Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 4, April 2016, pg.128 – 135

Optimization of Personalized Information Retrieval System Using Map Reduce and Vector Space Model

Shivangi Goel[1], Abhishek Sharma[2]

[1]Computer science and Engineering & SRM University, India
[2]Computer science and Engineering & SRM University, India
[1] goel.9.shivangi@gmail.com; [2] abhishek.s@ncr.srmuniv.ac.in

***Abstract*— This Personalised Information Retrieval systems are of great need now a day. With increasing data the need of its retrieval also arrived. Personalised information retrieval systems help user to find relevant information according to their interest with the advantage of no more time consuming. Such systems help to provide personalised services to the user and reduces the overloading of information. In this paper we are using an optimization technique for personalised information retrieval systems using modified k-means for clustering the user interests then Mapreduce for removing duplication of data and vector space model for classification and prediction.**

***Keywords*— Information retrieval, Modified K-means, Mapreduce, Vector space Model, Personalized Information Retrieval**

I. INTRODUCTION

A. *Personalized Information Retrieval systems:*

With the popularity of the World Wide Web, people have no other option to find proper information about anything. However they have to waste a lot of time in searching the relevant amount of information from a thousand of results get displayed. The most commonly used engines, such as Google, Baidu and Yahoo, are general search engines, by which users of diverse backgrounds and with various needs get the similar search results when they use the same keywords. What's more, a large number of these results are of no use of user. Therefore, personalization of search technology has become a major issue to be solved. Current general search engines cannot serve user's intelligentized and personalised needs according to their interest. The main reason for this is their failure to comprehend semantic information of page layout and analyze user's individual preference. Now there are a great many personalized information retrieval service systems, which have brought forward many ideas to realize personalized service. To achieve the aim of personalised retrieval by the means of the import of user's preference and interests. In order to realize the personalized information retrieval service on user interests, first of all, we need to track and store user's interests as well as to classify user interests into groups based on their access pattern, then design an appropriate way of expression, and finally recommend information considering user interests.

B. Hadoop MapReduce

1) Hadoop has:

- HDFS - A distributed file system
- A MapReduce framework- that allows algorithms to work on data in the distributed file system in parallel

The Hadoop Distributed File System (HDFS) is the heart of Hadoop. It provides scalability, reliability and high performance at a very low cost. The system is designed to run on commodity hardware. Although the system is written in Java, there are other ways to access and use it.

HDFS-The distributed file system is designed to handle large files (multi-GB) with sequential read/write operation. Each file is broken into chunks, and stored across multiple data nodes as local OS files. HDFS is a Java-based file system that provides scalable and reliable data storage, and it was designed to span large clusters of commodity servers. HDFS has demonstrated production scalability of up to 200 PB of storage and a single cluster of 4500 servers, supporting close to a billion files and blocks. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.

Hardware failure is the norm rather than the exception. An HDFS instance may consist of hundreds or thousands of server machines, each storing part of the file system's data. The fact that there are a huge number of components and that each component has a non-trivial probability of failure means that some component of HDFS is always non-functional. Therefore, detection of faults and quick, automatic recovery from them is a core architectural goal of HDFS. Large Data Sets-Applications that run on HDFS have large data sets. A typical file in HDFS is gigabytes to terabytes in size. Thus, HDFS is tuned to support large files. It should provide high aggregate data bandwidth and scale to hundreds of nodes in a single cluster. It should support tens of millions of files in a single instance. The primary objective of HDFS is to store data reliably even in the presence of failures.

MapReduce:

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters of commodity hardware in a reliable, fault-tolerant manner. A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks. MapReduce can take advantage of the locality of data, processing it near the place it is stored in order to reduce the distance over which it must be transmitted. The overall concept is simple, but is actually quite expressive when you consider that:

- Almost all data can be mapped into <key, value> pairs somehow, and
- Your keys and values may be of any type: strings, integers, dummy types… and, of course, <key,value> pairs themselves.

MapReduce is a functional programming paradigm that is well suited to handling parallel processing of huge data sets distributed across a large number of computers. MapReduce, as its name implies, works in two steps:

A MapReduce program is composed of a **Map()**procedure (method) that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) A **Reduce()** method that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies).

The main thing with a Map Reduce algorithm is that it reasons about <key, value > pairs all along, from the input format to the output format, if necessary using synthetic keys. If your input is a simple flat file, it will by default break it up on line ends and provide the offset into the file as key and the line as value. The main strength of the algorithm lies with the fact that between the map and the reduce phase, it will sort the data by key. The framework will then provide all data with the same key to the same reducer instance. Any successful MapReduce algorithm should leverage this mechanism.

Mapper - Breaks up the input text in tokens (filtering some common punctuation marks) and applies the character sorting to arrive at the required key.

Combiner (optional) - Removes duplicate values from the input.

Reducer - Collects anagrams and outputs the number of anagrams (key) and all the words concatenated (value).

Main and Run - This code configures the job to run on the MapReduce framework. The Combiner is used to do some preprocessing for the reducer.

C. VECTOR-SPACE MODELS:

The vector-space models for information retrieval are just one subclass of retrieval techniques that have been studied in recent years. The taxonomy provided in labels the class of techniques that resemble vector-space models ``formal, feature-based, individual, partial match'' retrieval techniques since they typically rely on an underlying, formal mathematical model for retrieval, model the documents as sets of terms that can be individually weighted and manipulated, perform queries by comparing the representation of the query to the representation of each document in the space, and can retrieve documents that don't necessarily contain one of the search terms. Although the vector-space techniques share common characteristics with other techniques in the information retrieval hierarchy, they all share a core set of similarities that justify their own class.

Vector-space models rely on the premise that the meaning of a document can be derived from the document's constituent terms. They represent documents as vectors of terms $d = (t_1; t_2; \ldots; t_n)$ where t_i $(1 \leq i \leq n)$ is a non-negative value denoting the single or multiple occurrences of term i in document d. Thus, each unique term in the document collection corresponds to a dimension in the space. Similarly, a query is represented as a vector $q = (\hat{t}_1; \hat{t}_2; \ldots; \hat{t}_m)$ where term $\hat{t}_i$ $(1 \leq i \leq m)$ is a non-negative value denoting the number of occurrences of $\hat{t}_i$ or $\hat{t}_i$ merely a 1 to signify the occurrence of term in the query. Both the document vectors and the query vector provide the locations of the objects in the term-document space. By computing the distance between the query and other objects in the space, objects with similar semantic content to the query presumably will be retrieved.

Vector-space models that don't attempt to collapse the dimensions of the space treat each term independently, essentially mimicking an inverted index. However, vector-space models are more flexible than inverted indices since each term can be individually weighted, allowing that term to become more or less important within a document or the entire document collection as a whole. Also, by applying different similarity measures to compare queries to terms and documents, properties of the document collection can be emphasized or deemphasized. For example, the dot product (or, inner product) similarity measure finds the Euclidean distance between the query and a term or document in the space. The cosine similarity measure, on the other hand, by computing the angle between the query and a term or document rather than the distance, deemphasizes the lengths of the vectors. In some cases, the directions of the vectors are a more reliable indication of the semantic similarities of the objects than the distance between the objects in the term-document space.

Vector-space models were developed to eliminate many of the problems associated with exact, lexical matching techniques. In particular, since words often have multiple meanings (polysemy), it is difficult for a lexical matching technique to differentiate between two documents that share a given word, but use it differently, without understanding the context in which the word was used. Also, since there are many ways to describe a given concept (synonomy), related documents may not use the same terminology to describe their shared concepts. A query using the terminology of one document will not retrieve the other related documents. In the worst case, a query using terminology different than that used by related documents in the collection may not retrieve any documents using lexical matching, even though the collection contains related documents.

Vector-space models, by placing terms, documents, and queries in a term-document space and computing similarities between the queries and the terms or documents, allow the results of a query to be ranked according to the similarity measure used. Unlike lexical matching techniques that provide no ranking or a very crude ranking scheme (for example, ranking one document before another document because it contains more occurrences of the search terms), the vector-space models, by basing their rankings on the Euclidean distance or the angle measure between the query and terms or documents in the space, are able to automatically guide the user to documents that might be more conceptually similar and of greater use than other documents. Also, by representing terms and documents in the same space, vector-space models often provide an elegant method of implementing relevance feedback. Relevance feedback, by allowing documents as well as terms to form the query, and using the terms in those documents to supplement the query, increases the length and precision of the query, helping the user to more accurately specify what he or she desires from the search.

Advantages:

The vector space model has the following advantages:

- Simple model based on linear algebra
- Term weights not binary
- Allows computing a continuous degree of similarity between queries and documents
- Allows ranking documents according to their possible relevance
- Allows partial matching

Limitations:

The vector space model has the following limitations:

- Long documents are poorly represented because they have poor similarity values (a small scalar product and a large dimensionality)
- Search keywords must precisely match document terms; word substrings might result in a "false positive match"
- Semantic sensitivity; documents with similar context but different term vocabulary won't be associated, resulting in a "false negative match".
- The order in which the terms appear in the document is lost in the vector space representation.
- Theoretically assumes terms are statistically independent.

Weighting is intuitive but not very formal.

II. LITERATURE SURVEY

A. Personalized Web Search Engine using Dynamic User Profile and Clustering Techniques [01], Internet is large interconnection of small networks that is commonly known as World Wide Web. The amount of information available on internet in digital form are very huge and growing at exponential rate following Moore’s law. So, it’s makes difficult to find exact search result according to user preferences. In this paper, we proposed a method for personalized web search. Personalized web search is any action taken to optimize the search result according to user’s individual preferences. Different information retrieval techniques have been widely used to reduce access latency problem of the internet. This paper comprised and focuses different techniques for efficient personalized web search and also suggest the techniques for personalized web search according to the merits and demerits of various available techniques.

B. Improved Keyword Aware Service Recommendation System for Big Data Applications [02], Service recommender systems are valuable tools for providing appropriate recommendations to users. In keyword aware service recommendation system, keywords are used to indicate both of user’s preferences and quality of candidate services. A user based collaborative filtering algorithm is adopted to generate appropriate recommendations. In CF based systems, users receive recommendations based on people who have similar tastes and preferences .The preference of previous users are extracted from their reviews and formalized into a keyword set. An active user can give his/her preferences about candidate services by selecting keywords from a keyword candidate list ,which reflect the quality criteria of the services he/she is concerned about. The previous users who have similar tastes to an active user are found based on similarity of their preferences and recommendations are provided to the active user .But the system only depends on explicit user feedback, and thus is intrusive. It only considers user reviews and does not consider the temporal information about locations of the services.

C. Algorithms and Methods in Recommender Systems [03], today, there is a big variety of different approaches and algorithms of data filtering and recommendations giving. In this paper we describe traditional approaches and explain what kind of modern approaches have been developed lately. All the paper long we will try to explain approaches and their problems based on movies recommendations. In the end we will show the main challenges recommender systems come across.

D. Content-based Recommender Systems: State of the Art and Trends [04], recommender systems have the effect of guiding users in a personalized way to interesting objects in a large space of possible options. Content-based recommendation systems try to recommend items similar to those a given user has liked in the past. Indeed, the basic process performed by a content-based recommender consists in matching up the attributes of a user profile in which preferences and interests are stored, with the attributes of a content object (item), in order to recommend to the user new interesting items. This chapter provides an overview of content-based recommender systems, with the aim of imposing a degree of order on the diversity of the different aspects involved the different the different aspects involved in their design and implementation. The first part of the chapter presents the basic concepts and terminology of content- based recommender systems, a high level architecture, and their main advantages and drawbacks. The second part of the chapter provides a review of the state of the art of systems adopted in several application domains, by thoroughly describing both classical and advanced techniques for representing items and user profiles. The most widely adopted techniques for learning user profiles are also presented. The last part of the chapter discusses trends and future research which might lead towards the next generation of systems, by describing the role of User Generated Content as a way for taking into account evolving vocabularies, and the challenge of feeding users with serendipitous recommendations, that is to say surprisingly interesting items that they might not have otherwise discovered.

F. Optimal keyword search for recommender system in big data application [05], Currently online searching process increases and people searches new information in the search process. Most of the search engine gives additional supporting information. Recommender system involves in this process and implements as service. The proposed work analyses issues occurring when service recommender system implements in large data sets. This work proposes a keyword-Aware services Recommender method, to split the services to the users and mainly focused keywords from the user preferences. Hybrid Filter algorithm generates keyword recommenders from the previous user preferences. To implement effective results in big data environment, this method is implemented using the concept of Map Reduce parallel processing on Hadoop. Experimental results are shown the effective results on real-world datasets and reduce the processing time from large datasets.

G. A Product Recommendation System using Vector Space Model and Association Rule [07], **t**his paper presents an alternative product recommendation system for Business-to-customer e-commerce purposes. The system recommends the products to a new user. It depends on the purchase pattern of previous users whose purchase pattern are close to that of new user. The system is based on vector space model to find out the closest user profile among the profiles of all users in database. It also implements Association rule mining based recommendation system, taking into consideration the order of purchase, in recommending more than one product. To make the association rule memory- efficient, cellular automata is used.

H. Towards Keyword Based Recommendation System [08], Recommender systems have been shown as valuable tools for providing appropriate recommendations to users. In the last decade, the amount of customers, services and online information has grown rapidly, yielding the big data analysis problem for recommender systems. Moreover, most of existing recommender systems present the same ratings and rankings of items to different users without considering diverse users' preferences, and therefore fails to meet users personalized requirements. This project proposes a Keyword based Recommendation method, to address the above challenges. It aims at presenting a personalized recommendation list and recommending the most appropriate items to the users effectively. Specifically, keywords are used to indicate users' preferences, and a user-based Collaborative Filtering algorithm is adopted to generate appropriate recommendations. To improve its scalability and efficiency in big data environment, it is implemented on Hadoop, a widely-adopted distributed computing platform using the MapReduce parallel processing paradigm. Proposed system is used to improve the accuracy and scalability of service recommender systems over existing approaches.

I. A Keyword-Aware Service Recommendation Method on MapReduce for Big Data Applications [09], Service recommender systems have been shown as valuable tools for providing appropriate recommendations to users. Moreover, most of existing service recommender systems present the same ratings and rankings of services to different users without considering diverse users' preferences, and therefore fails to meet users' personalized requirements. In this paper, we propose a Keyword-Aware Service Recommendation method, named KASR, to address the above challenges. It aims at presenting a personalized service recommendation list and recommending the most appropriate services to the users effectively. Specifically, keywords are used to indicate users' preferences, and a user-based Collaborative Filtering algorithm is adopted to generate appropriate recommendations. To improve its scalability and efficiency in big data environment, KASR is implemented on Hadoop, a widely-adopted distributed computing platform using the MapReduce parallel processing paradigm. Finally, extensive experiments are conducted on real-world data sets, and results demonstrate that KASR significantly improves the accuracy and scalability of service recommender systems over existing approaches.

*J. Effective Hierarchical Vector-based News Representation for Personalized Recommendation [11***],** with amount of information on the web, users often require functionality able to filter the content according to their preferences. To solve the problem of overwhelmed users we propose a content-based recommender. Our method for the personalized recommendation is dedicated to the domain of news on the Web. We propose an effective representation of news and a user model which are used to recommend dynamically changing large number of text documents. We work with the vector representation of the news and hierarchical representation of similarities among items. Our representation is designed with aim to effectively estimate user needs and generate personalized list of items in information space. This approach is unique thanks its low complexity and ability to work in real-time with no visible delay for the user. To evaluate our approach we experimented with real information space of largest Slovak newspaper and simulated recommending.

III. PROPOSED METHODOLOGY

The big data is the concept of large spectrum of data, which is being created day by day. In recent years handling these data is the biggest challenge. Hadoop is an open source platform which is used effectively to handle the big data applications. The two core concepts of the hadoop are Mapreduce and Hadoop distributed file system (HDFS). HDFS is the storage mechanism and map reduce is the programming language. Results are produced faster than other traditional database operations. We proposed vector space model algorithm and this improve the data classification and prediction and make it uniform. Then apply modified K-Means clustering on input data which we get from above algorithm and output is stored in clustered form. K means reduce the number of comparison which makes execution faster. Clustered Data act as input for MapReduce. MapReduce apply Mapper, Combiner and Reducer Mechanism over data and eliminate duplicate data from large amount of data set. For test data the divide and conquer approach is applied on each row of the cluster. Divide and conquer technique is used to match records within a cluster which further improves the efficiency of the algorithm.

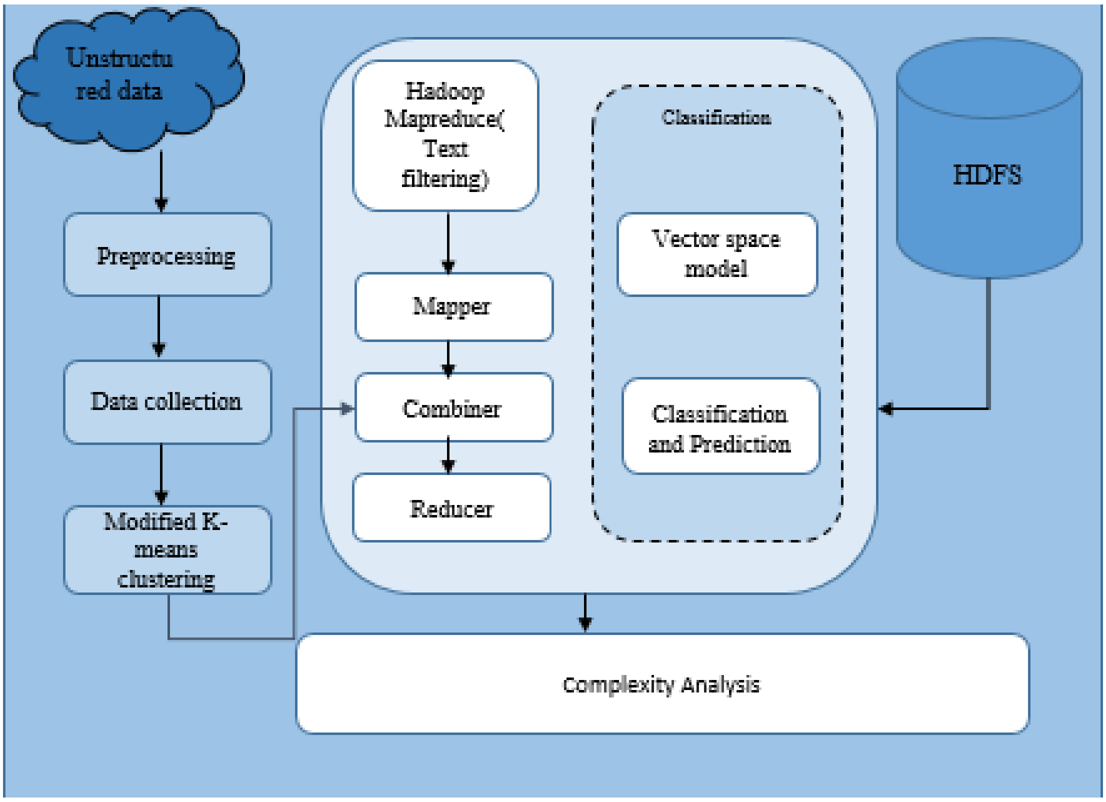

Fig. 1 Proposed system diagram

IV. CONCLUSION

Proposed model overcomes the limitations of the traditional filtering algorithm such as,

- Scalability
- Poor accuracy

Moreover, to improve the scalability and efficiency in "Big Data" environment, we have implemented it on a Map Reduce framework in Hadoop platform and Vector Space Model. In our future work, we will do further research in how to deal with the case where term appears in different categories of a domain thesaurus from context and how to distinguish the positive and negative preferences of the users from their reviews to make the predictions more accurate. The proposed system is more efficient in terms of complexity. And the system gives more accurate results or recommendations to the users.

REFERENCES

[1]. Anoj kumar, mohd.Ashraf, "Personalized Web Search Engine using Dynamic User Profile and Clustering Techniques", 2015 2ndInternationalConference on Computing for Sustainable Global Development (INDIACom)

[2]. Shakhy.P.S, Swapna.H, "Improved Keyword Aware Service Recommendation System for Big Data Applications", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 8, August 2015, ISSN (Print): 2320-9798

[3]. Daniar Asanov, "Algorithms and Methods in Recommender Systems"

[4]. Pasquale Lops, Marco de Gemmis and Giovanni Semeraro, "Content-based Recommender Systems: State of the Art and Trends"

[5]. J. Amaithi Singam and S. Srinivasan, "Optimal keyword search for recommender system in big data application", ARPN Journal of Engineering and Applied Sciences, VOL. 10, NO. 07, April 2015 ISSN 1819-6608

[6]. Ali Elkahky, Yang Song, Xiaodong He, "A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems", International World Wide Web Conference Committee (IW3C2), May 18–22, 2015, Florence, Italy, ACM 978-1-4503-3469-3/15/05

[7]. Debajyoti Mukhopadhyay, Ruma Dutta, Anirban Kundu and Rana Dattagupta, "A Product Recommendation System using Vector Space Model and Association Rule", International Conference on Information Technology, DOI 10.1109/ICIT.2008.48

[8]. Vinaya B. Savadekar, Pramod B. Gosavi, "Towards Keyword Based Recommendation System", International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064, Volume 3 Issue 11, November 2014

[9]. Shunmei Meng, Wanchun Dou, Xuyun Zhang, Jinjun Chen, "A Keyword-Aware Service Recommendation Method on MapReduce for Big Data Applications", DOI 10.1109/TPDS.2013.2297117

[10]. Pallavi R. Desai, B. A. Tidke, "A Survey on Smart Service Recommendation System by Applying Map Reduce Techniques", International Journal of Science and Research (IJSR), Volume 5 Issue 1, January 2016, ISSN (Online): 2319-7064

[11]. Maria Bielikova, Michal Kompan, and Dusan Zelenik, "Effective Hierarchical Vector-based News Representation for Personalized Recommendation", DOI 10.2298/CSIS110404070B

[12]. Malay K. Pakhira , "A Modified k-means Algorithm to Avoid Empty Clusters", International Journal of Recent Trends in Engineering, Vol 1, No. 1, May 2009

[13]. Hao Chen, Cheng Zeng, "Personalized Information Retrieval Model Based on User Interests", 2008 International Conference on Computer Science and Software Engineering, DOI 10.1109/CSSE.2008.171.