ISSN:0975-9646

Building a K-Nearest Neighbor Classifier for Text Categorization

A.Kousar Nikhath[1], K.Subrahmanyam[2], R.Vasavi[3]
[1]*Research Scholar, CSE Department , KL University, GUNTUR.*
[2]*Professor, CSE Department, KL University, GUNTUR.*
[3]*Assistant professor, VNRVJIET, HYDERABAD.*

Abstract-Text categorization is a process of assigning various input texts (or documents) to one or more target categories based on its contents. This paper introduces an email classification application of text categorization, using k-Nearest Neighbor (k-NN) classification[1]. In this work text categorization involves two processes: training process and classification process. First, The training processes use a previously categorized set of documents to train the system to understand what each category looks like[1].Second,the classifier uses the training 'model' to classify new incoming documents.The k-Nearest Neighbor classification method makes use of training documents, which have known categories, and finds the closest neighbors of the new sample document among all[2]. These neighbors enable to find the new document's category. The Euclideandistance has been used as a similarity function for measuring the difference or similarity between two instances[3].

Key words–Text categorization, machine learning, k-NN algorithm, similarity function

I:INTRODUCTION

Text categorization is the process of grouping text documents into one or more predefined categories based on their content. A number of statistical classification and machine learning techniques have been applied to the text categorization [4]. Among all methods, k-Nearest Neighbor is considered to be the best method for text categorization. The k-NN classification requires some initial files that are generally called as training documents [6]. These documents' categories are known before method is applied[1]. A new document with no category yet, is compared with all these training documents with respect to the terms they share. Finally, the documents, that the new document is closest to, are determined, and the category is assigned accordingly. Text categorization will give better results if machine learning is included in the approach, in such a way that newly categorized documents, could be added to the training documents set, together with their previously determined categories.

II:RELATED WORKS

2.1Classification

Classification is a data mining technique that maps data into predefined groups or classes. It is a supervised learning method which requires labeled training data to generate rules for classifying test data into predetermined groups or classes. It is a two-phase process. The first phase is the learning phase, where the training data is analyzed and classification rules are generated. The next phase is the classification, where test data is classified into classes according to the generated rules. Since classification algorithms require that classes be defined based on data attribute values[5].Text classification (TC) is a Natural Language processing (NLP) problem; it can be defined as the assignment of unclassified document to one or more predefined categories based on their content[6]. Automatic TC is very useful in terms of time and expense that it saves, many methods and algorithms have been applied to the problem of text categorization; these methods vary in their accuracy and computation efficiency.

Automatic document classification tasks can be divided into three sorts: supervisedocument classification where some external mechanism (such as human feedback) provides information on the correct classification for documents, unsupervised document classification (also known as document clustering), where the classification must be done entirely without reference to external information, and semi-supervised document classification, where parts of the documents are labeled by the external mechanism[4].

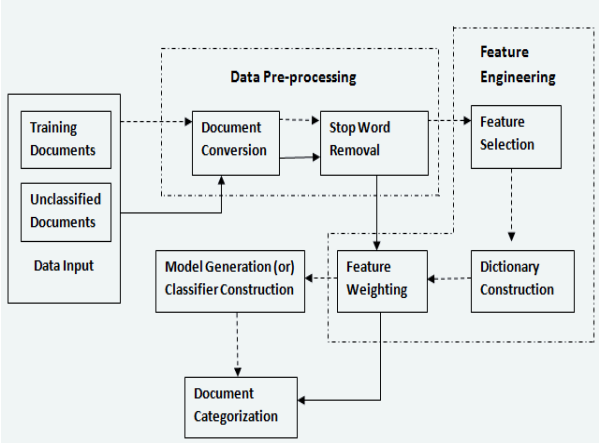

Figure 2.1: Architecture of Document Classification

In the above figure the arrow with dashed line represents the data flow in the training process and the arrow with solid line represents the data flow in the classification process[1].

Document categorization goes through a number of stages; figure 2.1 shows a general overview for the framework of a typical categorization system. The document classification is divided into two processes training and classification[2]. The training process comprises three key components: data pre-processing, feature engineering and model generation, where as first two phases are common for both training and classification processes and the classification process makes use of the model generated by the training process for new document classification[7].

2.1.1 Data Pre-processing

Text pre-processing meansto transform documents into a suitable representation for the classification task. The text documents have different stop words, punctuation marks, special character and digits and other characters[5]. In the preprocessing stage we should remove HTML Tags, Stop words from the text documents. After removing stop words, word stemming is performed .Word stemming is the process of suffix removal to general word stems. A stem is a natural group of words with similar meaning. In text pre-processing the tasks should be performed are:

- Removal of HTML tags and special character
- Conversation removal
- Tag most common text in a document
- Stop words removal
- White spaces removal
- *FeatureEngineering*
- Feature engineering will remove all the features that are not relevant and common across all documents[8]. It includes different techniques for selecting the features that are most relevant and yields more information needed for the classification task. Different phases in feature engineering include:
- Feature Selection
- Dictionary Construction
- Feature Weighting

III:MODEL GENERATION (OR) CLASSIFIER CONSTRUCTION

- Classifier construction is the key component of automatic text categorization. The role of this component is to build a k-Nearest Neighbor classifier or to generate a kNN model by learning from predefined documents, which will be used to classify unknown documents[4].A k-nearest neighbor classifier is constructed using a feature extractor, the number of neighbors (k) to consider and the Euclidean distance as a similarity measure. The training process will be completed with this model generation phase where the training process simply involves storing the feature vector for each training instance along with its category[1].

3.1Document Categorization

The document categorization component directly uses the model created in the classifier construction phase to classify new instances. All documents to be classified must be preprocessed as in the classifier construction phase.

2.2 *Distance Function*

Data analysts define distance metrics to measure similarity. A distance metric or distance function is a real-valued function d, such that for any coordinates x, y, and z:

1.d(x,y) ≥ 0,and d(x,y) = 0
if and only if x = y
2. d(x,y) = d(y,x)
3. d(x,z) ≤ d(x,y) + d(y,z)

Property 1assures us that distance is always nonnegative, and the only way for distance to be zero is for the coordinates (e.g., in the scatter plot) to be the same.

Property 2 indicates commutativity, so that, for example, the distance from New York to Los Angeles is the same as the distance from Los Angeles to New York. Finally.[5]

Property 3 is the triangle inequality, which states that introducing a third point can never shorten the distance between two other points[3].

The most common distance function is Euclidean distance, which represents the usual manner in which humans think of distance in the real world:

$$d_{Euclidean}(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

where $x = x_1, x_2, \ldots, x_m$, and $y = y_1, y_2, \ldots, y_m$ represent the m attribute values of two records.

IV:K- NEAREST NEIGHBOR ALGORITHM

One of the simplest classification methods used in data mining and machine learning is the k-Nearest Neighbor (k-NN). It is the most accepted classification method due to its ease and practical efficiency: it doesn't necessitate fitting a model and it has been proved to have superior performance for classifying several types of data[1]. But, the superior classification performance of k-NN is greatly dependent on the metric used for computing pair-wise distances between data points. Practically, to calculate k nearest neighbor data points of interest, Euclidean distances are often used as similarity metric. One often needs to find out or select a good distance metric to categorize high-dimensional data in real applications[2].

The classification rules of k-NN are created by the training samples alone, with no other additional data. In a more complicated approach, k-NN classification, finds a group of k objects in the training set that are nearest to the test object, and bases the assignment of a label on the predominance of a particular class in this neighborhood.

The k-Nearest Neighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification.

Algorithm

BEGIN
Input: D = {(x_1,c_1), . . . , (x_N, c_N)}
x = (x_1, . . . ,x_n) new instance to be classified
FOR each labeled instance (x_i, c_i) calculate distance(x_i, x)
Order d(xi, x) from lowest to highest, (i = 1, . . . ,N)
Select the K nearest instances to x: D_x^K
Assign to x the most frequent class in D_x^K
END

V: EXPERIMENTAL RESULTS

To demonstrate the proof of concept we built a custom simulator in C++ programming language. The prototype is built in an environment consisting of a PC with 4GB RAM and Core 2 Dual processor running in Windows 7. Visual Studio 2013 IDE is used for rapid prototype development. The experimental results are recorded and compared with the existing approach.

RESULTS

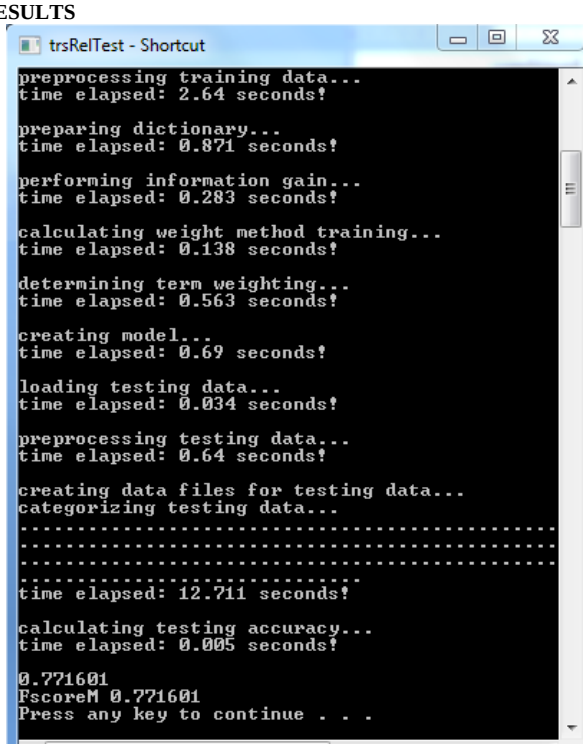

Figure 4.1: The Classification Demo of k-NN Classifier

This screen shows that the training data is pre-processed first and then it will prepare a dictionary of words, after that it will perform information gain to select the features which are most relevant to the current classification task and calculate tf*idf weights for each feature in a dictionary and then it will create a k-NN model which can be further used by the classifier to classify new incoming documents. Finally we can measure the accuracy of a k-NN classifier.

The performance of k-Nearest Neighbor classifier is measured in terms of Fscore- the harmonic mean of precision and recall. The file'_reliability_test_table' shows the performance results of the k-Nearest Neighbor classifier. It shows how many documents classified correctly and how many documents misclassified, and the overall accuracy of the kNN classifier.

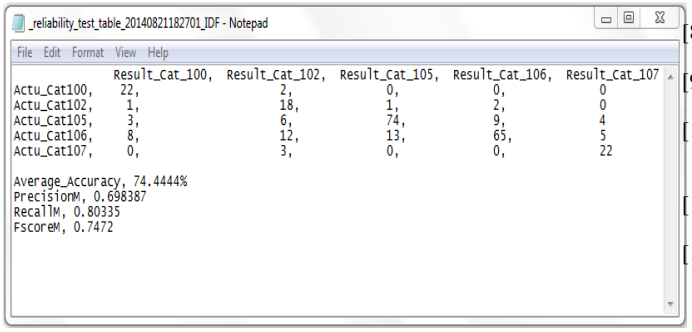

Figure 4.2: The results of k-NN classifier reliability test

VI: CONCLUSIONS AND FUTURE WORK

The k-Nearest Neighbor classifier is responsible for classifying incoming emails, where each email is considered as a single document[1]. The documents in the training set were preprocessed, after that they were represented using the vector space model based on the set of features extracted from each. For measuring similarity between two documents, the Euclidean distance measure was used. Finally for measuring the effectiveness of classification the traditional recall and precision measures was used[4].

The selection of the feature space, the training data set used, and the value of k can enormously affect the accuracy of classification[2].The k-Nearest Neighbor algorithm can be easily modifiable and it allows improving code further to increase efficiency and accuracy[1]. The following are the enhancements to make the kNN classifier more efficient.The traditional kNN classifier uses the vector space model for representing the features[2]. The use of kd-tree data structure for representing the features in a multi dimensional array will improve the efficiency of nearest neighbour search.

ACKNOWLEDGEMENT

I am thankful to my guide, Dr.K. Subrahmanyam, KL University, for his help and guidance. Also thanks to VNRVJIET management And CSE Dept. for providing me with the necessary resources as and when the situation demanded and for reposing trust in our abilities.

REFERENCES

[1] Shengyi Jiang, Guansong Pang, Meiling Wu, LiminKuang, An improved K-nearest-neighbor algorithm for text categorization.
[2] Dr. Riyad Al-Shalabi, Dr. GhassanKanaan, Manaf H. Gharaibeh, Arabic Text Categorization Using kNN Algorithm.
[3] H. Bashiri, FarhadOroumchian, A. Moeini,Persian Email Classification Based on Rocchio and K-Nearest Neighbor Approach.
[4] Xiaogang Han, Junfa Liu, ZhiqiShen, and Chunyan Miao,An Optimized K-Nearest Neighbor Algorithm for Large Scale Hierarchical Text Classi_cation
[5] Nicolette Nicolosi, Feature Selection Methods for Text Classification.
[6] ZHANG Yun-tao, GONG Ling, WANG Yong-cheng, An improved TF-IDF approach for text classification.
[7] Oliver Sutton, Introduction to k NearestNeighbour Classification and Condensed Nearest Neighbour Data Reduction.
[8] Juan Ramos, Using TF-IDF to Determine Word Relevance in Document Queries.
[9] GulenToker, OznurKırmemiş, Text categorization using k-nearest neighbor classification.
[10] Eui-Hong (Sam) Han George Karypis, Vipin Kumar, Text Categorization Using Weight Adjusted *k*-Nearest Neighbor Classification.
[11] XiaogangPeng& Ben Choi,Document Classifications Based On WordSemantic Hierarchies.
[12] GongdeGuo, Hui Wang, David Bell, Yaxin Bi and Kieran Greer, Using *K*nnModel- based Approach for Automatic Text Categorization.