Machine Learning approach for Content Mining System

Dr.A.Mekala, Asst.Professor,
Dept.of Computer Applications (UG)
Sacred Heart College, Tirupattur
and
. Dr.A.Prakash, Professor,
PG & Research Department of Computer Science
Hindusthan Arts and Science College, Coimbatore

Abstract

Text Classification (TC), also known as Text Categorization, is the mission of robotically classifying a set of text documents into dissimilar categories from a predefined set. If a manuscript belongs to exactly one of the categories, it is a single-label categorization task; otherwise, it is a multi-label categorization task. TC uses several tools from Information Retrieval (IR) and Machine Learning (ML) and has received much consideration in the last years from both researchers in the academia and manufacturing developers. In this paper, we first categorize the documents using KNN based machine learning approach and then return the most appropriate documents.

Keywords: *Text Mining, Naïve Bayes, KNN, Event models, Document Mining, Term-Graph, Machine Learning.*

1. Introduction

Information Retrieval (IR) is the information of searching for in order within relational databases, documents, text, multimedia files, and the World Wide Web. The applications of IR are miscellaneous; they comprise but not limited to extraction of information from large documents, searching in digital libraries, information filtering, spam filtering, item extraction from images, mechanical summarization, manuscript classification and clustering, and web searching. The get through of the Internet and web look for engines have urged scientists and great firms to generate very great scale recovery systems to remain pace with the exponential development of online data. Figure below depicts the structural design of all-purpose IR system. The client first submits a query which is executed over the retrieval system. The final, consults a database of manuscript compilation and proceeds the identical document. In general, in order to learn a classifier that is able to correctly classify hidden documents, it is essential to teach it with some pre-classified documents from each group, in such a way that the classifier is then able to simplify the representation it has learned from the pre-classified credentials and use that model to correctly classify the unnoticed documents. Figure 1 shows the summary of the document indexing and retrieval system. From experimentation, KNN shows the maximum accuracy as compared to the Naive Bayes and Term-Graph. The disadvantage for KNN is that its time complexity is high but gives a better accuracy than others.

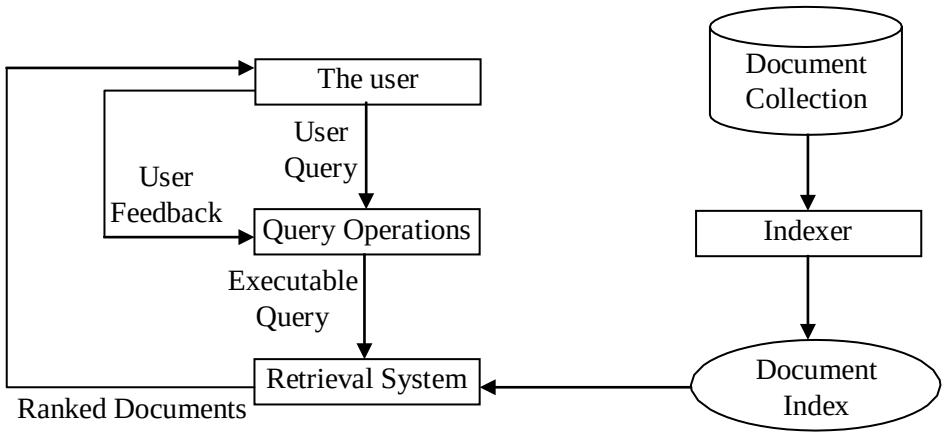


Figure 1: Overview of document retrieval system

In modern years, passage categorization has become an significant research topic in machine learning and in sequence retrieval and e-mail spam filtering. It also has become an important research topic in text mining, which analyses and extracts helpful information from texts. The survey is oriented towards the various probabilistic approach of KNN Machine Learning algorithm for which the text categorization aims to classify the document with best accuracy. Information recovery is also used in image retrieval. In recent works, to save and estimation accurate location moving object with energy constraint is proposed in using adaptive update algorithms. Some other recent approaches such as video summarization, 3D model of 2D image, gait pattern, and level imitation model can also be integrated with the proposed move toward to improve the efficiency.

This document categorizes the news articles into various categories. We work on two main scenarios:

a. Organization of documents into various categories.

Making it in the outline of a function where user can upload an article and we will classify it into different categories.

b. On entering keywords by the user we demonstrate the most appropriate document for the user.

2. Categorization Methods

This manuscript concerns methods for the organization of natural language text, that is, methods that, given a set of training documents with known categories and a new document, which is usually called the query, will predict the query's group.

Algorithm:

1) Checking the keyword in Test manuscript and storing it in a map.
2) Calculating yes and no occurrence of each keyword in the test document.
3) Calculating the probability of each keyword of the test document.

4) Classifying the Test manuscript into various categories on the basis of probability calculated.

Word Graph Model

The word graph model is an improved version of the vector space model [6] by weighting each term according to its comparative "importance" with regard to term associations. Specifically, for a text essay D_i, it is represented as a vector of term weights $D_i = <w_{1i}, w_{|T|i}>$, where T is the ordered set of terms that occur at least once in at least one document in the gathering. each weight w_{ji} represents how a great deal the corresponding term t_j contribute to the semantics of document d_i. Even though a number of weighting schemes have been proposed (e.g., Boolean weighting, frequency weighting, tf-idf weighting, etc.), those schemes settle on the weight of each term individually. As a result, important yet rich in sequence regarding the relationships among the terms are not captured in those weighting schemes.

We bring in to establish the weight of each term in a document collection by constructing a term graph. The basic steps are as follows:

1. **Preprocessing Step**:

For a compilation of document, remove all the terms.

In our term graph model, we will arrest the relationships surrounded by terms using the frequent item set mining method. To do so, we think each text document in the training collections as a transaction in which each word is an item. However, not all words in the manuscript are important adequate to be retained in the transaction. To decrease the processing space as well as augment the precision of our model, the text documents need to be preprocessed by (1) remove stop words, i.e., words that emerge frequently in the manuscript but have no important meanings; and (2) retaining only the root form of words by stemming their affixes as well as prefixes.

2. Graph Building Step:

(a) For each article, we view it as a transaction: the document ID is the corresponding transaction ID; the terms contained in the document are the items contained in the corresponding transaction. Association rule mining algorithms can thus be applied to mine the frequently co-occurring terms that occur more than minus times in the collection.

(b) The common co-occurring terms are mapped to a weighted and directed graph, i.e., the term graph.

As mentioned above, we will capture the relationships amongst terms using the regular item set mining method. While this idea has been explored by earlier research, our come up to distinguishes from previous approaches in that we maintain all such significant associations in a graph. The graph not only reveals the important semantics of the document, but also provides a foundation to extract novel features about the document, as we will show in the next section. Following the preprocessing step, each document in the text collection will be stored as a transaction (list of items) in which each item (term) is represented by a unique non-negative integer. Then ordinary item set mining algorithms can be used to find all the subset of items that appeared extra than a threshold amount of times in the collected works.

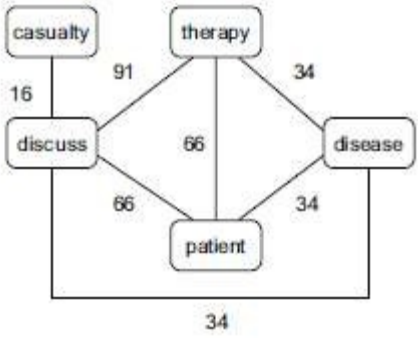
In our system, our aim is to discover the associations among the important terms of the text in a category and try to put together out a strategy to construct use of these relationships in the classifier and other manuscript removal tasks. Vector space representation cannot express such rich relationship in the midst of terms. Diagram is thus the most suitable data structure

in our circumstance, as, in general, each term may be associated with more than one terms. We propose to use the following simple method to construct the chart from the set of everyday item sets mined from the textbook collections. First, we build a node for each exclusive term that appears at least once in the common item sets. Then we generate edges between two node u and v if and only if they are both contained in one frequent item set. additionally, we assign weights to the edges in the following way: the influence of the edge stuck between u and v is the largest carry value among all the frequent item sets that contains both of them.

For, example, regard as the everyday item sets and their complete support shown in Figure 2(a). Its corresponding graph is shown in Figure 2(b).

Itemset	Support
$\{therapy, discuss\}$	91
$\{therapy, discuss, patient\}$	66
$\{therapy, discuss, patient, disease\}$	34
$\{casualty, discuss\}$	16

(a) Frequent Itemsets



(b) The Corresponding Graph

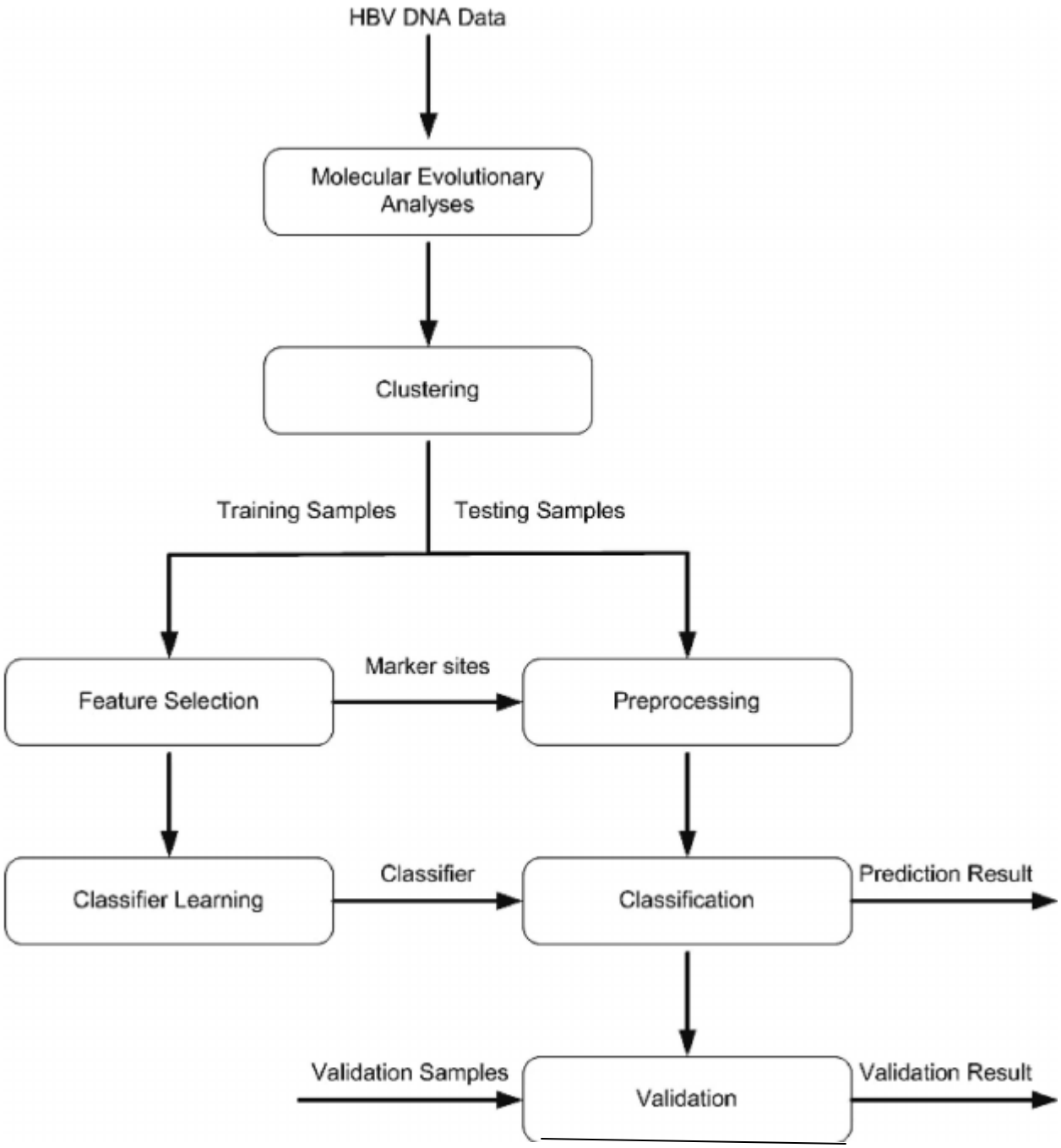
Figure 2: (a) Frequent item set with support, and (b) Corresponding graph.

Algorithm:

1) Setting every unique word occurring the text as nodes of the graph.
2) Making Adjacency Matrix of the keywords.
3) Making space Matrix using Dijkstra.
4) Calculating similarity stuck between the test document keywords and the keywords of every category.

k-Nearest Neighbors

The preliminary submission of k-Nearest Neighbors (KNN) to text categorization was reported in The basic idea is to settle on the category of a given query based not only on the document that is nearest to it in the manuscript space, but on the categories of the k documents that are nearest to it. Having this in mind, the Vector method can be viewed as an instance on the KNN method, where k=1. This effort uses a vector-based, distance-weighted matching utility, as did Yang, by calculating document's similarity similar to the Vector method.



3. Flowchart of text and content mining

Then, it uses a voting approach to find the query class: each retrieved document contributes a vote for its class, weighted by its similarity to the uncertainty. The query's probable classifications will be ranked according to the votes they got in the earlier step.

Algorithm:

1) Construct vector for each document in the test set.
2) Formulate centriod vector for each class.
3) Calculate similarity between each document vector and class vector.
4) Manuscript belongs to the class for which the likeness is maximum.

Figure 3 shows the flowchart of the proposed system for text and document mining using machine knowledge techniques.

3. Experimental Results

Dataset

The data set used for this paper is in the appearance of SGML files we have used Reuters-21578 dataset which is accessible at [1]. There are 21578 documents; according to the „ModApte" split: 9603 training docs, 3299 test docs and 8676 unused docs. They were labeled manually by Reuter's workers. Labels belong to 5 different category classes, such as „people", „places", „Exchange", Organization" and topics". The whole number of categories is 556, but many of them occur only incredibly rarely. The dataset is divided in 88 files of 1000 documents delimited by SGML tags.

Implementation

For classifying the documents in we initially pre-processed the data by performing various techniques:

a. Case Folding
b. Normalization
c. Bag of words
d. Stop word removal
e. TF-IDF

Then after pre-processing, we applied KNN, Term Graph algorithm, and Naïve Bayes algorithms to classify the documents in the training set into five categories. We additional applied our classifier model on the test documents and calculated the accuracy by comparing it with the default answers given for the investigation documents. To compare the above mentioned algorithms, we used the following metric:

Exactness, which is defined as the proportion of correctly classified documents, is generally used to evaluate single-label TC tasks.

$$Accuracy = \frac{\#\text{Correctly classified documents}}{\#\text{Total documents}}$$

We then created an request where user can input some keywords and based on the algorithm showing higher accuracy we show the applicable document to the user.

RESULTS

We compared the accuracy of Naïve Bayes, Term Graph and KNN for Text and Document categorization of our articles of Reuter 21578. we found that KNN shows the best result with accuracy as provided in Table 1.

Table 1: Accuracy for each method

Category/Method	NAÏVE	Term Graph	KNN
Places	71.68	94.41	98.00
Organization	51.23	98.11	98.11
People	43.19	98.21	99.82
Topics	85.20	99.11	98.29
Exchange	82.13	99.22	99.34

Commencing above results, we can articulate that KNN based learning method is added suitable than Naïve Bayes and Term Graph categorization method for the mining of text or documents. The accuracy reported for KNN is a good deal high than Naïve based method as shown in Table 1 for each category of the dataset.

Conclusion

We bring to a close that KNN shows the utmost accurateness as compared to the Naive Bayes and Term-Graph. The negative aspect for KNN is that its point in time complexity is high but gives a enhanced accuracy than others. We implemented Term-Graph with other methods to a certain extent than the long-established Term-Graph used with AFOPT. This hybrid shows a enhanced result than the traditional combination. Finally we made an information repossession application using Vector Space Model to give the result of the query entered by the client by showing the appropriate document. We will focus more in future on Reducing Complexity, Increasing exactness and manuscript Summarization.

REFERENCES

1. Lin, C.Y., Hovy, E.: Automated Text Summarization in SUMMARIST. In: Proc. of ACL Workshop on Intelligent, Scalable Text Summarization, Madrid, Spain (1997)
2. Song, Y., et al.: A Term Weighting Method based on Lexical Chain for Automatic Sum- marization. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 636–639. Springer, Heidelberg (2004)
3. HaCohen-Kerner, Y., Zuriel, G., Asaf, M.: Automatic Extraction and Learning of Key- phrases from Scientific Articles. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 657–669. Springer, Heidelberg (2005)
4. Villatoro-Tello, E., Villaseñor-Pineda, L., Montes-y-Gómez, M.: Using Word Sequences for Text Summarization. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 293–300. Springer, Heidelberg (2006)
5. Chuang, T.W., Yang, J.: Text Summarization by Sentence Segment Extraction Using Ma- chine Learning Algorithms. In: Proc. of the ACL 2004 Workshop, Barcelona, España (2004)
6. Neto, L., Freitas, A.A., Kaestner, C.A.A.: Automatic Text Summarization using a Machine learning Approach. In: Proceedings of the ACL 2004 Workshop, Barcelona, España (2004)
7. Ledeneva, Y., Gelbukh, A., García, H.R.: Terms Derived from Frequent Sequences for Ex- tractive Text Summarization. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 593–604. Springer, Heidelberg (2008)
8. Ledeneva, Y., Gelbukh, A., García, H.R.: Keeping Maximal Frequent Sequences Facili- tates Extractive Summarization. Research in Computing Science 34 (2008)
9. Cristea, D., Postolache, O., Pistol, I.: Summarization through Discourse Structure. In: Gel- bukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 632–644. Springer, Heidelberg (2005)
10. Kupiec, J., Pedersen, J.O., Chen, F.: A Trainable Document Summarizer. In: Proc. 18th ACM-SIGIR Conf. on Research and Development in Information Retrieval, pp. 68–73 (1995)
11. DUC. Document Understanding Conference 2002 (2002), http://www-nlpir.nist.gov/projects/duc
12. Xu, W., Li, W., Wu, M., Li, W., Yuan, C.: Deriving Event Relevance from the Ontology
Constructed with Formal Concept Analysis. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 480–489. Springer, Heidelberg (2006)
13. Mihalcea, R.: Random Walks on Text Structures. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 249–262. Springer, Heidelberg (2006)
14. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proc. Empirical Meth- ods in Natural Language Processing (EMNLP 2004), Barcelona, Spain (2004)
15. Hassan, S., Mihalcea, R., Banea, C.: Random-Walk Term Weighting for Improved Text Classification. In: Proc. Semantic Computing (ICSC 2007), Irvine, CA (2007)