as:

$$P(c) = Nc / N \quad (4)$$

where Nc is the number of documents in class ci and N is the total number of documents in training set.

P(d|c) is the likelihood which is the probability of predictor given class.

P(d) is the prior probability of predictor (d) for all classes.

In documents classification, the goal is to find the best class for the document. The NB classifier predicts the class with the maximum posterior probability [18].

4. Proposed Intelligent Documents Classification System

In this paper, a system for documents classification is present to classify the documents to the correct class based on its content (text of documents). This proposed system which is named as Intelligent Documents Classification (IDC) include two main phases which are the training phase utilized in order to learn features of documents and second phase assign unlabeled documents to the predefined class which called testing phase. The training phase includes input the documents, text preprocessing, extract the features, select the features and training the features of the documents by using the training part of MMNB while the testing phase includes input un-labeled document, text preprocessing and classified the document to the correct class by using the prediction part of modified model of naïve bayes (MMNB). In the testing phase, the output of preprocessing step is set of tokens for each document that fed to the classifying part of modified model of naïve bayes in order to predict the class.

The obtained result from the steps (features extraction and features selection) in the training phase helped to execute classifying step after preprocessing step directly in the testing phase. Algorithm in figure (1) illustrates the overall IDC system.

Algorithm (1): The overall IDC System

Input: set of documents.
Output: documents classification

Begin
// Learning phase
Step1: Read the set of labeled documents.
Step2: Preprocessing the documents do
Tokenization to separate each word from others.
Normalization to delete all the numbers, symbols and convert the remaining tokens to lower case.
Remove stop words.
Stemming to remove all the affixes from the tokens.
Step3: Features extraction
// building a vocabulary
For all the documents do
Begin
Construct a list of features from set of tokens, each feature associate with its index.
End
// documents representation
For each document do
Begin
Convert the document into vector.
Calculate the weight for each feature in the document
Begin
Compute TF-IDF by using Eq.(1).
End
End
Step4: Features selection based on their highest weights TF-IDF and determined the threshold algorithm (2).
Step5: Using MMNB classifier
Take list of features only.
Train the features by using the training part of MMNB algorithm (3).
// Testing phase
Step1: Read the set of unlabeled documents.
Step2: Preprocessing the documents do
Tokenization to separate each word from others.
Normalization to delete all the numbers, symbols and convert the remaining tokens to lower case.

Remove stop words.
Stemming to remove all the affixes from the tokens.

Step3: Classify the documents by using the prediction part of MMNB algorithm (3).
End

Figure (1): Overall IDC system of training phase and testing phase

The details of IDC system is explain as followings:

1. Datasets

Two datasets used in IDC system. These datasets are standard and collected, the standard dataset contains technical research documents distributed over five classes which are (business, entertainment, politics, sport, and technical) that available on the internet [19] while the collected dataset contains books documents distributed over six classes which are (artificial intelligent, computer network, image preprocessing, biology, chemical, and unknown). These dataset was collected during this work from many different resources.

2. Preprocessing Step

There are four sub steps to preprocess the text. These sub steps are; tokenization which is used to break the text into tokens by depending on the space to separate each word from other, normalization which is used to convert all the tokens to lower case and delete all tokens which are non-words such as numbers and symbols, remove stop words such as (the, that, most, few, are, etc) and these words does not have any marked significance in classification process so that these words are deleted, and finally the stemming which is used to delete all the affixes from the words. The output of this step is a set of tokens after perform the above four sub-steps.

3. Features Extraction Step

Bag of words model is used to extract the features from the documents by building a vocabulary that contains all the words in the

documents of the training set and converting each documents into vector of features then assign the weight TF-IDF to each feature in the document by using Eq. (1). TF-IDF is utilized to measure the significant of a feature in a collection of documents. The output of this step is vector of weights of features for each document.

4. Proposed Method of Features Selection

Documents classification has been focused on features selection step, because its effect on the accuracy of classification process. The proposed method for features selection is named as Features Selection Based on the Highest Weight of Feature (FSBHWF) and it is applied on the two datasets. This proposed method contain four steps. These steps are :-

A. The first step of this method is used the vector of weight of each document and the weight TF-IDF for each feature in the document is calculated according to Eq. (1).

B. For each document, the weights that associated with their features are sorted from maximum value to minimum value in order to put the highest value in the first index of document`s array.

C. Determine the threshold which is the simplest technique for dimensionality reduction by selecting the feature according to this predetermined threshold. In this method, the assigning threshold is equal to one in order to reduce the number of features and when select one feature which has the highest weight this means that these feature is more frequency in the document and it has high relevance with their class from the other features. When the determined threshold is one that means the choice of features will be only one feature, so that the feature in the first index is always selected for each document.

D. After that, each document is represented as vector contains one feature with their weight TF-IDF.

The output of this method is a list for each class, each list contains the features with their weights for that class. The number of

features in the list depend on the number of documents in that class, because of only one feature from each document is selected.

Figure (2) illustrates the algorithm of the proposed FSBHWF.

Algorithm (2): Proposed FSBHWF

```
Input : bag of words ( features extraction stage)
Output: list of features with its weights for each class
Begin
Step1: For each class Cm                    // m =number of classes
 begin
  Step2: For each vector of weight of document
   begin
  Step3: sort the weight in the vector from maximum weight to
           minimum weight then do the followings
         begin
              set i=1, n= number of weight in the vector
              while  (i< n) do
                  begin
                      max=a[i]
                     For p=i to n do
                         begin
                               if a[p]>max then
                                begin
                                      swap(a[p],max)
                                      p=p+1
                                  end if
                           End for
                        max=a[i]
                       min= a[n]
                       For p= i to n do
                          begin
                                if a[p]<min then
                                  begin
                                      swap(a[p],min)
                                      p=p+1
                                  end if
                          End for
                       a[n]= min
                       i=i+1
```

```
                Print sorted vector
                End while
            End
 Step4:  assign the threshold = 1.
 Step5: take the feature in the first index and represent the
        document as vector contains one feature with its weight.
 End for
Step6: build a list contains the features with their weights of the
       documents for each class.
End for
End
```

Figure (2): Proposed features selection method

5. The Modified Model of Naïve Bayes Classifier

MMNB classifier is used to train the features of the documents so that it take the list of features without its weight TF-IDF for each class. Each document in that class is represented as vector of one dimension that fed to the NB classifier in order to train the features (the vector of feature that entered to NB is string only and not numerical feature TF-IDF, because the selected features give enough description that this features have high relevance with their classes and these features are important in the documents so become no need for their weights). Eq.(4) has been used to compute the prior probability and Eq.(5) used to compute the likelihood probability in order to train the features and as follow:

$$P(d|c) = P(ft \mid c) \quad (5)$$

where ft refers to feature without its weight TF-IDF (string only) and the probability of this feature is calculated from the number of appearance of this feature in the list of the selected features for each class.

After trained the selected features, the document need to be classified by assigning the correct class to it so that the MMNB classifier predicts the class with the maximum posterior probability to the document.

MMNB algorithm includes two phases: training phase and prediction phase (classifying).

Figure (3) illustrates the MMNB classifier algorithm.

Algorithm (3): MMNB

Input: list of training features for each class

Output: prediction class

Begin

// training

Step1: for n=1 to number of categorizes do

Begin:

Compute for each class the prior probabilities by utilizing Eq. (4).

Compute likelihood probability for the features given category by utilizing Eq. (5).

End

// prediction

Step2: Compute posterior probability P(c|d) of target category.

for n=1 to number of categorizes do

Apply Eq. (2)**.**

Step3: return the category that has maximum posterior probability.

End

Figure (3): Modified model of naïve bays

6. Implementation and Results

This section illustrates the result of each step in IDC system, the output of preprocessing step is set of tokens for each document after perform (tokenization, normalization, remove stop words, and stemming), the output of features extraction is vector of weight TF-IDF for each document, The result of proposed features selection method is one feature that has the highest weight TF-IDF for each document in the class and these selected features is stored in list contains the features with their weights TF-IDF for each class. This proposed method applied on the standard dataset and the collected dataset. Table (1) shows the result of samples of proposed features selection when applied on the standard dataset for five classes which are (business, entertainment, politics, sport and technical), this table refers to the features that are selected according to their highest TF-IDF weight from each document.

Table (1): The output of samples of proposed features selection when applied on the standard dataset for five classes

The selected feature	The weight of selected features TF-IDF	The class of features
tax	16.6363836933178	politics
museum	20.049636102047586	entertainment
export	6.321829022019839	business
fifa	17.7784404839026	sport
theatre	12.296649039991754	entertainment
lend	35.26360524616162	business
olympic	23.350921788663758	sport
song	137.73846918940572	entertainment
terror	16.768500434506908	politics
wifi	26.839232781530484	technical
spyware	19.328236230400726	technical
marathon	20.78268520778354	sport
dvd	37.92338286760264	technical
prison	33.299876205327834	politics
consume	11.830385092062336	business

Table (2) shows the result of samples of proposed features selection method when applied on the proposed dataset for six classes which are (Artificial Intelligent, Computer Network, Image Preprocessing, Biology, Chemical and Unknown). This table refers to the features that are selected according to their highest TF-IDF weight from each document.

Table (2): The output of samples of proposed features selection when applied on the proposed dataset for six classes

The selected feature	The weight of selected features TF-IDF	The class of features
network	46.53035929532783	computer network
embryo	38.72155534551992	biology
servo	65.28746215958836	unknown
compress	31.562813306607552	image preprocessing
aco	64.42686488335097	artificial intelligence
sulphate	46.834228495478875	chemical
switch	61.5251930597211	computer network
histogram	34.77879293529127	image preprocessing
pso	154.7147166406878	artificial intelligence
ecosystem	168.85758441770994	biology

hominid	250.6612192809122	biology
modular	55.838209426805186	chemical
lan	39.32643703207792	computer network
ionic	105.57413527551945	chemical
median	19.82194428706009	image preprocessing
robot	37.22360332345957	artificial intelligence
motor	116.8278731230019	unknown
transistor	56.602717749683706	unknown

And the result of MMNB classifier is the prediction class of testing documents depending on the maximum posterior probability**.** Table (3) illustrates the result of MMNB classifier for the standard dataset. This table refers to the name of testing documents, the real class of documents that must be classified to it, the predict class which is the result of MMNB classifier, and the description of the result.

Table (3): The result of samples of testing documents when applied MMNB classifier for the standard dataset

The name of documents	The actual class	The predict class	The result
350	politics	politics	true
57	business	business	true
35	sport	sport	true
328	entertainment	entertainment	true
511	sport	entertainment	false
282	technical	technical	true
506	business	entertainment	false
476	sport	sport	true
345	politics	politics	true
391	technical	technical	true

Table (4) illustrates the result of MMNB classifier for the proposed dataset. This table refers to the name of testing documents, the real class of documents that must be classified to it, the predict class which is the result of MMNB classifier, and the description of the result.

Table (4): The result of samples of testing documents when applied MMNB classifier for the proposed dataset

The name of documents	The actual class	The predict class	The result
particle swarm	Artificial intelligent	Artificial intelligent	true
Remote sense image	Image preprocessing	Computer network	false
Personal computer hardware	Computer network	Computer network	true
Basic computer network	Computer network	biology	false
Fuzzy image filtering	Image preprocessing	Image preprocessing	true
Welding procedure	unknown	unknown	true
Foundation of chemical reaction	chemical	chemical	true
biological concept	biology	biology	true
Universal intelligent abstract	Artificial intelligent	Artificial intelligent	true
PSA	unknown	unknown	true

7. Conclusions

This paper introduces a proposed system for documents classification based on their contents by using many steps. The preprocessing is important step in the IDC system, because its helps to obtain the clear words without noisy data which make the classification more accurate. Features extraction step helped to present the documents in the format that suitable to fed that documents to the classification algorithms. The proposed features selection method helped efficiently to select important features that used in classification process to achieve high accuracy of the IDC system and this method take less time, because only one feature is selected from each document. The result of features extraction step and the proposed features selection step which used in the training phase helped and facilitate to directly performed classification step after perform preprocessing step in the testing phase. The MMNB classifier helps to deal with the features (which are string only without its weight in the list). The IDC system was applied on two datasets, the first dataset is standard available on the internet which contains 2225 documents for five classes and the second dataset is collected during this work and contains 428

documents for six classes. The experimental results shows that the IDC system has achieved high accuracy, precision, recall and f1-measure was about 95.1%, 95%, 95.8%, and 95.39% respectively in the standard dataset while the accuracy, precision, recall and f1-measure in the collected dataset was about 95.3%, 95.16%, 95.83%, and 95.49% respectively.

References

[1] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee and Khairullah khan, "A Review of Machine Learning Algorithms for Text-Documents Classification", Journal of Advanced in Information Technology, Vol. 1, No. 1, 2010.

[2] Mahesh Kini M, Saroja Devi H, Prashant G Desai, and Niranjan Chiplunkar, "Text Mining Approach to Classify Technical Research Documents using Naïve Bayes", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 7, 2015.

[3] Dmitry Tsarev , Mikhail Petrovskiy and Igor Mashechkin, " Supervised and Unsupervised Text Classification via Generic Summarization", International Journal of Computer Information Systems and Industrial Management Applications. ISSN 2150-7988 Vol. 5, pp. 509-515, 2013.

[4] Rajni Jindal, Ruchika Malhotra and Abha Jain, " Techniques for text classification: Literature review and current trends " Webology, Volume 12, Number 2, December, 2015.

[5] Shugufta Fatima and Dr. B. Srinivasu, "Text Document categorization using support vector machine", International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 02, 2017.

[6] Suresh Kumar, "Optimization of Text Classification using Supervised and Unsupervised Learning Approach", M.Sc. thesis, Thapar University, 2015.

[7] Khushbu Khamar, "Short Text Classification Using kNN Based on Distance Function", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 4, April 2013.

[8] Anuradha Patra and Divakar Singh, "Neural Network Approach for Text Classification using Relevance Factor as Term Weighing Method", International Journal of Computer Applications (0975 – 8887) Volume 68– No.17, April 2013.

[9] Rajeswari R.P, Kavitha Juliet and Dr.Aradhana, " Text Classification for Student Data Set using Naive Bayes Classifier and KNN Classifier",

International Journal of Computer Trends and Technology (IJCTT) – Vol. 43, No. 1, January 2017.

[10] Pattan Kalesha, M. Babu Rao and Ch. Kavitha, " Efficient Preprocessing and Patterns Identification Approach for Text Mining", International Journal of Computer Trends and Technology (IJCTT) – volume 6 number 2– Dec 2013.

[11] Zhengyang Lu, "Web Page Classification Using Features from Titles and Snippets", M.sc. thesis, University of Ottawa, Ontario, Canada, 2015.

[12] Y. H. LI AND A. K. JAIN, "Classification of Text Documents", The Computer Journal, Vol. 41, No. 8, 1998.

[13] Vandana Korde and C Namrata Mahender, " Text Classification and Classifier: A Survey ", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, March 2012.

[14] Arzucan Ozgur, "Supervised and Unsupervised Machine Learning Techniques for Text Documents Categorization ", MSC thesis, B.S. in Computer Engineering, Bogazici University, 2002.

[15] M.IKONOMAKIS, S.KOTSIANTIS and V.TAMPAKAS, "Text Classification Using Machine Learning Techniques", WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4, 2005.

[16] Serkan Gunal, " Hybrid feature selection for text classification", Turk J Elec Eng & Comp Sci, Vol.20, No.Sup.2, 2012.

[17] Pinar Yildirim, "Filter Based Feature Selection Methods for Prediction of Risks in Hepatitis Disease", International Journal of Machine Learning and Computing, Vol. 5, No. 4, 2015.

[18] Moromi Gogoi and Shikhar Kumar Sarma, " Document Classification of Assamese Text Using Naïve Bayes Approach", International Journal of Computer Trends and Technology (IJCTT) – volume 30 Number 4 – December 2015.

[19] http://mlg.ucd.ie/datasets/bbc.html.

نظام تصنيف الوثائق الذكي

أ.م.د. حسنين سمير عبدالله* الباحث: هاله ضياء حسن*

المستخلص: هناك عدد هائل من الوثائق المتاحة في العديد من المصادر المختلفة في شكل غير منظم، ولذلك فإن هذه الوثائق الغير مهيكلة تحتاج إلى تصنيف. في هذا البحث، تم اقتراح نظام يسمى "نظام تصنيف الوثائق الذكي" الذي يمثل نظام لتصنيف الوثائق إلى الفئة الصحيحة استنادا إلى المعلومات النصية. هذا النظام يحتوي على أربع خطوات وهي المعالجة المسبقة، الاستخراج الميزات، طريقة مقترحة لاختيار الميزات وتحديث المصنفNaïve Bayes. في هذا النظام تم استخدام مجموعتي بيانات، مجموعة البيانات الأولى هي مجموعة البيانات القياسية والتي يحتوي على وثائق البحوث التقنية الموزعة على خمس فئات والتي تتوفر على شبكة الإنترنت، ومجموعة البيانات الثانية هي عبارة عن مجموعة تم تجميعها اثناء عمل هذا البحث والتي تحتوي على وثائق الكتب والموزعة على ستة فئات. حقق نظام إدك نتائج قوية. لمجموعة البيانات القياسية accuracy هي 95.1٪، وprecision هي 95٪، و recall هي 95.8 ٪، وf1-measure هو 95.39٪ في حين أن accuracy لمجموعة البيانات التي تم جمعها هي 95.3٪، وprecision هي 95.16٪، وrecall هي 95.83٪، و f1-measure هو 95.49٪.

الكلمات المفتاحية: تصنيف المستندات، الخلجان الساذجة، استخراج الميزات، اختيار الميزات، TF-IDF.

*الجامعة التكنولوجية