RESEARCH ARTICLE OPEN ACCESS

Genetic Algorithm Based Text Categorization Using K-NN Method

Mr. R. AlaghuKarthiKeyan [1], Mrs. k .Brindha [2]
Department Of Computer Science
Research Scholar [1], Assistant Professor [2]
Sri Jayendra Saraswathy College of Arts & Science
Coimbatore
Tamil Nadu – India

ABSTRACT

Text categorization is to classify the documents into fixed number of predefined categories.each document on one in multiple exactly one or no category at all.K nearest neighbour determines the decision boundary locally.for KNN we assign each document to the majority class of its closest neighbour where K is parameter.the rational of KNN classification is based on contiguity s,we expect the test document to have the same training label as the training documents located in the local region surrounding the document .the genetic algorithm is used to modify the weight of the feature term until feature vector trained for every category can represent the feature of this category.

Keywords:- Text classification system, feature term, feature selection algorithm, classification model, genetic algorithm, KNN algorithm

I. INTRODUCTION

Text categorization (or classication) is the tak of assigning natural language texts to one or more thematic categories on the basis of their contents..the documents according to the user queries, which are based on the key terms. The key trems all belong to finite set called controlled vocabulary.The task of assigning keyboards from a controlled vocabulary to text documents is called text indexing.

Text categorization can be done automatically or manually.In most of the cass, th e text categorization is done automatically.an Automatic text categorization is the proces of automatically assigning natural language text to predefined categories based ion the content.It reduces thethe available time for extracting satisfiable information.text categorization is the large number of machine learning and makes more popularrule.

The idea in K-Nearest Neighbor methods is to identify K obesrvation in thr training datset that are similar to anew record that we wish to classify.the use of these (neighboring) records to clssifying the new record into class , assigning the new record to the prdominent class amonmg these neighbors.The central issues here is how to measure the distance between rcords based on their predictor values .After computing the distance between the record to be classified and existing recoirds , we need a rule to assign a class to the record to be classified , based on the classes of its neighbors.this intuitive idea of using a single nearest neighbor to classify records can powerful when we have a large number of records in the trsining set.The main contribution of this paper are as follows :1)proposes a new text categorization technique based on an existing KNN text classifier method. In this where text is updated dynamically were K|NN is more effective.

II. K-NEAREST NEIGHBOR TEXT CATEGORIZATION METHOD

Text categorization is the process of grouping documents into one or more predefined categories based on the types of the content which we goning to select. .A number of statistical classification and machine learning techniques have been applied to text categorization , including regression models, bayesian classifiers, decision trees, nearest neighbors .The first step In text categorization is to transform documents which is typically sre strings of characters, into a representation suitable for leaning algorithma and classification algorithm.

A word-by-document matrix **A**is used for a collection of documents, where each entry represents the occurrence of a word in a document, i.e., $\mathbf{A} = (a_{ij})$, where a_{ij} is the weight of word *i* in document *j*. There are several ways of determining the weight a_{ij} . Let f_{ij} be the frequency of word *i* in document *j*, *N* the number of documents in the collection, *M* the number of distinct words in the collection, and n_i the total number of times word *i* occurs in the whole collection. The simplest approach is Boolean weighting, which sets the weight a_{ij} to 1 if the word occurs in the document and 0 otherwise. Another simple approach uses the frequency of the word in the document, i.e., $a_{ij} = f_{ij}$. A more common weighting approach is the so-called *tf* ･ *idf* (term frequency - inverse document frequency) weighting:

$$a_{ij} = f_{ij} \times log\left(\frac{N}{n_i}\right). \qquad (1)$$

A slight variation [17] of the *tf* ･ *idf* weighting, which takes into account that documents may be of different lengths, is the following:

$$a_{ij} = \frac{f_{ij}}{\sqrt{\sum_{l=1}^{M} f_{lj}^2}} \times log\left(\frac{N}{n_i}\right). \qquad (2)$$

For matrix **A**, the number of rows corresponds to the number of words *M* in the document collection. There could be hundreds of thousands of different words. In order to reduce the high dimensionality, stop-word (frequent word that carries no information) removal, word stemming (suffix removal) and additional dimensionality reduction techniques, feature selection or re-parameterization [9], are usually employed.

To classify a class-unknown document ***X***, the *k*-Nearest Neighbor classifier algorithm ranks the document's neighbors among the training document vectors, and uses the class labels of the *k* most similar neighbors to predict the class of the new document. The classes of these neighbors are weighted using the similarity of each neighbor to ***X***, where similarity is measured by Euclidean distance or the cosine value between two document vectors. The cosine similarity is defined as follows:

$$sim(X, D_j) = \frac{\sum_{t_i \in (X \cap D_j)} x_i \times d_{ij}}{\|X\|_2 \times \|D_j\|_2} \qquad (3)$$

where *X* is the test document, represented as a vector; D_j is the *j*th training document; t_i is a word shared by *X* and D_j; x_i is the weight of word t_i in *X*; d_{ij} is the weight of word t_i in document D_j; $\|X\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2 + ...}$ is the norm of *X*, and $\|D_j\|_2$ is the norm of D_j. A cutoff threshold is needed to assign the new document to a known class.[4]

Table 1: Analogy between text categorization and intrusion detection when applying the *k*NN classifier.

Terms	Text categorization	Intrusion Detection
N	total number of documents	total number of processes
M	total number of distinct words	total number of distinct system calls
n_i	number of times *i*th word occurs	number of times *i*th system call was issued
f_{ij}	frequency of *i*th word in document *j*	frequency of *i*th system call in process *j*
D_j	*j*th training document	*j*th training process
X	test document	test process

The *k*NN classifier is based on the assumption that the classification of an instance is most similar to the classification of other instances that are nearby in the vector space. Compared to other text categorization methods such as Bayesian classifier, *k*NN does not rely on prior probabilities, and it is computationally efficient. The main computation is the sorting of training documents in order to find the *k* nearest neighbors for the test document.[4]

We seek to draw an analogy between a text document and the sequence of all system calls issued by a process, i.e., program execution. The occurrences of system calls can be used to characterize program behavior and transform each process into a vector. Furthermore, it is assumed that processes belonging to the same class will cluster together in the vector space. Then it is straightforward to adapt text categorization techniques to modeling program behavior. Table 1 illustrates the similarity in some respects between text categorization and intrusion detection when applying the *k*NN classifier[4].

III. GENETIC ALGORITHM

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. GAs are one of the best ways to solve a problem for which little is known. They are a very general algorithm and so will work well in any search space [1].

Genetic algorithm was in fact invented by nature. Charles Darwin named it 'evolution' and "survival of fittest" theory used to evolve genetic algorithm. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. Genetic algorithm, which was first presented and named by J.H. Holland in the 1875, imitates the natural algorithm of the evolution entirely [1]. GA works in an iterative manner by generating new populations of strings from old ones. Every string is the encoded binary, real etc which has known as chromosome. An evaluation function associates a fitness measure to every string indicating its fitness for the problem.

The basic terms in genetic algorithm are described below. □

Individual - Any possible solution

Population - Group of all individuals □

search Space - All possible solutions to the problem

Chromosome - Blueprint for an individual □

Allele - Possible settings for a trait □

Locus - The position of a gene on the chromosome

Genome - Collection of all chromosomes for an individual

There are many different sources from which information can be extracted. Genetic algorithm is used in Xml mining, opinion mining, web mining, knowledge discovery, feature extraction, classification and different text mining techniques to optimize the solution using the mechanisms of genetic evolution and survival of the fittest in natural selection. The major advantage of using GA in text mining is that they perform global search and its time complexity is less compared to other algorithms as the genetic algorithm is based on the greedy approach [2].

In [3] Genetic algorithm (GA) is applied on large XML data sets to discover the frequent item sets. First the samples of records are loaded from the transaction database that fits into memory. An initial population is created consisting of randomly generated transactions. Each transaction can be represented by a string of bits. Then by applying genetic operators, correct and appropriate results are gathered.

Figure 1 shows working model divided into two general levels of processing. The input is a corpus of technical and scientific natural language documents; the output is a small set of the hypotheses that the GA discovered [3]. Automatic text summarization takes an input text and extracts the most important content in the text

In [5] two different approaches have been used in the text summarization domain. The first one is using genetic algorithms to learn the patterns in the documents that lead to the summaries. The other one is using lexical chains as a representation of the lexical cohesion that exists throughout the text. The experiments performed on the CAST corpus showed that combining different classes of features and the results showed that features like sentence location, sentence centrality and named entities give better performance than the other features. However, the combination of the features yields better success rates than any individual feature.

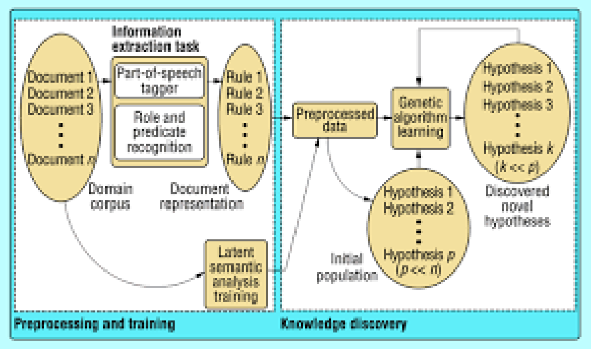

Fig.1. Genetic algorithm based knowledge discovery from text

IV.CONCLUSION AND FUTRUE WORK

In this paper genetic algorithm using the K-NN method is presented in detail and it has been discussed how it is beneficail to work with these methodologies..Electronic based records are broadly accessible becouse of the development of the extraction from the immense collection of text based data using K-NN method provide a thorough understanding of different text mining using genetic algorithm.this leads to help the to proceed and develop new mining techniques in the broad areas for the effective result.

REFERENCE

[1] S.N.Sivanandam, S.N.Deepa, "Introduction to genetic algorithms" Springer-Verlag Berlin Heidelberg 2008

[2] M. Srinivas, Lalit M. Patnaik, "Genetic Algorithms: A survey" Motorola Indian Electronics Ltd., Indian Institute of Science, IEEE.

[3] Soumadip Ghosh, Amitava Nag, Debasish Biswas, Arindrajit Pal, Sushanta Biswas, Debasree Sarkar, Partha Pratim Sarkar, " Xml mining using genetic algorithm" Journal of Global Research in Computer Science, Volume 2, No. 5, April 2011.

[4] https://www.usenix.org/legacy/event/sec02/full_papers/liao/liao_html/node4.html

[5] Mine Berker, Tunga Güngör, "Using genetic algorithms with lexical chains for automatic text summarization" Boğaziçi University, Computer Engineering Dept., Bebek 34342, Istanbul, Turke

[6] Deepankar Bharadwaj, Suneet Shukla, "Text mining technique using genetic algorithm" International Conference on Advances in Computer Application (ICACA - 2013) Proceedings published in International Journal of Computer Applications® (IJCA) (0975 –8887)

[7] Kietz J.U. and zeroski, S. D (1994) 'Inductive Logic Programming and Learnability' SIGART Bull., vol. 5, no. 1, pp. 22-32.

[8] Kloesgen W. (1996) 'Explora: A Multipattern and Multistrategy Discovery Assistant' Advances in Knowledge Discovery and Data Mining, pp. 249-271.

[9] Lewis, D.D. (1997) "Reuters-21578 Text Categorization Test Collection," Distribution 1.0, http://metaxa.net/,.

[10] Lewis, D.D. and Hayes, P.J. (1994) 'Guest Editors' Introduction to the Special Issue on Text Categorization' ACM Trans. Information Systems, vol. 12, no. 3, p. 231

[11] Agresti A. (2002) 'Categorical Data Analysis', WileyInterscience.

[12] Anthony M. and Biggs N. (1992) 'Computational Learning Theory' Cambridge Univ. Press.

[13] Antonie M. and Zaiane O. (2004) 'An Associative Classifier Based on Positive and Negative Rules,' Proc. Ninth ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD),.

[14] [Apte C. Damerau F.J. and Weiss S.M. (1994) 'Automated Learning of Decision Rules for Text Categorization' ACM Trans. Information Systems, Vol. 12, no. 3, pp. 233-251.

[15] Baralis E. and Garza P. (2006) 'Associative Text Categorization Exploiting Negated Words' Proc. 21st Ann. ACM Symp. Applied Computing (SAC '06), pp. 530-535.