Big Data in Job Recommendation Systems

Huiyi Chen
Indiana University Bloomington
Bloomington, Indiana
huiychen@indiana.edu

Yuanming Huang
Indiana University of Bloomington
Bloomington, Indiana
huang226@indiana.edu

ABSTRACT

In recent years, there are more and more recommendation systems merged as time goes on. It brought us a lot of convenience and improved the efficiency of finding resources by the development of technology. However, with the rapid growth of information, people feel hard to find the exact information they need in job searching sometimes. Thus, the birth of Job Recommendation satisfied the demand of finding exact job information as soon as possible and make this process more and more automatically. So, in this project, we will explore the system of Job recommendation to help people improve the efficiency in finding job information. LaTeX

KEYWORDS

Big Data, Job Recommendation Systems, Collaborative filtering, content-based filtering, Java, HID101, HID230

1 INTRODUCTION

As we browse on the internet nowadays, we realized that many websites push side advertisements that are related to what we have just browsed seconds ago. We were wondering how the internet has gotten so smart, yet it was a trick that many technology companies have made use of to better sort through the data they have about users and to generate profits more efficiently. These companies use data to filter out things that users like to read, which enhance users' browsing experience and therefore, use the website more often.[?] The so-called recommendation system has been more and more popular and is now an integral part of many e-commerce sites such as Netflix, Amazon.com, Google, and etc. While these big commercial companies have been using the recommendation system for so long, we also want to analyze how some job websites can utilize this recommendation system to enhance user experience.

According to Wikipedia's definition, a recommendation system is a subclass of the information filtering system that seeks to forecast and predict the "preference" or "rating" that users would give to an item. In terms of a job recommendation system, it is the criteria of "interesting and useful" and "individualized" that separate the recommendation system from traditional search engine and information retrieval systems.[?] It is more personalized for users that they could easily find the information they need without future clicks and research.

In regards of job recommendation systems, we want the system to recommend related job ads to job seekers based on their click history and basic information that they inserted in into the website such as majors, work history, extracurricular activities, skill sets, and etc.[?] In other words, a job recommendation system has no difference than an automated form of a job search agency that not only it will show you a list of jobs that might fit you based on your information, but it will also utilized the relative jobs that related users have gotten or interested in who share the similar backgrounds as you do. The system is well trained in up-selling and cross selling, just like an integrated job search agency.[?] The fact that the job recommendation system could recommend personalized content based on past experience and related users' information bring users back to the website and keep using it.

2 IMPORTANCE OF JOB RECOMMENDATION SYSTEMS

With the rapid development and progression of the Internet in the world, people's lifestyle has undergone tremendous changes. In the past, people may need to look for employment information by reading newspapers and reading magazines, but nowadays, people's lifestyles are changed. They are more and more inseparable from the Internet and more and more employment information is directly put online. People start to search for jobs online as a more direct and convenient way. According to data from China Internet Network Information Center, as of 2014, the number of Internet users in China has reached 632 million.[?] As a result, optimizing network information has become an increasingly important demand all around the world. Because of the development of the network, the amount of information increased more rapidly. However, people's ability to choose information can not keep up with the explosive growth of information, which leads to a huge conflict between information growth and information selection ability of people. Consequently, how to make this process more efficient by designing a system is becoming more and more important. On the other hand, the rapid development of the Internet brings us into a new era of big data. On the Internet, there are more and more different types of jobs, different types of requirements, and different types of recruitment processes. Job seekers often spend a lot of time to do research and browse for jobs information. However, they may still need to do something that is not their ideal position. Therefore, how to help job seekers intelligently to find the information they want in a short time has significant meaning at the time.

Besides helping the job seekers, we also need to focus on the company side. With the growing number internet companies, the competition is strong, and the companies need to compete against a big number of competitors. In order to differentiate themselves from others, using big data is a great method. Big data helps the company to load their websites more personalized and intelligent for the users. It gives the users what they want instead of giving everyone the homologous information. The fact that the users can user fewer clicks and less research time to find the information that fit their needs will increase the use and favor for certain websites. With that, internet companies can quickly differentiate themselves and increase profit. On the other hand, since the use of big data and recommendation systems becomes more and more popular, the ignorance of the technology might result in getting behind from the competitors. As a result, recommendation systems is gaining high popularity, and companies need to learn how to properly utilize them.

3 RECOMMENDATION TECHNIQUES

According to Resnick, P. and Varian, H.'s âĂŸRecommender SystemsâĂŹ. Communications of the ACM, they mentioned that recommendation techniques have many possible classifications. It is not about the types of interfaces and the properties of users' interaction with the job recommendation systems, but it is more about the sources of data that the system is based on and also the use to which the data is put.[?] More specifically, a job recommendation system need to contain (i) background data, the information which the system retains before the job recommendation process starts, (ii) input data, the information which user must communicate to the system in order to generate a recommendation, for example,majors, work history, extracurricular activities, skill sets, and etc, and (iii) an algorithm that combines input and background data to arrive at its suggestions.[?]

3.1 Collaborative Filtering Algorithms

In order to understand how a job recommendation system work, we need to understand different recommendation systems approaches in order to pick the one that fit our need the most. Collaborative filtering methods are based on a large section of collecting and analyzing information on users' activities, preferences and forecasting what job seekers will like based on similar job seekers who share similar background.[?] One of an important advantages of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore, it is capable of accurately recommending complex items such as a data science job without requiring and "understanding" of the job itself.[?] Many algorithms are used in measuring item similarity and user similarity in recommendation systems.

Collaborative filtering algorithms are based on the assumptions that most of the people who agreed in the past will later agree in the future, and that they will tend to like similar types of items as they liked in the past.

When we are building a model from users' behaviors, we need to make a distinction between implicit and explicit forms of data collection.

Examples of the collection of implicit data could be be the following:

- Observing the jobs that users view in the past.
- Analyzing job/user viewing times.
- Keeping a record of the jobs that users apply online.
- Obtaining a list of jobs that users have read to or researched on their computers.
- Analyzing the users' social network and discovering similar likes and dislikes.

Examples of the collection of explicit data could be be the following:

- Asking users to rate an job on a number scale.
- Asking users to search.
- Asking users to make a rank of a collection of jobs from favorite to least favorite.
- Presenting two jobs to users and asking them to choose one of the better in between the two.
- Asking users to come up with a list of jobs that they like.

The job recommendation system will compare the implicit and explicit data to similar and dissimilar data collected from outside resources and calculates a list of recommended jobs for the users.

Collaborative filtering methods often have these three drawbacks which are cold start, sparsity, and scalability.[?]

- Cold start: These recommendation systems often require a great amount of pre-existing data on users in order to make some accurate recommendations.[?] That means, if the data of one user is not comprehensive enough, there is a great possibility that the recommendation by the system does not align with the user's interest
- Sparsity: The number of jobs posted on major job search sites is extremely large. The most involved and active users will only rate a small subset of the entire database. With that being said, even the most popular jobs will only have very few ratings.
- Scalability: In many of the system environments in that these recommendation systems make recommendations, there are over millions of users and jobs, which means a large amount of computation power is definitely required and necessary to calculate all of the recommendations for all the users.

Collaborative filtering approaches are classified as model based and memory-based collaborative filtering.[?] A well-known example of memory-based approaches is user-based algorithm and that of model-based approaches is Kernel-Mapping Recommender.

3.2 Content-based Filtering Algorithms

Another popular approach that data scientists like to use to design job recommendation systems is content-based filtering. Content-based filtering approaches are based on profiles of the usersâĂŹ preferences and a description of the job item.[?] In a content-based recommendation system, keywords are implemented to describe the jobs and users' profiles are built to indicate the types of jobs these users like.[?] To put in a different way, this algorithm tries to recommend jobs that are similar to those that users viewed and liked in the past. In particular, various potential jobs are put together to compare with jobs previously rated by the users and the best-matching jobs are recommended. This particular approach has its origins in information filtering research and information retrieval.

To epurate the features of the jobs in the recommendation system, an job presentation algorithm would be applied. A widely popular used algorithm is the tfâĂŞidf representation.

To create a user profile and save it in the database, the recommendation system mostly wants to focus on two kinds of information:

1. A model of the users' preferences.
2. A history of the users' interactions with the recommendation system.

Generally, these methods use an job profile characterizing the job within the recommendation system. The system will create content-based profiles of users based on the item features. There might be different weights to every item in terms of users' preference. The weights that are assigned to each of the features depending on the users' preference can be computed and calculated from individually rated content using a series of techniques.[?] Simple approach uses the average rates of the item vector while other sophisticated approaches use machine learning techniques such as cluster analysis, Bayesian Classifiers, artificial neural networks and decision trees to calculate or estimate the potential probability that the users are going to like the job.[?]

The feedback the system got directly from a user, usually in the form of a like or dislike button, can be used to allocate lower or higher weights on the importance of specific attributes.

An significant drawback with content-based filtering is that whether the recommendation system is actually capable to learn user preferences from users' historical actions regarding content sources and use the sources across all the other content types.[?] When the recommendation system is limited to recommending jobs of the same types that the user is currently using, the retrieved value from the system is significantly less than that when another content type from other retrieving services could be recommended.[?] For instance, recommending current news articles based on historical browsing of news is definitely useful, but it would be more useful when products, music, videos, discussions and etc. from different retrieving services could be recommended based on the news browsing.

A great example of content-based filtering approach being used in the real world is Pandora Radio. It plays music based on the user's initial feed to the recommendation system and deliver recommended music with similar characteristics. Besides that, most of the movie, music, and book recommendation systems are based on content-based filtering algorithm since this particular one works the best for personalization based on historical data.

3.3 Hybrid Recommendation Systems

Recent researches have demonstrated that hybrid approaches, combining both content-based filtering and collaborative filtering could be much more effective in many cases. Hybrid approaches could be implemented in the following ways: by adding collaborative-based approach to a content-based capabilities , or vice versa; by collaborative-based making and content-based predictions separately and later combining them; or by unifying both approaches into one integrated model.[?] Recent studies empirically compare the performance of the hybrid recommendation systems with the pure collaborative and content-based methods and thus demonstrate that the hybrid approaches can provide much more accurate recommendations than the pure approaches.[?] These approaches can also be used to overcome many of the common problems that happen in pure approach recommendation systems such as sparsity and the scalability problem.

One of the examples of the use of hybrid recommendation systems is Netflix. Although many movie and music recommendation systems use pure content-based approaches, Netflix chooses the strong hybrid approach that makes recommendations by using collaborative filtering, comparing the searching and watching behaviors of similar users as well as by content-based filtering, offering shows and movies which share similar characteristics with content that users have highly rated.[?]

A variety numbers of techniques have been proposed as the fundamentals for recommendation systems: content-based, collaborative, demographic and knowledge-based techniques. Every single one of these techniques has some sort of drawbacks, such as the well-known scalability problem for collaborative approach and biased rating system for content-based systems.[?] A hybrid recommendation system is one that integrates multiple techniques together in order to achieve some synergy between all of the approaches, minimizing the influences of those drawback from a specific method.

- Content-based: The system generates recommendations from particularly two sources: the ratings that a user has given them and the features associated with items. Content-based recommendation system treats recommendations as user-specific classification problems and learns a classifier for the users' likes and dislikes based on item features, which in our case, the job features.[?]
- Collaborative: The system provides recommendations using exclusively information about the rating profiles from different users or jobs. Collaborative system locates similar users / jobs with rating histories similar to the incumbent user or job and generate recommendations using the neighborhood.[?] The users based and the items based nearest neighbor algorithms can be integrated to deal with the cold start problem and thus, improve recommendation results.
- Knowledge-based: A knowledge-based recommendation system suggests jobs based on inferences about users' preferences and needs.[?] This knowledge will contain explicit functional information about how certain job features would meet user needs.
- Demographic: A demographic recommendation system generates recommendations based on a demographic profile of the users. Recommended jobs can be retrieved for different demographic niches, combining the ratings of users in those niches.

The term hybrid recommendation system is used to describe any recommendation system that would combine multiple recommendation techniques as above together to produce the output.[?] We can also combine several different techniques of the same type, for instance, two different content-based recommendation systems could function together, and many of the projects have investigated the type of hybrid: NewsDude, which has both kNN classifiers and naive Bayes in its news recommendations, is a good example.

There are seven hybridization techniques that are popular:

- Weighted techniques: The scores of different recommendation elements are combined numerically.
- Switching techniques: The recommendation system chooses among all the recommendation components and applies only the selected one.[?]
- Mixed techniques: Recommendations from different recommendation systems are presented all together.

- Feature Combination techniques: Features that are derived from different knowledge resources are combined all together and delivered to a sole recommendation algorithm.
- Feature Augmentation techniques: Only one recommendation technique is used to calculate a set of features or a feature, which is part of the input to the following technique.
- Cascade techniques: Recommendation systems are given strict pre-existed priority, with the lower priority ones breaking ties in the scoring of the higher ones.[?]
- Meta-level techniques: Only one recommendation technique is applied at a time and produces a draft model, which is later the input that is used by the next technique.

4 HOW TO IMPLEMENT RECOMMENDATION SYSTEM

Recent researches has shown that there are increasing demands of Information Systems technologies for human resource management in the recruiting process in particular.

(1) User Information Acquisition and Modeling: Because users have different interests and different industry preferences, we need to deal with the log files, find out users' explicit and implicit requirements, and then analyze and build the user mold.[?]
(2) Model Design and Implementation: At this stage, the main contents include the combination of feature variables, similarity calculation, positive and negative samples of mobile phones and internet devices, weight value calculation and knowledge classification logistic regression.[?]
(3) System Design and Implementation: The user model and big data platform combine to meet the needs of the company's job recommendation system.[?]
(4) System Verification and Comparison: The group calculates the conversion rate off-line, determines the characteristic variable combination and the similar algorithm of the recommended model and uses the filtering recommendation algorithms to compare and verify to get the optimal combination recommendation system.[?]
(5) The Application and Research of the System: It is necessary to establish the application framework of the recommendation system in other application fields to study how to integrate with other business systems of the enterprise and to realize the diversification of the recommendation system.[?]

In order for the system to work, we need to utilize Hadoop as the platform for the recommendation system to be running on.

4.1 Based on Hadoop

The realization of our project is based on the Hadoop platform, so here we need to do a detailed study of Hadoop about the source and role of this platform. Hadoop is an open source framework for writing and running distributed applications for large-scale data, designed for offline and large-scale data analysis.[?] It is not suitable for online transactions that randomly read and write to several records Processing mode. Just like HDFS (file system, data storage technology related) + Mapreduce (data processing), Hadoop data sources can be any kind of form and have better performance in dealing with semi-structured and unstructured data than relational databases, With more flexible processing power, no matter what form of data will eventually be converted to key / value. Key / value is the basic data unit.[?] MapReduce can be used to replace SQL, the standard query language, and MapReduce uses scripts and code, while Hadoop, which is used to relational databases and custom SQL, has an open source hive instead. Thus, we can understand that Hadoop is a distributed computing solution.

Hadoop features: Hadoop good log analysis, facebook to use Hive for log analysis, in 2009 Facebook had non-programmers, 30 pecent of people use HiveQL for data analysis; China's Taobao search custom filtering is also used Hive; With Pig you can also do advanced data processing, including Twitter, LinkedIn to discover people you may know, and he can also implement recommendations similar to Amazon.com's collaborative filtering.[?] And China's Taobao's product recommendation is such a process. At Yahoo, 40 percent of Hadoop jobs are run on pigs, including spam identification and filtering, and user feature modeling.[?]

(1) Data integration
Data consolidation is called "enterprise data center" or "data lake." When users have different data sources, want to analyze their data. Such projects include getting data sources (real-time or batch) from all sources and storing them in hadoop.[?] Sometimes this is the first step to becoming a "data-driven company"; sometimes you may only need a beautiful report. Enterprise Data Centers typically consist of HDFS file systems and tables in HIVE or IMPALA. In the future, HBase and Phoenix should have bigger development in big data integration and create a new situation to create a brand new world of beautiful data.[?]
Often, salespeople love to say "read patterns," but in fact, to be successful, you have to know exactly what your own use cases will look like (Hive patterns do not look like you did in your enterprise data warehouse).[?] The real reason is that a data lake has more horizontal scalability and much lower costs than Teradata and Netezza. Many people use Tabelu and Excel when doing front-end analysis.[?] Many sophisticated companies use "data scientists" as front ends with Zeppelin or IPython notebooks.
(2) professional analysis
Many data integration projects actually start with the analysis of specific needs and a data set system. These are often incredibly specific areas such as liquidity risk / Monte Carlo simulation in the banking sector.[?] In the past, this professional analysis has relied on outdated, proprietary software packages, which often suffer from a limited feature set due to the inability to scale the data (largely because software vendors can not understand as much as professional organizations do).[?]
In the world of Hadoop and Spark, take a look at these roughly the same data consolidation systems, but often have more, if not unique, HBase, custom non-SQL code, and fewer sources of data. More and more based on Spark.
(3) Hadoop as a service

Any large organization in a "professional analytics" project will inevitably start to feel "happy" (ie, ache) managing several differently configured Hadoop clusters, sometimes from different vendors.[?] Next, they will say, "Maybe we should integrate these resource pools," rather than leaving most of the nodes idle most of the time. They should make up cloud computing, but many companies often can not or do not because of security reasons. This usually means a lot of Docker container packages.

(4) Flow analysis

In general, flow analysis is a real-time version of an organization's batching.[?] With anti-money laundering and fraud detection: why not on a transactional basis, take hold of it and not end it in a cycle? The same inventory management or anything else.

In some cases, this is a new type of trading system that analyzes the bits of a data bit because you are paralleling it into an analysis system. These systems prove themselves as popular data stores such as Spark or Storm and Hbase.[?] But flow analysis does not replace all forms of analysis, and for things you have never considered, people often want to analyze historical trends or look at past data.

(5) Complex event handling

Here we are talking about sub-second real-time event processing. While there is not yet fast enough for ultra-low latency (picosecond or nanosecond) applications like high-end trading systems, millisecond response times can be expected.[?] Examples include real-time evaluation of call data records processed by internet telecommunication carriers for a thing or event. Sometimes you see that such systems use Spark and HBase, but in the end it usually has to be converted to Storm based on the interference mode developed by LMAX Exchange.[?]

In the past, such systems have been based on custom messages or high performance from shelves, client-server messaging products - but today's data is overloaded. I have not used it yet, but the Apex project looks promising, claiming to be faster than Storm.

(6) ETL flow

Sometimes when we want to capture the stream data and store them up. These items usually coincide with No. 1 or No. 2, but add to their scope and characteristics.[?] These are almost Kafka and Storm projects. Spark is also used, but there is no reason, because no memory analysis is required.

(7) Change or add SAS

We do not need to buy storage for your data scientists and analysts and you can "play" the data. In addition, you can do a few different things besides SAS can do or produce beautiful graphical analysis.[?] This is your "data lake." Here is the IPython notebook (now) and Zeppelin (later). We use SAS to store the results.

These are all normal when I see other different types of Hadoop, Spark, or Storm projects every day. If you use Hadoop, you probably know about them. Some years ago I had implemented some of these projects, using other technologies. Although more and more things change, but the essence remains unchanged. We will find a lot of similarities, things you used to deploy and trendy technology are around the Hadoop Sphere rotation.

5 MAP AND REDUCE

Suppose the user wants to count a huge text file stored on a similar HDFS, want to know the frequency of occurrence of each word in this context.[?] So to start a MapReduce program, n the Map stage, hundreds of machines will read all parts of this file at the same time and separately counted the frequencies of the parts they read separately, resulting in pairs like (hello, 1100), (world, 1214) These hundreds of machines each produced the same set, and then hundreds of other machines started Reduce processing. Reducer Machine A will receive all statistical results starting with Mapper Machine A, and Machine B will receive the vocabulary statistics beginning with B (but in fact it does not begin with a letter, but uses the function to generate a Hash value to avoid data Stringing.[?] Since words beginning with X are certainly far fewer than the others, and we expect the workload of the data processing machines to differ too much). These reducers will then summarize again, (hello, 1100) + (hello, 1311) + (hello, 35881) = (hello, 38291). Each Reducer will do this, and eventually we get the word frequency result for the entire document.

This is a seemingly simple model, but many algorithms can be described using this model. The simple model of Map + Reduce, though easy to use, is also very cumbersome. The second generation of Tez and Spark In addition to new features such as memory caching, essentially the Map / Reduce model is more generic, blurring the boundaries between Map and Reduce, making data exchange more flexible and with less disk reads Write in order to more complex description of complex algorithms to achieve higher throughput.

With MapReduce, Tez, and Spark, programmers find it really troublesome to write MapReduce programs. So we want to simplify this process. We would like to have a higher level and more abstract language layer to describe the algorithm and data processing flow.[?] So there is Pig and Hive. Pig is close to the script to describe MapReduce, Hive is using SQL. They translate scripts and SQL into MapReduce programs and throw them into computational engines for computation, and we're freed from cumbersome MapReduce programs and written in simpler and more intuitive languages.[?]

With Hive, people found that SQL has a huge advantage over Java. One is that it is too easy to write. Just word frequency things, described in SQL on only one or two lines, MapReduce write about dozens of hundreds of rows.[?] More importantly, users with non-computer backgrounds write SQL, so data analysts are finally freed from the dilemma of begging engineers and engineers are freed from the weird one-off handlers.[?] This result makes the whole process more efficient. Hive has evolved into a big data warehouse core components. Even many company pipeline assembly is entirely SQL described, because easy to write and easy to maintain.

Since data analysts began to analyze data with Hive, they found that Hive was running too slow on MapReduce. But data analysis, people always want to run faster. For a huge site of massive data, this process may take dozens of minutes or even hours.[?] And this analysis may be only a small part of the need to analyze how many people browse the electronic products, analysis of how many people read the Rachmaninoff CD, and so on, and then come to our proportion of the type of user. Due to the high demand for speed, the new Impala, Presto, Drill was born (and of course innumerable non-famous interactive SQL engines). The core idea of the three systems is that the MapReduce engine is too slow because it is too generic, too strong, too conservative, and we SQL needs a lighter and faster access to resources, more specialized SQL optimization, and less Fault Tolerant Assurance (because of a system error, we can restart the task, if the entire processing time is shorter, such as a few minutes). These systems allow users to more quickly handle SQL tasks, sacrificing features such as general stability.

But in fact these systems, has not reached the desired level of popularity. Because at this time two new ones were made. They are Hive on Tez / Spark and SparkSQL. Their design philosophy is because MapReduce is slow, but if I run SQL with a new generation of general purpose computing engines like Tez or Spark, then I can run faster. And users do not need to maintain two systems. The above introduction, the basic structure is a data warehouse. The underlying HDFS runs above MapReduce / Tez / Spark and runs Hive, Pig on it. Or run Impala, Drill, Presto directly on HDFS. This solves the low-speed data processing requirements.

6 DESIGN SYSTEM MODEL

As a kind of data mining, recommendation system is one of the more special data mining systems. He embodies the system and user interaction and real-time.[?] Recommend interest-based objects to users based on their hobbies or browsing behaviors, and further correct and optimize the recommendation results based on the feedback results of user interaction.[?] In this professional recommendation system, there are mainly three parts, data collection, offline data processing and real-time online recommendation.

6.1 Data Collection

In the process of data collection. Because there are many ways for users to provide their preference information to the system, they can be divided into two kinds of explicit and implicit information.[?] This information forms the basis of user behavior analysis. In this project, the main sources and channels of data are information about job-seekers registering, browsing jobs, and web-logging for job postings. User behavior categories: registration, browsing, residence time, job application.[?] Their respective types of information are: explicit, implicit, implicit, implicit. The following is an explanation of the characteristics and actions of the four user behaviors.

Registration: job seekers registered behavior, including the basic characteristics of job seekers, registration information we can get job preferences, Through job seekers' preferences, we can get more precise career preferences.

Browse: job seekers on the job browsing information, through the frequency of the frequency of statistics, job seekers get the preference. This process can to some extent reflect their concern about job postings and the likelihood that they will be interested in positions. Thereby enhancing the accuracy of the analysis

Dwell time: The user's dwell time information analysis, you can know whether the user is interested in the content of the visit and the degree of concern, so as to get their preference information. The longer you stay on a page, the more likely they are to be interested in the content of the page, as well as the level of attention. However, there are occasional noise data that is difficult to use based on this standard.

Job Application: Boolean preferences, the value is 0 and 1. This information can be used to determine whether the user is interested in this position.

6.2 Offline Data Processing

Generally, the historical data of job seekers will be very large. Therefore, if the system wants to analyze massive data online and recommend it in real time, it is unrealistic. Therefore, if offline processing of data can make the data processing more Efficient and easy to implement, when we have collected enough user behavior data, we can pre-process the data off-line, such as noise reduction, and then analyze the user's behavior log and train user profiles through recommendation strategies.[?] Finally, offline calculation of the user's character data and get the initial recommendation seen, the next step, the recommended results provided to the online implementation of the recommended use.

6.3 Online Real-Time Recommendation

The online real-time recommendation is to analyze the user's real-time behavior in a very short period of time and give the recommended result. Therefore, the online recommendation system and offline processing are two different processes and concepts.[?] The online real-time recommendation can not process the user's historical behavior log and can not handle too complicated data.[?] The online real-time recommendation usually deals with simple data, for example, querying the basic information of job seekers, job seekers applying, And then combine the result of the analysis with the user characteristic data that has been processed offline to get the final recommendation result to the job seekers through the multi-dimensional analysis of filtering, screening and recommendation ranking.

7 TECHNICAL MODEL

In order to build a job recommendation system, we need to implement a technical model. There are several things that we need to pay attention to when building the model.

7.1 Similarity retrieval technology

Content similarity retrieval technology refers to comparing the text feature information in a resource with the text feature value of a user's interest, comparing the user's historical preference information with the content feature of a resource, and calculating the similarity, filtering out to meet the user's search expectations.[?] An example of content similarity retrieval technology. First, you need to model the content and attributes of your position, for example, by industry, function, job type, and place of employment. Then through the characteristics of each position data, we can find the similarity between positions.[?] If the professions, functions, types of jobs and workplaces are the same, we can think of these two posts as similar.

7.2 Demographic Collaborative Recommendation Technology

Demographic information can be viewed as a kind of user knowledge information that can be used to determine similar equivalence across networks, so that demographic information can be considered as a synergistic approach.[?] Collaborative demographic recommendation process: First, establish a data model for job seekers. Then according to job seekers model to calculate the similarity between job seekers. Find job seekers with the highest degree of similarity. Finally, recommend jobs to current job candidates based on their preferences.[?] As a result, demographic data can initiate a referral system even when job seekers do not have a feedback evaluation of the position.

7.3 Collaborative Filtering Recommendation Techniques

Collaborative Filtering Recommendation Techniques is a technique that is widely used to predict user interest preferences.[?] Its basic principle is based on the user's preference for the object, found the relevance of the object or user, and then recommend based on these correlations. The recommendation of collaborative filtering consists of three components: they are, item-based recommendation, user-based recommendation and model-based recommendation.[?] In this project, we mainly study this work recommendation system based on the idea of collaborative recommendation. That is to say, we need to train the recommendation model based on the sample data of job seekers' preferences and then make predictions based on the real-time information of job seekers Calculate recommended.

7.4 Big data processing technology

In this professional recommendation system, we assume that the big data processing framework that we need to use is the log acquisition system Flume, the big data platform Hadoop, the streaming computing framework Storm, and the message cache system Kafka.

Below we will introduce these different frameworks to understand the function and features of them.

7.4.1 Flume. Flume is a distributed, reliable, and highly available mass log collection, aggregation and delivery system. In this project, we anticipate that all the real-time input data needed will be realized through this technology platform. An Agent is the basic component of a Flume stream.[?] An Agent contains Source, Channel, Sinks, and other components that use these components to pass an Event from one node to the next or for the final purpose. The data is finally encapsulated into an Event for transmission. The Source is used to accept an external source.[?] For example, the Apache server delivers an Event to the source. The channel is used to temporarily store the Event. Sink then outputs the data. Other components can add pretreatment and classification capabilities.

7.4.2 Hadoop. At present, the large number of user data are obtained based on the site log records, job site for very large number of users, a simple single machine is difficult to complete offline data processing, so this project will be presumably we have introduced a distributed Framework to deal with, Hadoop is one of the most popular distributed framework today, which includes the distributed file system HDFS and distributed computing framework MapReduce and so on.

7.4.3 Storm. Storm is Twitter's open source distributed real-time computing flow data processing system, the calculation model is Topology as a unit, and a Topology is a series of Spout and Bolt formed by the graphic structure. Events Stream will flow between Spout and Bolt, Spout will generate Event, and Bolt will logically process the received Event and get the result of the calculation.

7.4.4 Kafka. Kafka is a high-throughput, distributed messaging subscription system that is organized as a topic and can be used as an active data and offline processing system for real-time processing of caches between systems for offline and online data users Provide data pipes to handle activity data from different sources.[?]

7.5 Mixed Recommendation Mechanism

In the popular web site, generally in order to achieve a better recommendation, we tend to mix a variety of recommended methods, rather than simply using a particular recommendation mechanism.[?] This mechanism for mixing multiple recommended methods and strategies is called, Hybrid Recommendation Mechanism.

8 OVERALL GOAL OF THIS SYSTEM

In today's social conditions, job seekers want to find a suitable job is not easy. therefore. We need a home-based recommendation system for personal job seekers' information intelligence that uses big data mining and analytics technologies to change the traditional hiring process. With these ideas, we know that applying personalized recommendation techniques to job hunting, work becomes easier, more efficient and smarter.[?] In order to complete this system, we need to accomplish the following goals:

(1) The recommendation system needs to intelligently collect explicit user needs and invisible user expectations information based on the confidentiality of user privacy information, and provide an effective data foundation for offline data processing and online real-time recommendation.
(2) The recommended system should be able to handle a large amount of behavioral log data without affecting its normal use. Learning offline interests and training feature models through big data technology to provide data support for online real-time recommendation.
(3) Recommended system must be based on user behavior online, rapid response to analyze user needs, in a timely manner to the user to make the list of their interest.

9 RECOMMENDATION SYSTEM FUNCTION DESCRIPTION

In this project, the user's occupation model, user interest in job hunting, job similarity, collection of positive and negative samples, machine learning and other algorithms related to the calculation of the model are based on the offline data processing module.[?] However, its research is done under massive log file processing. Therefore, the system's offline data processing module is deployed on the Hadoop platform. As mentioned by the big data processing technology can know, Hadoop is one of the most popular distributed framework, including the distributed computing framework MapReduce and distributed file system HDFS.

Offline data processing results is to achieve career recommendation system online real-time recommended services important data support. The module externally requests the data from the data collection module by reading the request and reading the data from the data collection module. The internal data processing request is executed through the general control interface.[?] Various implementation algorithm models are run on the Hadoop platform and offline by MapReduce and HDFS. , The final results will be stored in the HBase database. Offline data processing module is mainly composed of the Hadoop platform running algorithms and algorithms to achieve the algorithm model, and the algorithm code, and by a calculation of the total control interface implementation of the call.

The essence of running algorithm is distributed framework Hadoop platform, which is mainly composed of MapReduce distributed computing framework and distributed file system HDFS. Job log files of job seekers are stored on HDFS, the log file data is cleaned by Pig Latin, formatted and stored in data warehouse Hive, and then processed by Hadoop Streaming stream or Hadoop Java API to calculate the job similarity, the job seeker's interest, etc. Series of MapReduce programs, model results processed offline will eventually be stored in the HBase database to provide data support for real-time analysis.

The implementation algorithm mainly includes the realization of the user-job model and the realization of the weighting coefficient. In this project, it takes a large amount of computation and time-consuming tasks to be completed offline. One of the most prominent features that need to be done off-line, such as user-job model calculation and machine learning, is that computing requires the user's historical behavior log, so implementing this aspect requires a computing platform that runs algorithms.

10 PERFORMANCE MEASURES

After the implementation of the job recommendation system, evaluation is always necessary and important in evaluating and assessing the effectiveness of the recommendation algorithms that we implemented.[?] The most commonly used evaluation metrics are the root mean squared error and mean squared error, the former having been used in the Netflix Prize, one of the key events that energized research in recommendation systems. Such information retrieval metrics as recall and precision or DCG are very useful to assess the quality and performance of recommendation approaches.[?] Recently, diversity, coverage, and novelty are also considered as important aspects in evaluating the recommendation systems. Results of so-called offline evaluations often do not correlate with actually assessed user-satisfaction.

To evaluate recommendation systems, we can use the popular concept of precision-recall. We need to be familiar with this in terms of the idea and classification is very similar.

- Recall:
  - What ratio of jobs that users like were actually recommended.
  - If a user likes 10 jobs and the recommendation decided to show 6 of them, then the recall is 0.6
- Precision
  - Out of all of the recommended jobs, how many does the user actually like?
  - If 10 items were recommended to the user out of which he or she liked 8 of them, then precision is 0.8

An ideal job recommendation system is the one that only recommends the jobs which a user likes. So in this case, precision=recall=1. This is an optimal recommendation, and we should try and get as close as possible.

11 CONCLUSIONS

In this paper we wrote, we used a comprehensive literature analysis of many fundamental elements, problems faced, and technical model related to the job recommendation systems. We also analyzed other popular recommendation systems that are widely used nowadays, and tried to implement the advantages of those systems into the job recommendation system. The job recommendation system technologies will accomplish significant success in broad ranges of applications and will be potentially a powerful searching and recommending techniques. Consequently, there is a great opportunity for applying these technologies in recruitment environment to improve the matching quality. Additionally, in order to have readers understand job recommendation problem, we detailed a series of recommendation techniques, how to implement the job recommendation system, and performance measurement methods. Finally, we plan as a continuation of this work to present a designed algorithm of job recommendation approaches that have been proposed to produce the best fit with design order and technical model.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Gregor von Laszewski for his support to write this paper as well as TAs' helpful suggestions on this paper.

REFERENCES