A Hash Map based Binary Matrix Approach for Text Document Classification

Suhail Afroz[1] and M. Hanimi Reddy[2]
[1] Assoc. Professor, CVR College Of Engineering / CSE Department, Hyderabad, India
Email: suhailafroz@cvr.ac.in
[2] Sr. Asst. Professor, CVR College Of Engineering / CSE Department, Hyderabad, India
Email: mh.reddy@cvr.ac.in

Abstract: **The conventional model uses the sequential approach for classifying the text document. In this paper, a new approach for the text document classification is proposed. The proposed method preserves the sequence of words that are occurring in a document. The data structure that is used in this method to preserve the word sequences is called "Binary Matrix". A classification technique is also proposed for classifying the text document. To index the terms, it uses Hash Map and this is associated with the list of class labels of the document in which the word is present.**

Index Terms: **Text Document, Hash Index, Hash Map, Binary Matrix, Classification**

I. Introduction

The World Wide Web (WWW) is widely distributed and dynamic information gallery. The total number of websites have grown from 130 in 1993 to 1 billion in 2016. The number of search queries per day started to skyrocket. In 1998 Google saw 9800 queries per day which grew to 40,000 searches per second and amounts to 3.5 billion searches per day [1] and these numbers are increasing rapidly every day.

Internet and corporate, spread across the global produces textual data in exponential growth, which needs to be shared, on need basis by individuals. If the data generated is properly organized, classified then retrieving the needed data can be made easily with least efforts. Hence the need of automatic methods to organize and classify the documents become inevitable. Due to such exponential growth in documents, the increase usage of the internet by the individual takes place.

Automatic classification refers to assigning the documents to a set of predefined classes based on the textual content of the document.

The classification can be Flat (or) Hierarchical

Flat Classification: In this there is no structure defined that specifies the relationship between the different documents. When the number of categories increase, it becomes difficult to search the category. [3, 7]

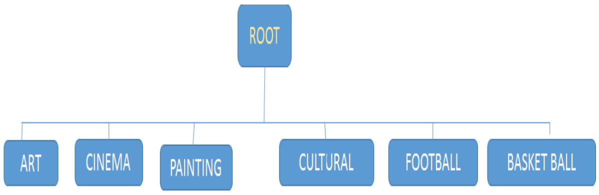


Figure 1. Flat classification

Hierarchical Classification: It uses Divide – and – Conquer approach. The classification can be repeated on the document in each sub category until it reaches to leaf and can be classified further. [3, 7]

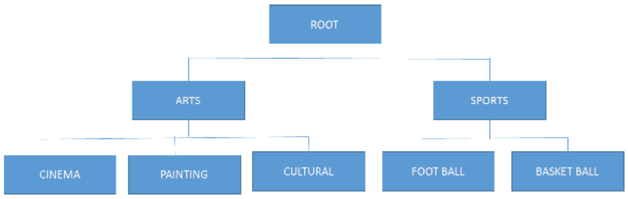


Figure 2. Hierarchical classification

Binary Classification: Initially Machine Learning is applied for binary classification where the classifier determines whether the document belongs to some pre – defined category or not. It is a kind of single label classification where the classifier decides whether the document belongs to a category or not. This kind of classification is useful to decide whether the mail received belongs to inbox or spam category. [6, 7]

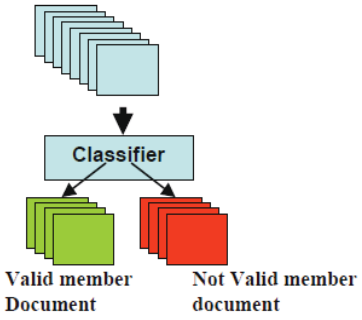


Figure 3. Binary classification

Single Label Multi Class Classifier: In some situations where there are more pre – defined categories present, the binary classifier needs to be modified for the classification of document into multi categories. [6, 7]

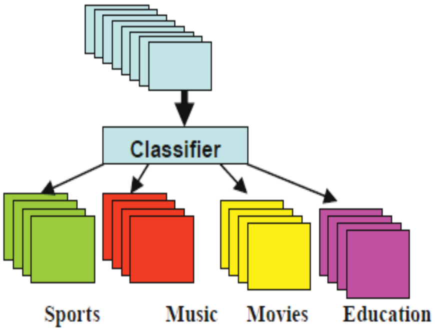


Figure 4. Multi-class single label classifications

Multi Class Multi Label Classifier: A single document may be classified into more than one class. This is called multi class or multi label classifier. [7]

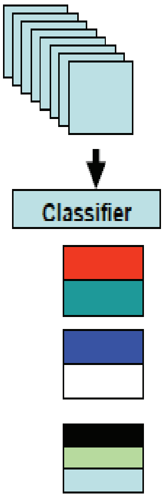


Figure 5. Multi-class multi-label classifications

II. Related work

A. Representation Models

The representation model for the text classification can be classified into two categories: those based on endogenous information of the given corpus (i.e. Content – based models) and those exploiting additional, external information in order to get more on textual information (i.e. Content – aware models). [6]

Vector Space Model (VSM) is an algebraic model for representing text documents as vectors of identifiers, such as, index terms. The major limitation of the VSM is that the correlation and context of each term is lost which is very important in understanding the document. [11]

Ontology Model can be used as a conceptual model which maintains a dictionary like structure for retrieving the information from the text document. It basically describes the document as a multi-dimensional vector that would encompass not only words in the document but also concepts. [10]

N – gram is a sequence of terms, with the length of N. Mostly, words are taken as terms. Each word from the document is represented as a set of overlapping N – gram. If N – gram is found in all the document, it gives no information about the context between the documents.

Latent Semantic Indexing (LSI) is one of the most popular linear document indexing methods which produces low dimensional representations using word co – occurrence which could be regarded as a semantic relationship between terms. It minimizes the reconstruction error (the Euclidean distance between the original matrix and its approximation matrix). The deficiencies of LSI are, it include few negative values in the reconstruction matrix which is difficult to explain by the model. [12]

Locality Preserving Indexing (LPI) is proposed for document indexing. Each document is represented as a vector with low dimensionality. Unlike LSI, which determines the global structure of the document space, LPI discovers the local structure and obtain a compact document representation sub space that best detects the essential semantic structure. LPI is based on the manifold theory. It tries to find the linear approximation to the Eigen function of the Laplace Beltrami operator on the compact Riemannian manifold. It is capable of discovering non – linear structure of the document space to some extent.

After the representation is given to the document then the classification task is performed which can classify the document to predefined category. Many statistical computational models are available and are developed based on the Naïve Bayes Classifier [5], K-NN Classifier [8], Centroid classifier [9].

III. Proposed Method

The proposed text documentation classification system avoids sequential matching of terms during representations and proposes to index the term in hash map. An efficient index scheme for preserving the sequence of occurrence of words in a text document a “Binary Matrix” is used.

Let there be c – classes and each having n number of documents, extract the words from each document. After doing some text processing, each word of the text document is labeled with the class in which the word is present. If the word is present in the text document of more than one class the label consists of all class names using hash map. [2]

The input to the proposed classifier is a text document and knowledge base processing is done on the text document to extract the words from it and constructs a Binary Matrix to decide for which class the document belongs to. The output of the classifier is the class to which the text document belongs.

Limitations:

1. If the length of the sub substring is same in multiple classes, FCFS method is used to label the document with class.
2. The tolerance factor if the length of the sub string is differing by atleast one is not addressed.

IV. Approach

In the proposed approach, a large data set is being used to verify the classifier. The hash map is used for indexing the term present in the document. Hash indexes are basically very much useful than B – Tree because we need to go all the way to the leaf node while searching in B – Trees. A suitable size of hash table will give a complexity of O(1) and is not the same with B – Tree where it is not a constant and has the complexity of O(log n). For representing the data in the proposed approach we use hash map with a key – value pairs where the key will be representing the terms present and value is corresponding to the class in which the document is present.

Let there be four classes C1, C2, C3 and C4 having documents. Assuming that, the following are list of terms extracted from all the documents of the different classes:

C1 = Brilliant, Performance, Excellent,
Republican, Underdogs
C2 = Republican, Country, Excellent, Splendid
C3 = Technology, Contribute, Common,
Champion, Lead
C4 = Quick, Champion, Excellent, Lead,
Technology

Table I shows the hash map constructed for the above classes

TABLE I.
HASH MAP CONTENT

Key	Value
Brilliant	C1
Performance	C1
Excellent	C1, C2, C4
Republican	C1, C2
Underdogs	C1
Country	C2
Splendid	C2
Technology	C3, C4
Contribute	C3
Common	C3
Champion	C3, C4
Lead	C3, C4
Quick	C4

Once the knowledge base is ready, the system can be given the test document or the query document as an input to the classifier. The classifier then creates a Binary Matrix for the test document and the knowledge base present. Binary Matrix is the one that contains only 0's and 1's as its entries. It has the dimension C * T_q [4]

Where, C is the number of classes and
T_q is the number of terms in the query / test document after pre processing

The hash map is then accessed to search for the occurrence of each term in the test / query document. If a term T_i exists in a particular class C_j, then the entry into the binary matrix will be 1 otherwise it is set to 0. If M is a binary matrix, it will be given as [4]

$$M_{ij} = \begin{cases} 1 & if\ T_i \in C_j \\ 0 & otherwise \end{cases}$$

Let the test / query document containing the following terms after processing

T_d = Brilliant, Champion, Technology,
Republican, Contribute, Quick

Table II shows the binary matrix with the entries for the test document words against the knowledge base.

TABLE II.
BINARY MATRIX STRUCTURE

	Brilliant	Champion	Technology	Republican	Contribute	Quick
C1	1	0	0	1	0	0
C2	0	0	0	1	0	0
C3	0	1	1	0	1	0
C4	0	1	1	0	0	1

Once the Binary Matrix is constructed it will be containing the value 1 for the terms that are identified in a specific class. These 1's are representing the substring available in the given test document.

In order not to lose the correlation and semantic of the sentence all the consecutive 1's i.e. the longest substring will be identified in the binary matrix. Once the longest substring is identified then the corresponding class in which the similar substring is present will be labelled as the class for the test / query document.

V. SYSTEM ARCHITECTURE

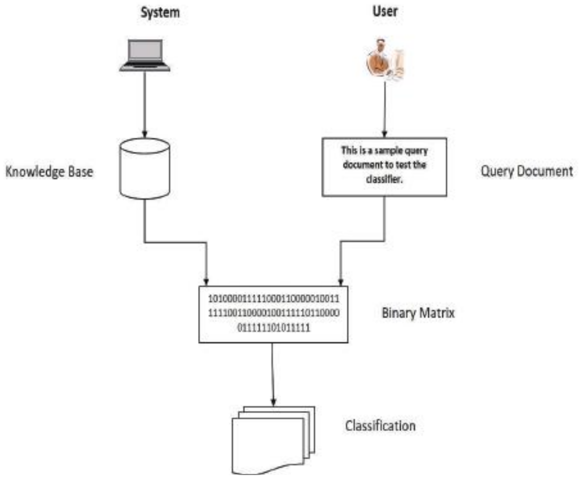


Figure 6. Architecture of Classifier

VI. Research Methodology

A. *Knowledge Base Creation*:

Let there be K number of classes (C1, C2, C3, . . ., CK), each containing n number of documents.

```
Step 1: Extract all the words from all the documents in
            the class
Step 2: After extracting the words from a class remove
            stop words and perform stemming to drop
            unnecessary words.
Step 3: Remove the duplicate words after Step 2.
Step 4: Store unique words into hash map with its class
identification. i.e. <word – class> pair.
If the word already exists in the hash map, simply add a new
class name to the <word – class> pair, otherwise add a new
word and its class pair to the hash map for the first
occurrence of the word.
Step 5: Repeat Step 1 to Step 4 to process all classes.
```

Algorithm 1. Knowledge base creation

B. *Binary Matrix Construction:*

After the knowledge base is created, it is accessed for constructing the binary matrix.

```
Step 1: Extract the words from the test / query document
Step 2: After extracting the words from a class remove
            stop words and perform stemming to drop
            unnecessary words.
Step 3: Binary Matrix is constructed with both knowledge
            base classes and extracted words of the test
            document.
Size of Binary Matrix = Number of Knowledge Base
Classes * Number of Extracted Words in Test Document
Step 4: Depending on the occurrence of the word in a
            particular class place an entry into the binary
            matrix as 1 or 0
```

Algorithm 2. Binary Matrix Construction

C. *Finding Class:*

After the binary matrix is constructed containing 0’s and 1’s, identify the class for the test document. Each row of binary matrix will be representing a binary string.

```
Step 1: Look for a row with a longest substring containing
only 1’s
Step 2: Declare the corresponding class as the class the test
document belongs to.
```

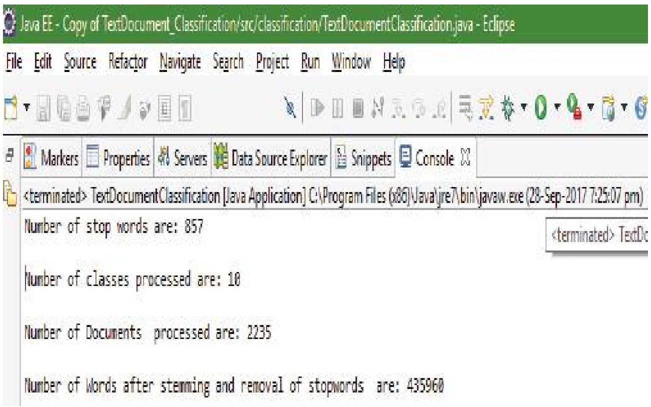
Algorithm 3. Finding a class for the test / query document

VII. Experimental Setup

The dataset that is used for the experiment is taken from BBC dataset. The test run was conducted using the data set against classifier. The test results are as shown.

A. *Preprocessing*:

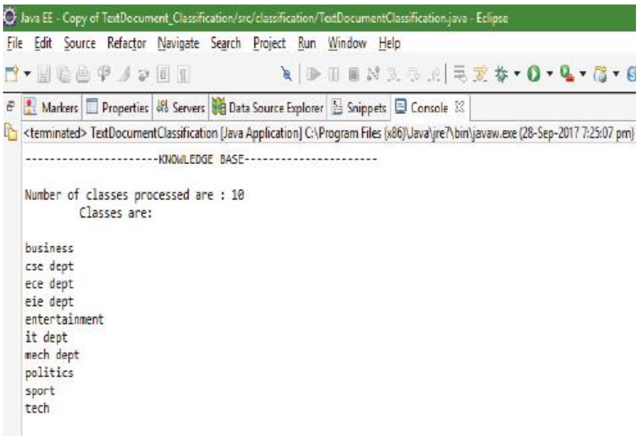
Simple text processing is done on the dataset available to remove the stop words and perform stemming on the text documents.


Java EE - Copy of TextDocument_Classification/src/classification/TextDocumentClassification.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console
<terminated> TextDc

```
<terminated> TextDocumentClassification [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (28-Sep-2017 7:25:07 pm)
Number of stop words are: 857

Number of classes processed are: 10

Number of Documents  processed are: 2235

Number of Words after stemming and removal of stopwords  are: 435960
```

Figure 7. Pre Processing of the Classes

B. *Phase 1: Knowledge Base Creation*

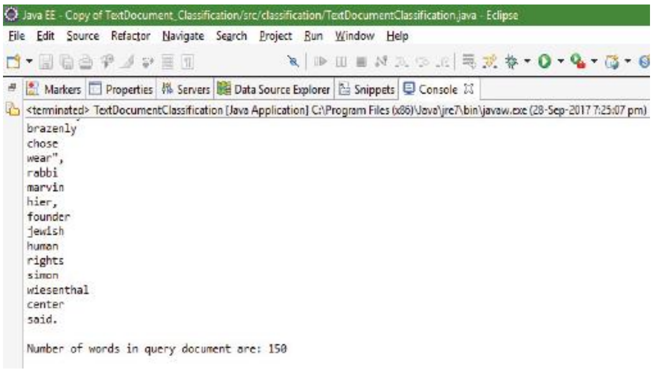

Java EE - Copy of TextDocument_Classification/src/classification/TextDocumentClassification.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console

```
<terminated> TextDocumentClassification [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (28-Sep-2017 7:25:07 pm)
---------------------KNOWLEDGE BASE---------------------

Number of classes processed are : 10
        Classes are:

business
cse dept
ece dept
eie dept
entertainment
it dept
mech dept
politics
sport
tech
```

Figure 8. Knowledge representing no. of classes

C. *Phase 2: Binary Matrix Construction*

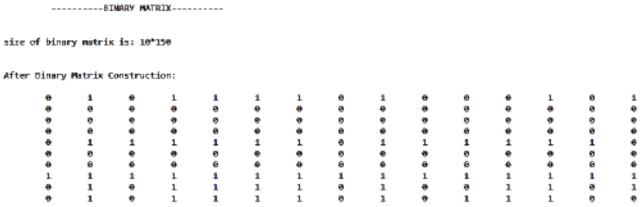
After the Knowledge Base is ready the classifier is given this and the test document as input. The classifier then extract the words available in the test / query document.


Java EE - Copy of TextDocument_Classification/src/classification/TextDocumentClassification.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console

```
<terminated> TextDocumentClassification [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (28-Sep-2017 7:25:07 pm)
brazenly
chose
wear",
rabbi
marvin
hier,
founder
jewish
human
rights
simon
wiesenthal
center
said.

Number of words in query document are: 150
```

Figure 9. No. of words extracted from test document

Then after the words are extracted from the query document the Binary Matrix is constructed with the required dimension.

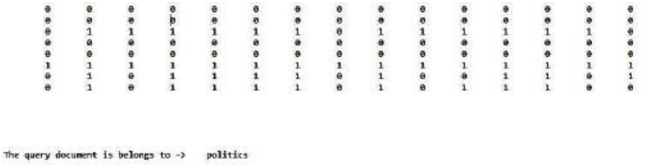


```
          ----------BINARY MATRIX----------

size of binary matrix is: 10*150

After Binary Matrix Construction:

    0    1    0    1    1    1    1    0    1    0    0    0    1    0    1
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    1    1    1    1    1    1    0    1    1    1    1    1    1    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    0    1    0    1    1    1    1    0    1    0    0    1    1    0    1
    0    1    0    1    1    1    1    0    1    0    1    1    1    0    0
```

Figure 10. Binary Matrix for the test data

D. Phase 3: Labeling the Class

Later the classifier will identify the longest substring from the binary matrix and assign the test/query document with the class label.



```
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    1    1    1    1    1    1    0    1    1    1    1    1    1    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    0    1    0    1    1    1    1    0    1    0    0    1    1    0    1
    0    1    0    1    1    1    1    0    1    0    1    1    1    0    0


The query document is belongs to ->    politics
```

Figure 11. Class labeling

VIII. Conclusions

A new text document classification algorithm using hash indexing is proposed that basically makes use of a new data structure called "Binary Matrix" which is used to preserve the sequence of terms in the test document. Even though the term sequence of the test document is preserved the algorithm did not preserve the sequence of terms of the training data. In order to speed up the classification the representation model and the data structure are very useful. Addition of the new classes and deletion of the existing classes can also be performed easily using hash mapping.

The experiment is also performed on the New Groups articles where its results were accurate.

References

[1]. www.internetlivestat.com/google-search-statistics

[2]. B S Harish, S Manjunath and D S Guru: Text Documentation Classifiation: An Approach Based On Indexing. International Journal of Data Mining & Knowledge Management Process Vol.2, January 2012.

[3]. Axim Sun and Ee – Peng Lim, Hierarchical Text Classification and Evaluation, In the Proceedings of the IEEE International Conference on Data Mining, Pages 521 – 528, California, USA, November 2001.

[4]. R.Dinesh,B S Harish,D S Guru and S Manjunath.: Concept of status matrix in classification of text documents.4th Indian International Conference on Artificial Intelligence(IICAI-09).

[5]. McCallum, A., Nigam, K.: A Comparision of Event Models for Naïve Bayes Text Classification. Journal of Machine Learning Research, vol.3, pp. 1265 – 1287 (2003).

[6]. Li, Y.H., Jain, A.K.: Classification of Text Documents. The Computer Journal. vol. 41, pp.537--546 (1998).

[7]. Mohammed.Abdul.Wajeed, T.A.Adilakshmi: Text Classification using machine learning. Journal of Theoretical and Applied Information Technology, pp.119-123, 2005-2009.

[8]. Sadegh Bafandeh Imandoust And Mohammad Bolandraftar: Application of K - Nearest N–ighbor (KNN) Approach for Predicting Economic Events: Theoretical Background, Vol.3, pp 605 – 610, Sep – Oct 2013.

[9]. Zehra Cataltepe, Eser Aygun Ayazaga and Sariyer: An Improvement of Centroid – Based Classification Algorithm for Text Classification, ICDE, 2007.

[10]. Kuwar Aditya, Bhalekar Arjun and Bade Ankush: An Ontology Based Text Mining, International Journal of Engineering Trends and Technology (IJETT) – Vol. 10, April 2014.

[11]. Jitendra Nath Singh and Sanjay Kumar Dwivedi: Analysis of Vector Space Model in Information Retrieval, National Conference on Communication Technologies & its impact on Next Generation Computing CTNGC 2012, Proceedings published by International Journal of Computer Applications (IJCA).

[12]. Hemalata Tekwani and Mahak Motwani: Text Categorization Comparison between Simple BPNN and Combinatorial Method of LSI and BPNN, International Journal of Computer Applications, Vol. 97, July – 2014.

[13]. mlg.ucg.ie/datasets/bbc.html (for BBC Dataset)

[14]. http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html (for News Groups Dataset)