INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY
VOLUME 4 ISSUE 2 – APRIL 2015 - ISSN: 2349 - 9303

Text Categorization Using Improved K Nearest Neighbor Algorithm

Femi Joseph
PG scholar
MEA Engineering College
Perinthalmanna, India
femy.j46@gmail.com

Nithin Ramakrishnan
Assistant Professor
MEA Engineering College
Perinthalmanna, India
nithinramakrishnan@gmail.com

Abstract— Text categorization is the process of identifying and assigning predefined class to which a document belongs. A wide variety of algorithms are currently available to perform the text categorization. Among them, K-Nearest Neighbor text classifier is the most commonly used one. It is used to test the degree of similarity between documents and k training data, thereby determining the category of test documents. In this paper, an improved K-Nearest Neighbor algorithm for text categorization is proposed. In this method, the text is categorized into different classes based on K-Nearest Neighbor algorithm and constrained one-pass clustering, which provides an effective strategy for categorizing the text. This improves the efficiency of K-Nearest Neighbor algorithm by generating the classification model. The text classification using K-Nearest Neighbor algorithm has a wide variety of text mining applications.

Index Terms— Text Categorization, K Nearest Neighbor classifier, One-pass clustering, document class, training data, test data, classification model

1 Introduction

With the sudden growth of available information on the internet, it is easy to extract information than before. Collecting useful and favorable information from huge repositories is a complex and time consuming task. Here Information Retrieval (IR) plays an important role. Information Retrieval is a process that collects relevant information from a large collection of resources with the help of some keywords. But the amount of information obtained through Information Retrieval is larger than that a user can handle and manage. In such a case, it requires the user to analyze the obtained results one by one until the satisfied information is attained, which is time-consuming and inefficient. Therefore, certain tools are required to identify desired documents. Text categorization is one such possible implementation.

Text categorization can be done automatically or manually. In most cases, the text categorization is done automatically. Automatic text categorization is the process of automatically assigning natural language texts to predefined categories based on the content. Without engendering training documents by hand, it automatically generates training sentence sets utilizing keyword lists of each category. This in turn reduces the available time for extracting satisfiable information. A large number of potential applications make it more popular. For Text Categorization a large number of machine learning, knowledge engineering, and probabilistic-based methods have been proposed. The most popular methods include Bayesian probabilistic methods, regression models, example-based classification, decision trees, decision rules, K Nearest Neighbor, Neural Networks (NNet), Support Vector Machines (SVM), centroid-based approaches and association rules mining. Among all these algorithms, K-Nearest Neighbor is widely used as text classifier because of its simplicity and efficiency.

K Nearest Neighbor classification finds a group of k objects from the training set that are closer to the test object, and predicates the assignment of a label on the predominance of a particular class in its neighborhood. There are three key elements for this approach: a set of labeled objects, a distance or similarity metric to compute the distance between given objects, and the value of k, the number of nearest neighbors. In order to classify an unlabeled object, the distance from the unlabeled object to the other labeled objects is calculated, and its nearest neighbors are identified. The class labels of these nearest neighbors are then used to determine the class label of the object.

The main contributions of this paper are as follows: 1) Proposes a new text categorization technique based on an existing KNN text classifier algorithm and a constrained one pass clustering. The resulting text categorization system equally performs or outperforms one of the best performers in this task (i.e., KNN). 2) It is suited in many applications well, where data is updated dynamically and required rigorously in real-time performance. This method will incrementally or dynamically updates the classification model. 3) The method significantly outperforms KNN and more effective.

3 PROBLEM STATEMENT

K Nearest Neighbor (KNN) is an efficient algorithm for text categorization. But the main problem is that it does not generate the classification model. Instead, it uses all the training data to categorize the input test. By incorporating a constrained one-pass clustering, the KNN algorithm can overcome this defect. In this improved method, the constrained one-pass clustering is used to generate the classification model.

The classification model will allow the KNN algorithm to work efficiently by reducing the time taken by the KNN algorithm to categorize the input text. In addition, this improved model will incrementally update the classification model dynamically.

2 RELATED WORK

Pascal Soucy and Guy W. Mineau[1] propose a simple KNN algorithm for text categorization. This method does an aggressive feature selection. Feature Selection is used for selecting a subset of all available features. Here, the feature selection method allows the removal of features, which adds no new information and features with weak prediction capability. When such features interact with other features, it may lead to redundancy. Redundancy and irrelevancy could mislead KNN learning algorithm by some unwanted preconception, and increases its complexity. By considering both the redundancy and the relevancy of the features, the simple KNN algorithm provides an efficient method. The KNN algorithm provides a solid ground for text categorization in large document sets and diverse vocabulary. In this method, each text document is called as an instance. In order to categorize texts using KNN, each example document X is represented as a vector of length |F|, the size of the vocabulary. The algorithm uses the simple feature weighting approach. In this approach, distance is treated as a basis to weight the contribution of each k neighbor in the class assignment process. With text documents, text categorization may involve thousands of features, most of them being irrelevant. It is very difficult to identify the relevant feature for text categorization because the interaction between the features is much closer. The feature selection method used here to aggressively reduce the vocabulary size using feature interaction. This simple KNN algorithm can reach impressive results using a few features.

Songbo Tan [2] proposes an effective refinement strategy for KNN text classifier, where DragPushing is proposed as a refinement strategy to enhance the performance of KNN. Since KNN uses all training documents to predict classes of test documents, all training documents within one class can be taken as class-representative of that category. It will make use of training errors to refine the KNN classifier by ‘dragging’ and ‘pushing’ operation, as called ‘DragPushing Strategy’. If one training example is misclassified into a wrong class, then the ‘DragPushing’ technique ‘drag’ the class-representative in the correct class to the example, and ‘push’ the class-representative in the misplaced class against the example. Clearly, after the DragPushing, the correct class tends to put more examples in K-nearest neighbor set and these nearest neighbors share larger similarities with test document, and vice versa. The reason why DragPushing Strategy using K Nearest Neighbor (DPSKNN) can overcome the problem of imbalance of text corpus is that, if it takes class A as minority category and class B as majority category, then according to traditional KNN decision rule, the examples in category A tends to be classified into class B. As a result, the weight vector of class A has many `Drag' operation than that of class B. After a few rounds of `DragPushing' operations, the minority category A tends to have much larger weight vector than majority category B. Consequently, the different weight vector associated with each category could counteract the impact of unbalance of training corpus to a high degree. During each iteration, all training documents need to be classified. If one document labeled as class A is classified into class B, DragPushing is utilized to adjust it. Conspicuously after the `Drag' and `Push' operation, all or most of elements in the weight vector of class A will be increased while all or most of elements in the weight vector of class B will be decreased. As a result, the similarities of all or most documents in the class A to document will be increased and in the class B to document will be decreased. The weights, i.e., drag weight and push weight, are used to control the step-size of `drag' and `push'. The major advantage of this technique is that, it does not need to generate sophisticated models, but only requires simple statistical data and the traditional KNN classifier. The DPSKNN could make a significant difference in the performance of the KNN Classifier and distributed better performance than other commonly used methods.

Songbo Tan [3] proposes a neighbor weighted K-nearest neighbor for unbalanced text corpus. To unbalanced text corpora, the majority class tends to have more examples in the K-neighbor set for each test document. If we utilize the traditional KNN technique to classify the test document, the document tends to assign the label of the majority class to the document. Hence, the big category tends to have higher classification accuracy, while the other that is the minority class tends to have low classification accuracy. So that, the total performance of KNN will be degraded. In this method, instead of balancing the training data, the algorithm assigns a big weight for neighbors from small class, and assigns a little weight for the neighbors that are in large category. For each test document d, first select K neighbors among training documents contained in K* categories. For test document d, it should be assigned the class that has the highest resulting weighted sum, as in the case of traditional KNN. The NWKNN yields much better performance than KNN. The algorithm NWKNN is an effective algorithm for unbalanced text classification problems.

Zhou Yong, Li Youwen and Xia Shixiong [4] propose an improved KNN text classification algorithm based on clustering, which reduces the high computational complexity. This method uses a weight value for each new training sample, which indicates the degree of the importance when classifying the documents. This algorithm improves both the efficiency and the accuracy of text categorization. In order to classify the document, it uses the preprocessing technique. The pre-processing of text classification includes many key technologies, such as sentence segmentation, remove stop words, feature extraction and weight calculation. Sentence segmentation is the process of dividing the given text document into meaningful units such as words, sentences, etc. The words that do not have any importance in the content of document and nearly have no effect on classification are called stop words, so the removal of the stop words is necessary. Feature extraction means the process of reducing the amount of resources required to describe a large set of data. This method does not use all training samples as a traditional KNN algorithm, and it can overcome the problem of uneven distribution of training samples. This improving algorithm uses the k-means clustering algorithm to get the cluster centers, which were used as the new training samples. Then a weight value is introduced for each cluster center that indicates the importance of them. At last, the revised samples with weight are used in the algorithm.

2.1 Observations and analysis

The most efficient and simple algorithm that is used to categorize the text is the K Nearest Neighbor algorithm. Different methods, based on the KNN text classification algorithm, are used to classify the text data. The result of analysis of different text categorization methods is shown in the TABLE 1 below.

TABLE 1: COMPARISON OF VARIOUS METHODS USING KNN ALGORITHM

Technique Used	**Method**	**Advantages**	**Disadvantages**
Simple KNN Algorithm	Feature selection method	Reduces the vocabulary size	Less efficient
Effective refinement strategy for KNN	DragPushing strategy	No need to generate sophisticated models	Cannot predict the number of iterations
Neighbor Weighted KNN	Neighbor Weighted KNN	Better performance for unbalanced text	Difficult to calculate the weight sum
KNN based on clustering	k-means clustering	Overcome defect of uneven distributions of training samples	Difficult to select the threshold parameter

4 PROPOSED WORK

This paper is organized as follows: In Section 4.1, the system architecture is illustrated. In Section 4.2, a brief description about each module is provided. In Section 4.3, the data set description and the evaluation of the proposed work and the base work is given.

4.1 System Architecture

The figure.1 provides an overview of the system architecture.

A set of collected documents are used as the training set. Using the collected documents, the training sentence set is created. This training set is used to generate the classification model. The generation the classification model is performed using the clustering process. During this process, a set of document clusters are generated. After the generation of document clusters, the input text is compared with the document clusters and it is assigned to the corresponding class. The input text is first represented in its text representation format. Then this represented input is given to the text classifier. The text classifier makes use of the document clusters to categorize the text into its corresponding category.

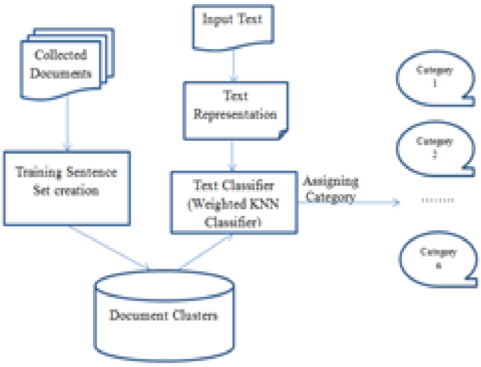


Figure 1: System Architecture

4.2 Module Description

In the proposed work, the text categorization technique is divided into three modules. Each of these modules will work together to obtain a better result for the text categorization. First one is the classification model generation module which make use of constrained one-pass clustering. Next is the text categorization module which is implemented using Weighted KNN algorithm. Then finally the updating classification model module which updates the classification model dynamically.

Classification Model Generation Module: In this module, the classification model is built by making use of clustering. Clustering is used to generate different clusters of the training document. One-pass clustering algorithm is a kind of incremental clustering algorithm with approximately linear time complexity. To build the classification model with the training text documents, one pass clustering algorithm is used to constrainedly cluster the text collections. The number of the clusters obtained with constrained clustering is much less than the number of training samples. As a result, when KNN method is used to classify the test documents, the text similarity computation is substantially reduced and the impact of its performance affected by single training sample is also reduced.

Text Categorization Module: Text categorization is done using Weight Adjusted KNN text classifier. It will classify the test input into the predefined category, which is done in the previous module. This categorization is done by calculating the weight of the test document which is learned using an iterative algorithm. It is then used for computing the similarity measurement of the test document with each training class.

Updating classification model Module: In this module the classification model is updated incrementally. i.e., whenever a new training data arrives, the classifier will update the existing classification model. Many previous text categorization algorithms are hard to or cannot update their classification model dynamically. However, the size of the text data are huge and increases constantly in the real-world applications, and rebuilding model is very time-consuming. INNTC is an incrementally modeling algorithm, which can update its classification model quickly based on the new training samples. It is valuable in practical applications.

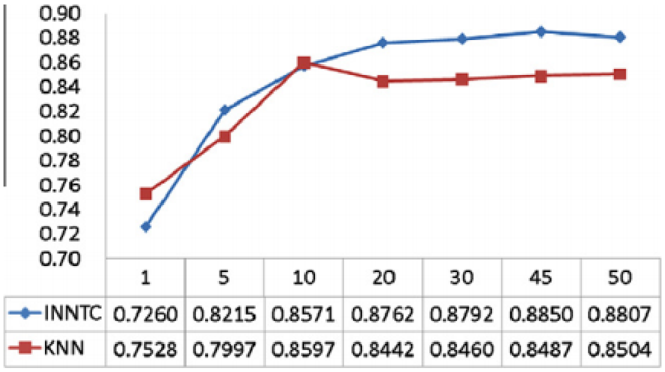
4.3 Data Set and Evaluation

Reuters-21578 is a standard benchmark for text categorization. The Reuters-21578 collection contains 21578 documents and 135 categories appeared on the Reuters news wire in 1987. The dataset used in this project is the Reuters Transcribed Subset. It consists of data that was created by selecting 20 files each from the 10 largest classes in the Reuters-21578 collection. The data format consists of 10 directories labeled by the topic name. Each contains 20 files of transcriptions which belong to the area, business.

5 RESULTS

MATLAB has been implemented and three programs have been built around it. The first of these programs is capable of generating the document classes from the existing dataset. For this document class generation, first calculate a threshold value by selecting a random number of documents. The second program is for generating the Weighted KNN algorithm. The third program is used to update the document class dynamically whenever a new training data arises.

The text categorization has been evaluated with different k values and it is compared with the existing KNN algorithm.



	1	5	10	20	30	45	50
INNTC	0.7260	0.8215	0.8571	0.8762	0.8792	0.8850	0.8807
KNN	0.7528	0.7997	0.8597	0.8442	0.8460	0.8487	0.8504

Figure 2: The classification results with different k values in Reuters-21578

6 Conclusion and Future Work

This paper describes a text categorization technique using K-Nearest Neighbor algorithm. The text categorization technique based on improved K Nearest Neighbor Algorithm is the most suitable one due to its minimal time and computational requirements. In this text categorization technique, clustering is a great tool to discover the complex distribution of the training texts. It uses constrained one pass clustering algorithm to obtain the categories relationship by the constrained condition (each cluster only contains one label). This can better reflect the complex distributions of the training texts than original text samples. As integrating the advantages of the constrained one pass clustering and Weight Adjusted KNN approach, this method has significant performance comparable with other text classifiers.

Acknowledgment

An exertion over a long period may be successful only with advice and guidance of many well wishers. We take this opportunity to express our gratitude to all who encouraged us to complete this work. We would like to express our deep sense of gratitude to our respected Principal for his inspiration and for creating an atmosphere in the college to do the work. We would also like to thank Head of the department, Computer Science and Engineering.

References

[1] Pascal Soucy, Guy W Mineau. “A Simple KNN Algorithm for Text Categorization.” Proceedings of International Conference on Date Mining. 2001: 647-648.

[2] Tan, Songbo. "An effective refinement strategy for KNN text classifier." Expert Systems with Applications 30.2 (2006): 290-298.

[3] Tan, Songbo. “Neighbor-weighted k-nearest neighbor for unbalanced text corpus” Expert Systems with Applications 28.4 (2005): 667-671.

[4] Zhou, Yong, Youwen Li, and Shixiong Xia. "An improved KNN text classification algorithm based on clustering." Journal of computers 4.3 (2009): 230-237.

[5] Jiang, Shengyi, et al. "An improved K-nearest-neighbor algorithm for text categorization." Expert Systems with Applications 39.1 (2012): 1503-1509.

[6] L. R. Bahl, S. Balakrishnan-Aiyer, J. Bellegarda, M. Franz, P. Gopalakrishnan, D. Nahamoo, M. Novak, M. Padmanabhan, M. Picheny, and S. Roukos, “Performance of the IBM large vocabulary continuous speech recognition system on the ARPA wall street journal task.” In Proc. of ICASSP ’95, pages 41–44, Detroit, MI, 1995.

[7] S. Agarwal, S. Godbole, D. Punjani and S. Roy, “How Much Noise is too Much: A Study in Automatic Text Classification”, In Proc. of ICDM 2007.

[8] D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. http://www.research.att.com/ lewis, 1999.