(IJIEST) 2020, Vol. No. 6, Jan-Dec **e-ISSN: 2454-9584; p-ISSN: 2454-8111**

Evaluation of Text Classifier Based on Different Stemming Algorithms

Assistant Lecturer. Ebtehal Talib Kudair
Ministry of Higher Education and Scientific Research

ABSTRACT

Text classification is an important field of machine learning, is a supervised learning method and it depends on dividing texts into groups according to the predefined categories. In general, the text carries a lot of information but in an unstructured form, and this unstructured data must be converted into structured data. In this paper, texts will be classified using the traditional k-Nearest Neighbor algorithm (KNN), and the performance of the KNN classification algorithm will be compared through text preprocessing with the use of different stemming algorithms such as (Porter Stemmer, Snowball Stemmer). The snowball stemmer reduced the number of features in comparison with porter stemmer, thus the results proved that the classifier are more accurate when using snowball stemmer.

Key words: *Text Classification, k-Nearest Neighbor algorithm, Stemming.*

1. INTRODUCTION

There are many applications in the text mining field, one of the most important of which is text classification. An important example of text classification is the classification of electronic articles into predefined categories Such as art, sports, literature and politics[4]. The whole texts are dividing into two groups (training and testing). The text classification process begins with classify the training group into the previously defined categories. The next step in text classifying is to create a model that has ability to assign each text into a specific category [4].

To implement text classification, we need to understand how natural languages are processed [3]. Text classification helps us to recognize between a huge numbers of information, to choose according to our interests. In text classifying there are many important terms like (word (feature or term), Document (collection of words), Class (category), data set (training and testing)) [4].

In this paper, build of a text classifier by integrating k-Nearest Neighbor algorithm (KNN) and inverse term frequency (TF-IDF) for feature selection and using two stemming algorithm (porter and snowball) to analysis how to get a good accuracy.

2. BACKGROUND AND PROPOSED TEXT CLASSIFIER SYSTEM

The proposed classifier will be built through several stages (figure (1) show the flowchart) namely: Preprocessing, feature Extraction and Selection, Classification using KNN algorithm and Performance evaluation to show which stemming algorithm is the best.

2-1- Preprocessing

It is the first stage in the proposed system, through it dividing the text into words, and then examines these words by comparing them with the group of stop words in order to delete them [3]. After that, the stemming algorithms are implemented for reducing the features. When applied the (porter and snowball) stemmer, all the words return to their root. For example, the word (bank) is the root for words (banks, banked, banking and banker) [2].

2-2- Feature Extraction and Selection

It is the most important step in implementation the classifier, in this step the words that represent the text are identified through (representation of document and feature weight)[8]:

- Representation of Document: In this step, Vector Space Models was implemented to represent the document, each word in a text is considered as feature. Thus, the number of features in the vector is equal to the number of words in text after preprocessing step [3].

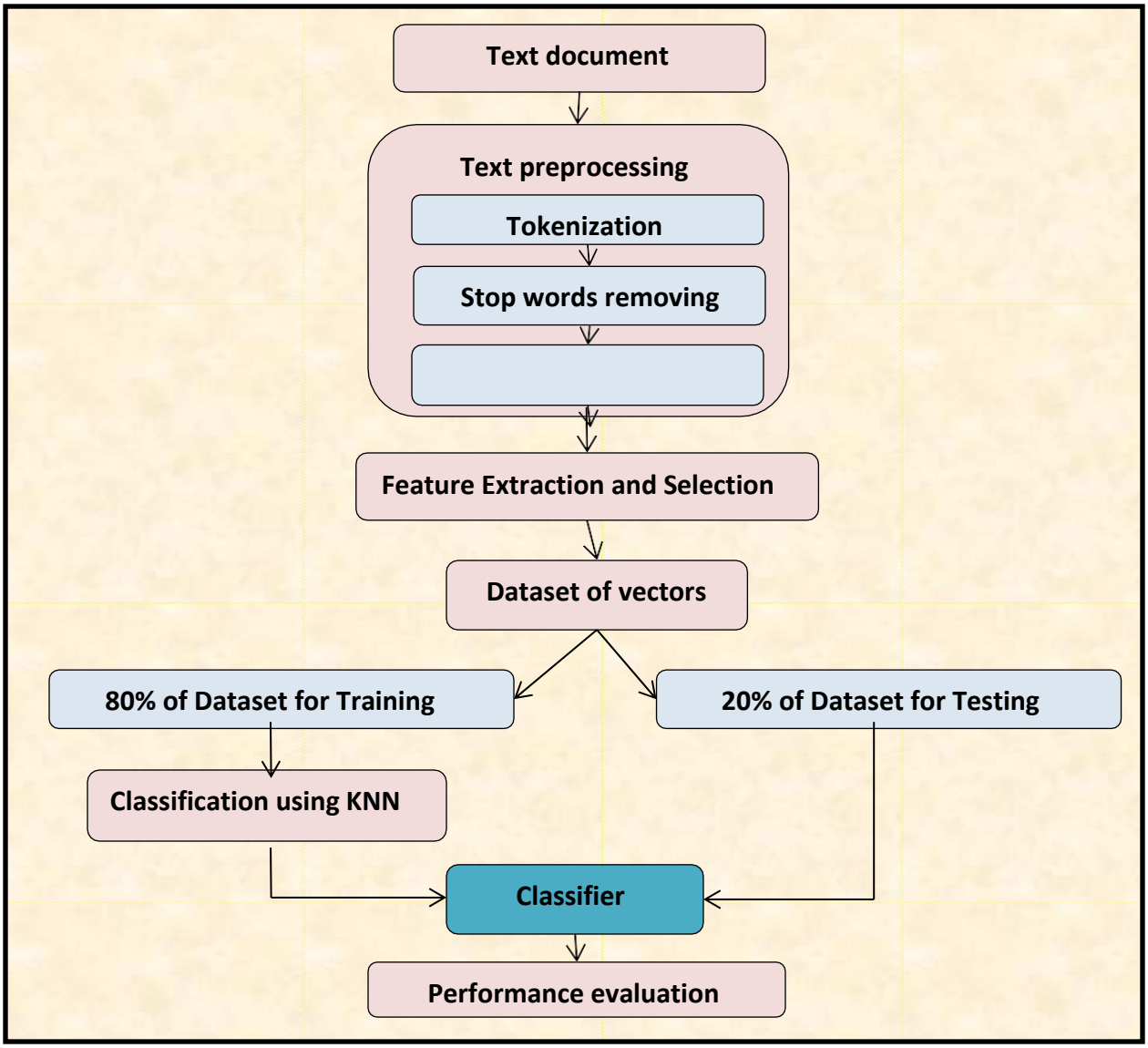
- Feature Weight: In this step, the text is processing through converted it into a vector of numbers. After the features are identified, each feature replace by number, this number represent feature weight [6]. In this paper "Term Frequency/Inverse Document Frequency" are used to represent features weight according to this equation (1)[6]:

$w_{ij} = tf_{ij} \log(N/df_i)$……. (1)

Where:
tf : the number of a word occurrence in a document. df: the number of documents containing that word.
N: is the number of documents.

Then a group of sub-features that represent the text is chosen from the total features, in order to reduce the dimensions that represent the text.

Figure (1) the proposed text classifier system diagram



2-3- Classification using KNN algorithm

The k-Nearest Neighbor algorithm (kNN) is an objects classifier, k-NN is lazy learning because it does not truly do anything trough the training phase, which is based on relative closed training data[1]. This algorithm (look algorithm (1)) is classified the object into the most common class among the closest neighbors (k). K is integer, positive and typically small. This algorithm is based on the distance function, and usually uses the Euclidean distance or cosine similarity measure [5]. In this paper the Euclidean distance measure is used.

Algorithm (1): KNN Algorithm

Input X = {A_1, A_2, xA_3, ..., A_n} *#set of n vectors with several attributes*

Output: Model is ready

Step-1: Select (K) the number of the neighbors

Step-2: Calculate the Euclidean distance *(Eq.2)* of K number of neighbors

$$similarity = \frac{A*B}{\|A\|_2\|B\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (2)$$

Step-3: According to the Euclidean distance take the closest K neighbors

Step-4: count the number of the data points in each category, for these k neighbors

Step-5: Set the new data points for that category whose number of neighbor is the maximum.

2-4- Performance Evaluation

The performance of the KNN algorithm measured by using Precision and Recall (look equation (3 and 4). Comparison the classifier with using different stemming algorithms to show which stemming algorithm is the best[7].

Precision (P) = (tp) / (tp + fp)..... (3)
Recall (R) = (tp) / (tp + fn)..... (4)

Where:

tp: the number of true positives fp: the number of false positives

fn: the number of false negatives

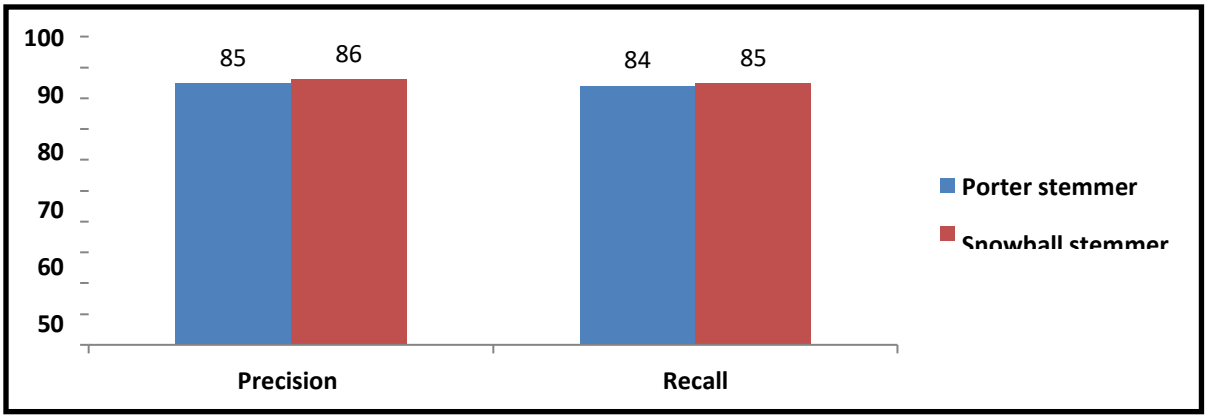
3. RESULTS AND DISCUSSION

In this paper, used the average values of precision and recall evaluating the performance of our classifier and tested our approach on 'comp.graphics', 'soc.religion.christian', 'alt.atheism' and 'sci.med' categories of 20 Newsgroup dataset. The results (Table (1)) showed that the KNN classifier works better with snowball stemmer compared to porter stemmer (look figure (2) too).

Table (1): Results of Implementation of KNN algorithms through the comparison between the Porter Stemmer and Snowball Stemmer

	Porter stemmer	***Snowball stemmer***
Precision	**85%**	**87%**
Recall	**84%**	**85%**

Figure (2) Comparisons of Performance for Porter and Snowball Stemmer



4. CONCLUSION

In this paper, comparison different stemmer (snowball and porter) by using TF/IDF method for feature selection and KNN classifier, the results showed that Snowball Stemmer improves the performance of the classifier. Snowball Stemmer reduces number of features, thus the classifier is faster in comparison by using porter stemmer.

REFERENCES

1. Bijalwan, Vishwanath & Kumar, Vinay & Kumari, Pinki and Pascual, Jordan (2014) ***" KNN based Machine Learning Approach for Text and Document Mining"*** International Journal of Database Theory and Application Vol.7, No.1.

2. Jivani, Anjali Ganesh (2016) ***"A Comparative Study of Stemming Algorithms"*** Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938.

3. Kannan, S. and Gurusamy, Vairaprakash (2015) ***"Preprocessing Techniques for Text Mining"*** Conference Paper.

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

5. Miah, Muhammed ***"Improved k-NN Algorithm for Text Classification"***, Department of Computer Science and Engineering University of Texas at Arlington, TX, USA.

6. Qaiser, Shahzad and Ali, Ramsha (2018) ***"Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents"*** International Journal of Computer Applications (0975 – 8887) Volume 181 – No.1, July.

7. Sulaiman, M.N and Hossin, M. (2015) ***"A Review on Evaluation Metrics for Data Classification Evaluations"*** International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.2, March.

8. Venkatesh, B. and Anuradha, J. (2019) ***"A Review of Feature Selection and Its Methods"*** Cybernetics and Information Technologies, Volume 19, No 1.