Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 8, Issue, 06, pp.32474-32477, June, 2016

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

AUTOMATED TEXT ANALYTICS AND CLASSIFICATION OF TEXT DOCUMENTS WITH MACHINE LEARNING

Nihar Ranjan, Abhishek Gupta, Ishwari Dhumale, Payal Gogawale and *Rugved Gramopadhye

Department of Computer Engineering, Sinhgad Institute of Technology and Science, Narhe, Pune, India

ARTICLE INFO

Article History:
Received 23rd March, 2016
Received in revised form
15th April, 2016
Accepted 26th May, 2016
Published online 15th June, 2016

Key words:

Text Mining,
Text Categorization,
Dynamic learning.

ABSTRACT

With the dramatic increase of digital form of data, it is difficult to manage the huge amount of documents. Whenever any individual tries to find any information about particular topic, he may receive a large set of documents on the internet. Some of these documents may be in .pdf format some may be in .txt format or simply any word document. The title of these documents may seem relevant to what the individual is looking for but the content in those documents may differ. Thus there was a necessity to read, understand and analyze contents of all the documents at one glance. As a result, it has become necessary to categorize large texts (documents) into specific classes. In our propose system we are classifying the documents, both single and multiple documents into predefined classes. The documents can be of any form i.e .txt, .doc, .docx, .pdf. Then preprocessing techniques are used like tokenization, stop words removal, stemming on input files. The document is classified according to the given learning. Dynamic learning is used to update the learning datasets. This project covers how the classification of document is done and how exactly the desired output is determined (classified documents). We also aimed at generating a classification report of number of documents in a particular class with respect to total number of documents. The pie chart can also showcase why a particular document is inclined towards any particular category and what percentage of its content consists of related information towards that category.

Citation: Nihar Ranjan, Abhishek Gupta, Ishwari Dhumale, Payal Gogawale and Rugved Gramopadhye, 2016. "Automated text analytics and classification of text documents with machine learning", *International Journal of Current Research*, 8, (06), 32474-32477.

INTRODUCTION

Text mining is a variation on a field called data mining (Navathe *et al.,* 2000), that tries to find interesting patterns from large database. Nowadays through the sudden growth in digital world and available documents, the task of organizing text data becomes one of the principle problems. To classify millions of text document manually is an expensive and time consuming task. Therefore, automatic text classifier is constructed using pre-classified sample documents whose accuracy and time efficiency is much better than manual text classification. In this paper we summarize text classification techniques that are used to classify the text documents into predefined classes (Christoph Goller *et al.,* 2009). A major approach is text categorization, the task which tries to automatically categorize documents into their respective classes. This paper gives the overview of how exactly the documents are classified. Initially bunch of documents of different types will be accepted then data preprocessing will be done.

Corresponding author: Rugved Gramopadhye,
Department of Computer Engineering, Sinhgad Institute of Technology and Science, Narhe, Pune, India.

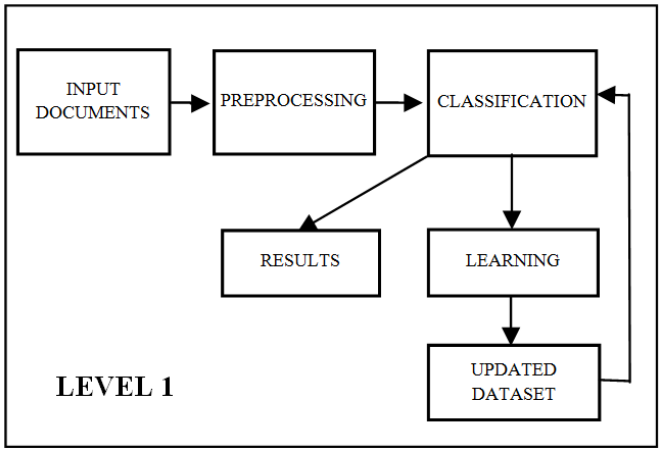
Preprocessing is termed as an important step in data mining process. Data preprocessing includes cleaning, normalization, transformation etc. and the product of data preprocessing is the final training set. Classification of documents is done according to learning given. Dynamic learning is included as well. New words identified in documents are updated accordingly in the dataset of that particular class. Graphical representation is done at the end describing the percentage of the document classified in a particular class.

Overview System Architecture

Our system architecture depicts that the input of documents (bunch of files or single file or folder) will be given by the user. The Files will be in any format like .txt,.doc,.docx,.pdf. Then preprocessing techniques will be applied on the documents. Once the preprocessing is done then the documents will be classified i.e. classification will be done based on the content of the document. The document will be classified based on the learning given. The datasets are created in the form of .txt files and learning is provided accordingly. Dynamic learning is also added if new words are to be added automatically in the datasets. Whenever a new word related to

a particular class is identified then dataset gets updated dynamically.

The diagram below shows the overall architecture of our system:



Level 1. Architecture

Preprocessing

The main objective of pre-processing is to obtain the key features or key terms from stored text documents and to enhance the relevancy between word and document and the relevancy between word and category. Pre-Processing step is crucial in determining the quality of the next stage, that is, the classification stage (Nalini and Jaba Sheela, 2004). The figure depicts the working of the preprocessing in our project. In preprocessing the input document will be given by the user. 3 main techniques are used under preprocessing:

- Tokenization
- Stop word removal
- Stemming

Tokenization

Once the input document is given tokenization is done on the documents. In tokenization the sentences of the documents are chopped into pieces, called tokens, and at the same time throwing away certain characters, such as punctuation.

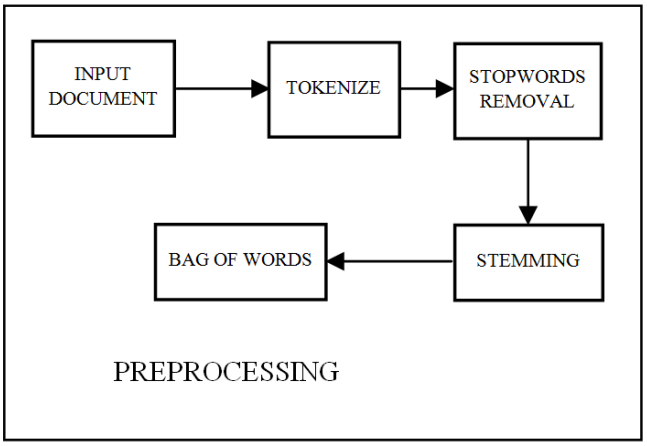
Stopword Removal

Once the tokens are generated then stop word removal technique is applied. The unwanted tokens such as articles, punctuation are removed. Ex: a, an, the, are, is, from, has, he, she, it etc.

Stemming

When the stop words are removed then stemming technique is applied Stemming basically means reducing different grammatical forms or word forms of a word like its noun, verb, adjective etc. to its root form. We can also say that the goal of stemming is to reduce inflected (or sometimes derived) words to their common root form. It is important step in Information Retrieval (IR) as well as in our system too. Usually, stemming algorithms can be classified into three groups: truncating methods, statistical methods, and mixed methods (Deepika Sharma, 2012). Stemming is actually done by removing the suffixes and prefixes (affixes) from the word.

The diagram below explains how the preprocessing techniques are used in our project.



Level 2. Architecture

Classification and Learning

Once the preprocessing is done bag of words are created. These bags of words are then classified according to the learning given. After classification is done result of the classification is displayed. If in case any new words are detected of a particular class then dynamic learning is applied and the new word is added in the dataset of that particular class. Unigram frequency is used for dynamic learning.

Unigram Frequency

The unigram posits that each word occurrence in a document is in dependent of all other word occurrences. I.e. we can think of the document generation process as a sequence of dice rolls, where there is a fixed probability of occurrence associated with each word (Jason, 2005).

-a-aaa-abba-abc-ac-ad-ada-add-baa-bad-cab-cb-cdc-dab-dad-dada-dc-

Figure 3. Example Text for Invented Language

You may collect unigram frequencies in tables using either tallies or percentages. From Figure 3, count the number of times '-', 'a', 'b', 'c', and 'd' each appear. Each character count produces a tally, tabulated in Figure 4. Compute each character's frequency as a ratio of the number of times that character appears and the total number of characters. You may tabulate the frequencies as ratios, as shown in Figure 5. These tables are called unigram tables**.** Although the tables might appear two dimensional, the numbers are in a single row. You may choose to count characters in any order, so the tables in each pair are equivalent. However, note that the count and

percent frequencies differ! To access a frequency for a particular character, use the notation freq j, where *j* is any character from the character set, including'-'. For example, freq a=31 tells you that 'a' occurs 31% of the time.

-	a	b	C	d
17	20	8	7	12

Or

-	a	b	C	d
17	12	8	7	20

Figure 4. Unigram Frequencies (Tallies)

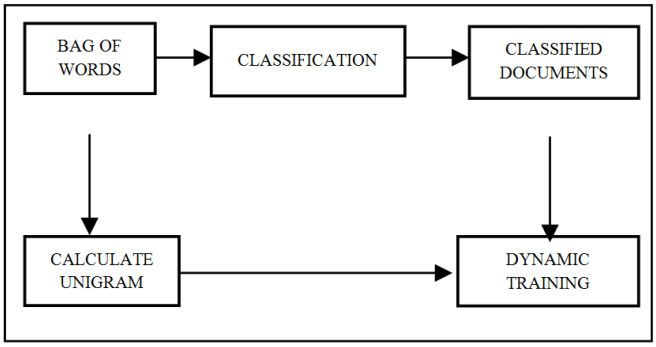
-	a	b	C	D
17	12	8	7	20

Or

-	a	b	C	D
17	12	8	7	20

Figure 5. Unigram Frequencies (Percentages (%))

Thus, the diagram below explains about the classification and learning done in our project.



Algorithm Proposed

Algorithm for Preprocessing
Input

A.Input File Path
B.Dataset for Stop words

Output

A. A set of unique words present in Input file.
B. Count of every word.

1. Loading the Dataset of Stop words in an array & calculating the number of Stop words.
2. Open the input file and access it using an object f.
3. Remove all punctuation marks from the file
4. Store the words present in the file in a list
5. For each word present in the list

   - For each stop word
   - If(Both word and stop word are same)
   - Set the flag f1.
   - Break.
   - Else
   - Reset the flag f1.
   - If (Flag f1 is reset)
   - If(word ends with s)
   - Remove the last bit.
   - Add the remaining word in word list.
   - Else If(word ends with ‘s)
   - Remove the last 2 bits.
   - Add the remaining word in word list.
   - Else
   - Add the remaining word in word list.

6. For every unique word in word list

   - Calculate the Unigram frequency of the word present in list.
   - Calculate the Unigram percentage and store in Hash map with respective to the word.

Algorithm of Classifier
Input:

A.Trained data.
B.Input File Path
C.A set of unique words present in Input file.
D.Count of every word.

Output:

A. Assigned Class.
B. Dynamically update trained data.
C. Files get stored in the various folder acc. to their classes.

1. Accessing the trained data.
2. Loading the trained data in list for every class.
3. Loading the unique words in another list.
4. For every class

   - For every unique word in list
   - For every unique data in trained data list of specific class
   - If(data==word)
   - Increment the counter for Specific class.
   - Break;

5. If(politics>science && politics>health && politics>weather)

   - Training class politics
   - Create the folder namely “Politics” and add the file in it.

6. Else If(science>politics && science>health && science>weather)

   - Training class science

- Create the folder namely "Science" and add the file in it.

7. Else If(health>politics && health>science && health>weather)

   - Training class health
   - Create the folder namely "Health" and add the file in it.

8. Else If(weather>politics && weather>science && weather>health)

   - Training class weather
   - Create the folder namely "Weather" and add the file in it.

9. Else

   - Create the folder namely "Others" and add the file in it.

RESULTS AND DISCUSSION

The proposed algorithm is classifying the given set of documents into predefined categories. The effectiveness of a proposed algorithm is evaluated using performance measures like precision, recall and F-measure. The standard definitions of precision, recall and F-measure are as follows:

$$Precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

$$Recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relavant\ documents\}|}$$

$$F \quad measure = 2.\frac{precision.recall}{precision + recall}$$

To calculate performance measure we are using standard dataset i.e. 20-newgroup dataset. The documents of different classes and different formats like .txt, .doc, .docx, .pdf canbe given to system as input.

Following table shows the precision, recall and F-measure for proposed algorithm

No. of files	Precision	Recall	F-measure
3000	2996/3000: 0.998	2996/3000: 0.998	0.998

Conclusion

This Paper explains the working of the classification of the documents in detail. We have allowed different types of input files like .txt, .doc, .pdf, .docx. These files can be given as a single file, multiple files as per user's selection or even a whole directory consisting any number of files. The work in designing our system is motivated by the belief that such implementation can help user in classification of large amount of documents. Every minute thousands of new files are being uploaded on internet. Our system just makes it easy to differentiate these files so that they remain organized and easy to access. The accuracy of our system is above average as we have tested it on more than thousands of documents. The great part is, this accuracy gets better and better as we process more and more documents due to its machine learning technique. Analysis part is done in the end which gives a classification report after the output in the form of pie chart. This report is very essential as it helps user understand system in very simple yet efficient language which can be understood easily even if end user has no technical knowledge about our system. This paper hence provides the overview of working of the classification model.

REFERENCES

Christoph Goller, Joachim Löning, Thilo Will and Werner Wolff, 2009, "Automatic Document Classification: A thorough Evaluation of various Methods", "doi=10.1.1.90.966 ".

Deepika Sharma, Stemming Algorithms, A Comparative Study and their Analysis,International *Journal of Applied Information Systems (IJAIS)* –ISSN : 2249-0868, Foundation of Computer Science FCS, New York, USA, Volume 4–No.3, September 2012 –www.ijais.org.

Nalini, K. and Dr. L. Jaba Sheela, 2004. "Survey on text classification", *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, ISSN: 2349-2163, Volume 1 Issue 6 (July 2014).

Navathe, Shamkant B., and ElmasriRamez, 2000. "Data Warehousing And Data Mining", in "Fundamentals of Database Systems", Pearson Education pvtInc, Singapore, 841-872.

The Unigram Term Frequency, Jason D. M. Rennie, jrennie@gmail.com , June 18, 2005.

www.cs.cornell.edu/courses/cs100/2000sp/p7/node16.html.

****