EasyChair Preprint

№ 2429

Efficient Ranked Multi-Keyword Search using Machine Learning Algorithms

Namdeo Kedare and Chaitanya Mankar

January 20, 2020

Efficient Ranked Multi-Keyword Search using Machine Learning Algorithms

Namdeo Kedare
Dept of Computer Engineering
Dhole Patil College of Engineering
Pune, India
ns.soft21@gmail.com

Chaitanya Mankar
Dept of Computer Engineering
Dhole Patil College of Engineering
Pune, India
chaitanyamankar@gmail.com

Abstract — **The growing amount of documents in the search index of information retrieval systems make the problem of ranking documents crucial. The modern state of the problem leads to the point where machine learning becomes the most efficient way to optimize the ranking function. Keyword search is an effective method to retrieve information from such useful networks. The aim of keyword search is to find a set of answers covering all or part of the queried keywords. A challenge in keyword search systems is to rank answers according to their relevance to the query. This relevance lies in the textual content and structural compactness of the answers. Classification is the process of classifying the text documents based on words, phrases and word combinations with respect to set of predefined categories. Data classification has many applications such as mail routing, email filtering, content classification, news monitoring and narrow-casting. Keywords are extracted from documents to classify the documents. Keywords are subset of words that contains the most important information about the content of the document. Keyword extraction is a process used to take out the important keywords from documents. In this proposed system keywords are extracted from documents using TF-IDF and naïve bays algorithm. TF-IDF algorithm is used to select the candidate words. The words which have highest similarity are taken as keywords. The experiment has been done using Naive Bayes algorithms and its performance is analyzed based on machine learning.**

Keywords: - Keyword based search, machine learning, naïve bayes algorithm, TF-IDF algorithm, Ranking, classification.

I. INTRODUCTION

Over the last decade, the number of digital documents available for various purposes has grown enormously with the increasing availability of high capacity storage hardware and powerful computing platforms. The vivid increase of documents demands effectual organizing and retrieval methods mainly for large documents. Text classification is one of the key techniques in text mining to categorize the documents in a supervised manner. The processing of text classification involves two main problems are the extraction of feature terms that become effective keywords in the training phase and then the actual classification of the document using these feature terms in the test phase. Text classification can be used for document filtering and routing to topic specific processing mechanisms such as information extraction and machine translation. Various methods are used for document classification such as Naive Bayes, Support Vector Machine, K-Nearest Neighbor, Fuzzy C-means, Neural Networks, Decision trees and Rule based learning algorithms out sourcing.

II. LITERATURE SURVEY

A. Ghanbarpour, H. Naderi [1] In this paper, an attribute-specific ranking method is proposed based on language models to rank candidate answers according to their semantic information up to the attribute level. This method scores answers using a model enriched with attribute-specific preferences and integrating both the structure and content of answers. The proposed model is directly estimated on the sub-graphs (answers) and is defined such that it can preserve the local importance of keywords in nodes.

Karl Severin, Swapna S. Gokhale Aldo Dagnino. [2] In this scheme supporting efficient ranked keyword search for achieving effective utilization of remotely stored encrypted data. Inside this structure, we use a feasible once-over to in addition enhance the intrigue suitability, and get the ostensibly debilitated constrain framework to cover get the chance to instance of the demand client. Security examination shows that our course of action can accomplish gathering of records and report, trapdoor confirmation, trapdoor unlinkability, and hiding access instance of the intrigue client.

Vidhya.K.A, G.Aghila [3] showed a safe multi-catchphrase arranged look design over encoded cloud information, which meanwhile underpins dynamic fortify operations like destruction and development of reports. Naive Bayes works well for the data characteristics with certain deterministic or almost deterministic dependencies that is low entropy distribution, however the fact is that algorithm work well even when the independence assumption is violated.

Pawar Supriya, Dr. S. A. Ubale [4] proposed a gainful multi-catchphrase break even with word ask for over blended cloud information by recovering best k scored records. The vector space model and TFIDF demonstrate are utilized to gather record and question time. The KNN calculation used to scramble record and demand vectors and develop a unique tree called Balanced M-way Search (BMS) Tree for asking for and propose a Depth First Search Technique (DFST) figuring to complete reasonable multi-catchphrase proportionate word arranged search for. The effectiveness and precision of DFST estimation are addressed with a case, BMS tree, it takes sub-straight time multifaceted nature.

Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, Christopher Ré [5] It has proposed a paradigm for the program aticcreation of training sets called data programming in which users express weak supervision strategies ,which are programs that label subsets of the data, but that are noisy and may conflict, which gives high quality with the increasing availability of high capacity storage hardware and powerful computing platforms. The vivid increase of documents demands effectual organizing and retrieval methods mainly for large documents. Text classification is one of the key techniques in text mining to categorize the documents in a supervised manner. The processing of text classification involves two main problems are the extraction of feature terms that become effective keywords in the training phase and then the actual classification of the document using these feature terms in the test phase. Text classification can be used for document filtering and routing to topic specific

processing mechanisms such as information extraction and machine translation. Various methods are used for document classification such as Naive Bayes, Support Vector Machine, K-Nearest Neighbor, Fuzzy C-means, Neural Networks, Decision trees and Rule based learning algorithms out sourcing.

III. PROPOSED METHODOLOGY

The proposed architecture of four modules: user interface, log pre-processing, Feature Clustering using Naïve Bayes Classification, Training and testing using support vector machine for more accurate categorization of opinion. This system can solve irrelevant data and more accuracy by associating Modified K means with Naïve Bayes Classification algorithm.

Naive Bayes (NB):

Naive Bayes Classifier uses Bayes Theorem, which finds the probability of an event given the probability of another event that has already occurred. Naive Bayes classifier performs extremely well for problems which are linearly separable and even for problems which are non-linearly separable it performs reasonably well.

TF-IDF Algorithm:

TF_IDF stands for Term frequency-inverse document frequency. The TF-IDF weight is often used in information retrieval and text mining. Variations of the TF-IDFweighting scheme are often used by search engines in scoring and ranking a document's relevance given a query. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.

A. System Architecture

Basically Proposed System is consists of 3 modules

User

This module helps clients to enter their query keyword to get the most important documents from set of uploaded documents. This module recovers the documents from cloud which coordinates the query keyword.

Data Owner

After expansion of keywords the data owner assist data with multiple keywords the document utilizing based on machine learning Algorithm and after that upload the document to store the database.

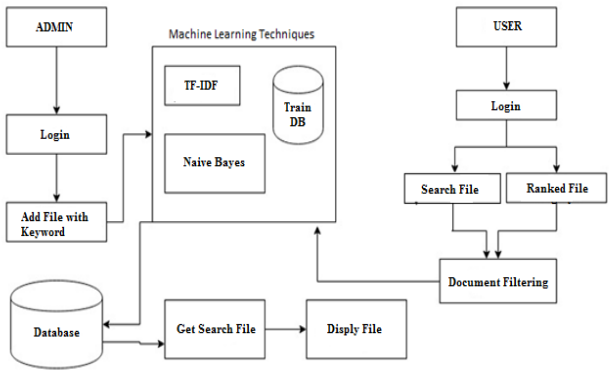

Fig. Architecture of System

Ranked Results

Clients/user can download the resultant arrangement of documents just if he/she is approved client who has allowed consent from data owner to download specific document. Here user get the ranked and mostly search records from the ranking system to get exactly data to the all user.

B. Algorithms

1. Naive Bayes Algorithm

Naive Bayes classifier calculates the probability of an event in the following steps:

Step 1: Calculate the prior probability for given class labels.

Step 2: Find Likelihood probability with each attribute for each class.

Step 3: Put these value in Bayes Formula and calculate posterior probability.

Step 4: See which class has a higher probability, given the input belongs to the higher probability class.

Among the performed survey aims in getting an intuitive understanding of Naïve Bayes approach in which the application of various Machine Learning Techniques to the text categorization problem like in the field of e-mail filtering, medicine, including rule learning for knowledge base systems has been explored. The survey is oriented towards the various probabilistic approach of Naïve Bayes Machine Learning algorithm for which the text categorization aims to classify the document with maximum accuracy.

Naïve Bayes Model works with the conditional probability which originates from well known statistical approach "Bayes Theorem", where as Naïve refers to "assumption" that all the attributes of the examples are independent of each other given the context of the category. Because of the independence assumption the parameters for each attribute can be learned separately and this greatly simplifies learning especially when the number of attributes is large. In this context of text classification, the probability that a document d belongs to class c is calculated by the Bayes theorem as follows

$$P(c/d) = \frac{p(d/c)/p(c)}{p(d)}$$

Bayes' Theorem provides a way that we can calculate the probability of a piece of data belonging to a given class, given our prior knowledge.

Bayes' Theorem is stated as:

P(class|data) = (P(data|class) * P(class)) / P(data)

Where P(class|data) is the probability of class given the provided data.

2. TF-IDF

Term frequency-inverse document frequency is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.

It has many uses, most importantly in automated text analysis, and is very useful for scoring words in machine learning algorithms for Natural Language Processing (NLP).

TF-IDF was invented for document search and information retrieval. It works by increasing proportionally to the number of times a word appears in a document, but is offset by the number of documents that contain the word. So, words that are common in every document, such as this, what, and if, rank low even though they may appear many times, since they don't mean much to that document in particular.

The term frequency of a word in a document. There are several ways of calculating this frequency, with the simplest being a raw count of instances a word appears in a document. Then, there are ways to adjust the frequency, by length of a document, or by the

raw frequency of the most frequent word in a document.
The inverse document frequency of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.
So, if the word is very common and appears in many documents, this number will approach 0. Otherwise, it will approach 1.
Multiplying these two numbers results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document.
To put it in more formal mathematical terms, the TF-IDF score for the word t in the document d from the document set D is calculated as follows:

tf idf (t, d, D) = tf (t, d) . idf (t, D)

Where:
T f (t, d) = log (1 + freq (t, d))
idf (t, D) = log (N/((count(d € D: t € d))

Machine learning with natural language is faced with one major hurdle – its algorithms usually deal with numbers, and natural language is, well, text. So we need to transform that text into numbers, otherwise known as text vectorization. It's a fundamental step in the process of machine learning for analyzing text, and different vectorization algorithms will drastically affect end results, so you need to choose one that will deliver the results you're hoping for. Once you've transformed words into numbers, in a way that's machine learning algorithms can understand, the TF-IDF score can be fed to algorithms such as Naive Bayes and Support Vector Machines, greatly improving the results of more basic methods like word counts. Simply put, a word vector represents a document as a list of numbers, with one for each possible word of the corpus. Vectorizing a document is taking the text and creating one of these vectors, and the numbers of the vectors somehow represent the content of the text. TF-IDF enables us to gives us a way to associate each word in a document with a number that represents how relevant each word is in that document. Then, documents with similar, relevant words will have similar vectors, which is what we are looking for in a machine learning algorithm.
It's important to use TF-IDF for text extraction so that you can gain a better understanding of how machine learning algorithms function. While machine learning algorithms traditionally work better with numbers, TF-IDF algorithms help them decipher words by allocating them a numerical value or vector. This has been revolutionary for machine learning, especially in fields related to NLP such as text analysis.
In text analysis with machine learning, TF-IDF algorithms help sort data into categories, as well as extract keywords. This means that simple, monotonous tasks, like tagging support tickets or rows of feedback and inputting data can be done in seconds.
Every wondered how Google can serve up information related to your search in mere seconds? Vectorization transforms text within documents into numbers, so TF-IDF algorithms can rank articles in order of relevance.

IV. RESULT AND DISCUSSIONS

Naïve Bayes generally outperforms for large datasets in text classification problem in spite of the Naïve independence assumption but as of small data sets Naïve Bayes doesn‟t show promising results in accuracy or performance.[6] Even though Naïve Bayes technique achieves better accuracy, to fine tune the classification accuracy it‟s combined with the other machine learning technique like SVM, neural networks , decision trees which has been discussed above.
Basically Naïve Bayes work with the conditional probability derived from the idea of Bayes theorem which is modified according to the application of Naïve Bayes for text classification. To evaluate the text classifier system with the Naïve Bayes approach there are two metrics factor, precision, recall and F1-measure can be used to find the effectiveness of document classifier which is given by,

tp (True Positive): The number of documents correctly classified to that class. tn (True Negative): The number of documents correctly rejected from that class. fp (False Positive): The number of documents incorrectly rejected from that class.
fn (False Negative): The number of documents incorrectly classified to that class.

P: Precision= tp/(tp+fp)
R: Recall= tp/(tp+fn)
F1(Measure) = 2(P*R)/(P+R)

The formulas for precision, recall and F-measure is given in The performance of Naïve Bayes Machine learning technique when combined with the other method shows better performance.
The discussion about the Naïve Bayes performance with the micro F1-measure values for the multinomial methods available from paper [6] such that the variants of the classifiers significantly outperform the traditional multinomial Naive Bayes at least when the 20-Newsgroup is used. In the graph showing the microF1 values, SRF _ l at _of 0.2 achieves the best performance. RF _ u and SRF _ u also achieve better performance [6] than baseline performance and less so than the Rf _ l or SRF _ l, but trivial. It means that there is no significant difference between using the number of tokens and the number of unique terms.
The biggest difference between the microF1 and macroF1 is that the performance increase by the normalization over the baseline is much greater in the case of macroF1 (0.2238 for the baseline versus 0.5066 for RF-l). Since macroF1 values in the Reuters21578 collection tend to be dominated by a large number of small categories, which have a small number of training documents [6], From the above survey of this paper it is understood that the proposed normalization methods are quite effective, particularly in the categories where the number of positive training documents is small where the traditional Naïve Bayes Technique fails, the author have done subsequent experiments and found the method is quite effective.
For Text categorization there are various benchmark datasets available like Reuters21578, Cora, WebKB and 20Newsgroup. Reuters21578 and 20Newsgroup datasets are designed with either set of long or short document. There are predefined categories where the hierarchy structure for each category is specified. The dataset WebKB is generally preferred for spam mail classification simulation. However the results of the Naïve Bayes along with the other hybrid methods for text document classification with these datasets and feature selection technique is depicted in the following Table1. Performance of Naïve Bayes when combined with other methods.

Table I: Text Document Classification and Naïve Bayes Machine Learning Approach

Text Document Classification and Naïve Bayes Machine Learning Approach			
Naïve Bayes Model (Method)	Feature Selection Techniques	Data Sets Used	Accuracy/Performance
Naïve Bayes Model with Noun Phrase approach	User defined Feature selection template	The training material comes from four sections (sections 15-18) of the Wall Street Journal (WSJ) part of the Penn Treebank-II corpus, including 400 English texts composed of 211727 words.	93.7%
Naïve Bayes with Probability estimation Tree	Small Size Data –No Feature Selection Required	Experiments on 9 UCI Datasets are conducted	On Average 87%
Naï ve Bayes with Support Vector Machine	TF-IDF (Term Frequency and Inverse Term Frequency Method)	20Newgroup and Prepared own Dataset for testing	Flat Ranking – 88.89% Flat Ranking with High Ranking Keyword – 90.00%
Naïve Bayes with Active Learning Boosting Method	Weightage Scheme	6 Datasets from the UCI Machine Learning Repository	Achieved Higher Accuracy by 0.05% compared to Adaboost
Naïve Bayes with Generative/Discriminative Technique	Wavelet transformation Feature subset of Documents	Reuters21578, Cora, WebKB and 20Newsgroup Dataset	92.5% on Average
Naïve Bayes for Learning Object Identification	Weightage Scheme, Normalized Statistics	Set of own data files	Good Learning Object Identification is achieved.
Naïve Bayes for E-mail Spam Filtering	Mutual Information Gain	Lingspam corpus and PUI corpus	Multivariate – 98.86% Accuracy ,Multinomial - 98.06%
Naïve Bayes with Multivariate and Multinomial Distribution	New Feature Weightage scheme was proposed and tested	Reuters21578 and 20Newsgroup	F1-Measure is compared for various weightage scheme. - 0.5066 Multinomial- 0.2238

TF-IDF

TF*IDF is an information retrieval technique that weighs a term's frequency (TF) and its inverse document frequency (IDF). Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF*IDF weight of that term. This term weighting suggests that the valuable terms will has a high frequency in particular documents, but low frequency on the outside. Two main components of the term weight must be distinguished: term frequency factor and inverse document frequency. The term frequency tf is the number of times a term appears in a document.

$TF(ij) = F(ij) / L(i)$

Where F(ij) is the frequency of term j in document i, and L(i) is total number of keywords in the document i.

There are various weighting schemes to discriminate one document from others. This factor is called inverse document frequency IDF.

$IDF(i) = \log(n/n)+1 \qquad n(j) > 0$

Where n is the number of documents and n(j) is the number of documents in which term j occurs. The concepts of term frequency and inverse document frequency are combined, to produce a composite weight for each term in each document.

TF – IDF =TF*IDF

V. CONCLUSION

This system is designed keyword with top-k ranked search over secure server data. The system provides the accurate result ranking documents & search efficiency due to the use of tree based index and efficient search algorithm. For future work there are many challenges in symmetric searchable encryption scheme. As it is assumed that all the data users are trustworthy, but in practical, the dishonest data user may distribute his secure keys to unauthorized users.

REFERENCES

[1] A. Ghanbarpour, H. Naderi, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2018. "An Attribute-Specific Ranking Method Based on Language Models for Keyword Search over Graphs"

[2] Karl Severin, Swapna S. Gokhale Aldo Dagnino. 2019 IEEE43rd Annual Computer Software and Applications Conference (COMPSAC), pp: 978-1-7281-2607-4. "Keyword-Based Semi-Supervised Text Classification"

[3] Vidhya.K.A, G.Aghila (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 2, 2010. "A Survey of Naïve Bayes Machine Learning approach in Text Document Classification"

[4] Pawar Supriya, Dr. S. A. Ubale.International Journal for Research in Applied Science & Engineering Technology (IJRASET) *Volume 5 Issue VII, July 2017. "*Multi-Keyword Top-K Ranked Search over Encrypted Cloud Using Parallel Processor"

[5] Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, Christopher RéStanford University, pages 3567–3575, 2016. "Data Programming:Creating Large Training Sets, Quickly"

[6] Sang-Bum kim, Kyong-soo Han, Hae-Chang Rim, Sung Hyon Myaeng "Some Effective techniques for Naïve Bayes Text Classification" IEEE Transactions on Knowledge and Data Engineering -2006.

[7] Siham JABRI, Azzeddine DAHBI, Taoufiq GADI, Abdelhak BASSIR, "Ranking of Text Documents using TF-IDF Weighting and Association Rules mining"

[8] Xinggao Cai, Shujin Cao, "A keyword extraction method based on learning to rank"

[9] S.V. Semenikhin, L.A. Denisova "Learning To Rank Based on Modified Genetic Algorithm"

[10] Tian Weixin, Zhu Fuxi "Learning to Rank Using Semantic Features in Document Retrieval"