Mathematical Biosciences and Engineering, 16 (4): 2906–2926.
DOI: 10.3934/mbe.2019143
Received: 16 January 2019
Accepted: 12 March 2019
Published: 10 April 2019

http://www.aimspress.com/journal/MBE

Research article

Research on massive information query and intelligent analysis method in a complex large-scale system

Dailin Wang*, Yunlei Lv, Danting Ren and Linhui Li

Northeast Forestry University, Harbin, 150040, China

* **Correspondence:** Email: darling_wang@163.com; Tel: +86-451-82191528.

Abstract: With the rapid growth of big data and network information, it is particularly important to perform information query and intelligent analysis on unstructured massive data in large-scale complex systems. The existing methods of directly collating, sorting, summarizing, and storing retrieval of documents cannot meet the needs of information management and rapid retrieval of massive data. This paper takes the standardized storage, effective extraction and standardized database construction of massive resume information in social large-scale complex systems as an example, and proposes a massive information query and intelligent analysis method. The method utilizes the semi-structured features of the resume document, constructs the extraction rule model of various resume data to extract the massive resume information. On the basis of HBase distributed storage, with the help of parallel computing technology to optimize the storage and query efficiency, which ensures the intelligent analysis and retrieval of massive resume information. The experimental results show that this method not only greatly improves the extraction accuracy and recall rate of resume information data, but also compared with the traditional methods, there are obvious improvements in the three aspects of massive information retrieval methods, query usage efficiency, and the intelligent analysis of complex systems.

Keywords: unstructured information; extraction rule model; HBase based distributed storage; information intelligence system

1. Introduction

With the in-depth application of the Internet and mobile Internet, complexity of social systems is increasing, and a large number of multi-channel and multi-structure types of massive document data have been grown and accumulated. Therefore, it is particularly important to conduct information

query and intelligent analysis on unstructured massive information. Taking resume documents as an example, the number of them will reach tens of millions or even more every year. How to store storage management and efficient information retrieval of massive resume documents more effectively has become one of the problems in the field of intelligent intelligence analysis systems. The traditional resume document management method stores the electronic resume on the hard disk or the network disk, and manually reads and filters the information. However, for a large number of resume documents, this method requires a lot of manpower and time cost, and the efficiency cannot meet the demand.

Text information extraction technology [1] is the automatic extraction of relevant or specific types of information from text, which is one of the important topics of pattern recognition and artificial intelligence. Information extraction aims to cope with the challenges brought by the explosive growth of information, helping people to use more massive information and to fully exploit the value of information. Text information extraction is an important part of the field of natural language processing. At present, the research results of the mainstream text information extraction model include the following three types: dictionary-based text information extraction model, rule-based extraction model and Hidden Markov Model-based extraction model. The dictionary-based extraction model [2] requires high dictionary establishment and needs to accumulate experience in a specific field. The definition of the best dictionary is complicated, the scope of application is narrow, and the versatility is poor. Therefore, it is often used as an auxiliary with other methods. Rule-based information extraction is a phased process of establishing constraint rules by defining target information constraint features and applying the rules to the extraction process. Rule-based information extraction is simple but requires manual programming of rules for specific languages, domains, and text formats, while its robustness and portability are poor. The Hidden Markov Model is based on training data and is based on a Hidden Markov Model with a strong statistical basis to construct the extraction algorithm. In 1996, Della Pietra and other people first applied the maximum entropy method [4,5] to the establishment of a language model for natural language processing in the reference [3]. The maximum entropy method is suitable for solving the classification problem in information extraction, and takes into consideration the effect of feature information on improving the performance of text information extraction. However, the maximum entropy model also has problems such as large amount of calculation and sparse data. Another statistical-based information extraction method uses the Hidden Markov Model that does not require large-scale dictionary set rule set to extract text information. It has a wide application field and good portability, but the accuracy and efficiency of text information extraction still need to be improved. Reference [6] extracted header information such as title, author and abstract of a computer science research paper by HMM; Reference [7] proposed to combine the Maximum Entropy Markov Model to achieve the extraction of the text information. In reference [8], Liu Yunzhong and others first analyzed the text layout format, separators and other information, and used these features to segment the text, and combined the Hidden Markov Model on the basis of the block to extract the text information. Reference [9] took the characteristics of web page data into consideration and combined the unique attributes of web page with the Hidden Markov Model to extract text. In reference [10], in order to improve the applicability and extraction performance of the algorithm for different formats of text in different fields, a clustered Hidden Markov Model text information extraction algorithm was adopted. In reference [11], a method of geographical name extraction based on hybrid Hidden Markov Model was proposed to extract geographical names quickly and accurately from the

metadata of literature.

Most of the above studies on text information extraction focus on the extraction of free text information without any structure. The semi-structured text extraction also concentrated on the information extraction of semi-structured text such as Web pages, and there is very little literature on the information extraction of resume documents. In the extraction of resume information, Ciravegna and Lavelli [12] studied the use of the (LP) 2 toolkit to learn the rules for the extraction of English resumes, which mainly extracts names, streets, cities, provinces, e-mails, faxes and postal codes. In terms of Chinese resume extraction, Yu Kun et al. [13,14] proposed an automated extraction method for resume information based on two-level cascade text classification. In 2011, reference [15] proposed a Wikipedia-based entity relationship extraction method to extract resume information such as background, education, job and other resume information in the web page resume data. In reference [16], Ren Ning proposed to apply the knowledge base's rule to design positioning information to study the extraction of the title information of the characters. In 2017, reference [17] proposed the word segmentation and part-of-speech tagging of resume sentences, which were expressed as feature vectors. A method of classifying sentences uses a classification algorithm, but the classified categories only have six general categories. In conclusion, there are still some problems in the extraction of resume information, such as single extraction item, poor portability of extraction algorithm and unsatisfactory extraction effect. It is still necessary to continuously optimize the relevant extraction algorithms and models in practice. However, in the extraction and storage management of massive text information, there are still problems of poor adaptability of algorithms, large amount of calculation, low accuracy of data retrieval recall rate. Therefore, how to achieve efficient and accurate automatic extraction processing and storage management of massive resume data is still a problem that needs to be solved urgently.

At present, the premise of the natural language processing method is to be able to effectively read the input text. However, in the process of processing the resume information, the key question is how to turn different types of resume into texts that can be effectively processed. In this paper when the resume is turned into text, our research will focus on how to quickly and effectively query the transformed text. This is the main work of this paper. To this end, this paper proposes a HBase-based massive resume data distributed storage. The query method is designed to improve the accuracy of the resume information extraction and the efficiency of the query.

(1) According to the semi-structured features of the resume text, the resume text information extraction rule model is constructed, the reasonable and effective text information extraction algorithm is designed, the backtracking algorithm is proposed to ensure the special information extraction effect, and the information extraction accuracy rate and recall rate of the resume text is improved.

(2) This paper proposes HBase-based data storage, which utilizes the characteristics of distributed non-relational database columnar storage to reduce the storage cost of massively sparse resume data, and optimizes query efficiency by means of parallel computing technology to meet the storage and query requirements of massive resume text information.

2. Rule model of resume information extraction

Text information extraction is a text information mining technology based on semi-structured or unstructured natural language texts. It is used to extract the information of users' focus and

automatically convert them into structured information from text paragraphs. The text information extraction is divided into three categories according to the characteristics of the extracted text objects, structured text information extraction, semi-structured text information extraction and unstructured text information extraction. The resume text is a typical semi-structured text that is between fully structured text and unstructured text. Semi-structured text is grouped into a collection of nouns according to the text format. It has the combination of format and freedom, and can take into account the uniformity of style and the flexibility of the content. This paper first expresses each resume as a collection of information items, each of which represents an independent semantic content and contains only one aspect of information content, each information item is represented by a noun or a combination of nouns, as shown in Figure 1.

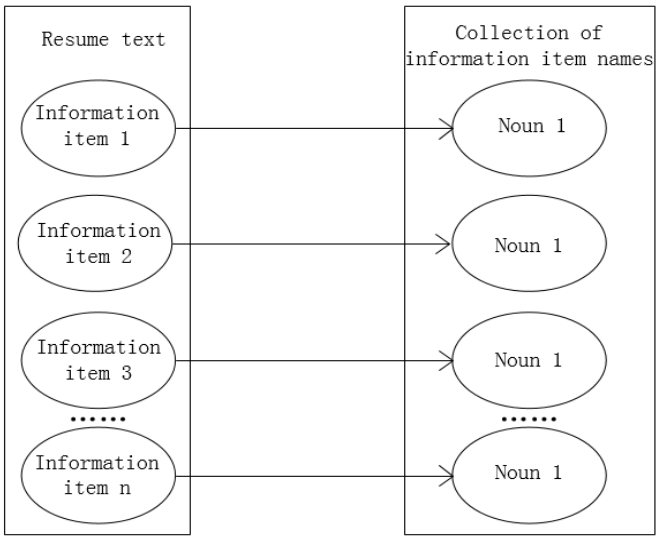

Figure 1. Schematic diagram of the information item.

2.1. Rule model

In the design of the resume extraction rule model, the target information item to be extracted should be defined firstly. It includes 25 items including name, email, and telephone. The personal information part is defined as the basic information item, and the educational background and skills are partially defined as a detail item. In fact, each resume contains some of the target information items. The establishment of the extraction rules makes full use of the semi-structured features of the resume text, considering the impact of its flexible and versatile text format on the extraction algorithm model. Through the analysis and summary of the structural characteristics of the resume, the textual structure features of the resume text include information such as the strength and weakness of the resume information, the layout features, the separator and some local features of the information items, etc.

Starting from the actual needs of the resume text extraction scene, the extracted objects include electronic documents in four formats: word, pdf, ppt, and txt. In order to facilitate the uniform format and remove the text format and layout differences within the document, the first step requires that the

various documents be uniformly formatted and finally unified into the txt text format, as shown in Figure 2. During the conversion process, the exceptions such as encryption, damage, and empty documents that exist in the document should be treated differently. After the unified formatting process, in order to facilitate the extraction, the second step is to divide the text segment twice. According to the characteristics of the resume text, the first segmentation is divided into two parts: the basic information block and the detailed information block.

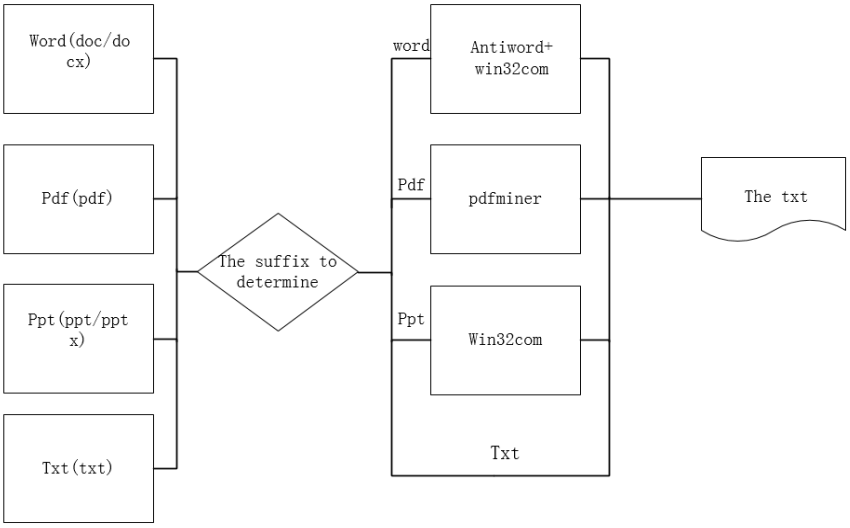

Figure 2. Document preprocessing flow chart.

Usually, the basic information item of the resume document appears at the beginning of the resume document. The schematic diagram of the segmentation process is shown in Figure 3.

The first detail item keyword that appears in the body text is the split flag of the rough segmentation. The fine segmentation is to divide the text information into short text segments in units of target information items as much as possible, and to reduce the noise influence in the information extraction phase. The basis of the fine segmentation is the layout features, separators, and the partial features of the information items that are analyzed and summarized for a large number of resume documents.

The implementation of the information extraction algorithm is analyzed and designed on the basis of the segmentation results, as shown in Figure 4. Similarly, according to the characteristic types of information items, it is divided into strong and weak identification information for matching extraction. In the actual resume text, the boundaries of different types of information are not very obvious, and there is a cross between them. Therefore, the identification and location of the information and the design of the extraction algorithm should not only be hierarchical and emphasize the role of the identifier, but also consider the positioning under the weak identification.

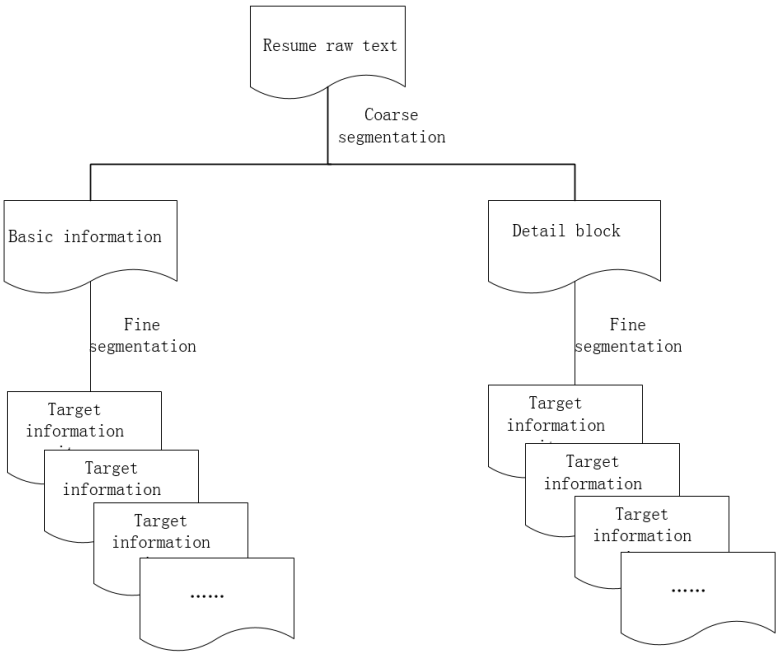

Figure 3. Curriculum vitae text segmentation.

In the design of the basic information block part of the whole extraction algorithm, the strong identification information is directly matched and extracted by the method of regular strong matching because of its obvious feature words or format, while the weak identification items are based on the method of keyword + feature matching + position location to match extraction.

The detail block usually appears in the form of a subtitle + natural text segment, so the initial positioning range of the title can be used, and then the correlation analysis is performed on the relevant text segment for final determination. The addition of the backtracking algorithm is mainly to avoid the error of the segmentation process and the omission of the matching process, which will affect the extraction of the name and other information in the resume text, and perform multiple backtracking to ensure the recall rate of the information.

2.2. Extraction algorithm

The system divided the test experiment into two parts according to the requirements of the core functions: resume information extraction experiment and massive resume data query performance experiment.

(a) Resume information extraction experiment

In this paper, we take the resume documents including word, pdf, ppt, txt.

(1) Data preprocessing

According to the design of resume information extraction process in the resume extraction system, it is necessary to uniformly format and process documents in word, pdf, ppt and other formats by judging the suffix of file names before the unified extraction and processing of the resume documents in different formats. In the end, the document is in txt format for further processing.

A. The resume document in word format is formatted into a txt document, and the content of the word document is read by antiword; the state of the document is judged from the read content, while the encrypted document is directly copied to a previously set folder, and the unencrypted document is written into txt format and stored in a specified folder for extraction processing.

B. Convert the resume documents in pdf and PPT format to txt format by using Python third-party tool win32com to read the pdf document then determines whether to encrypt the document according to the abnormal state of reading. For encrypted documents, copy them to the specified directory, and for unencrypted documents, write the contents of the documents to a txt file and save it to the specified directory for extraction processing.

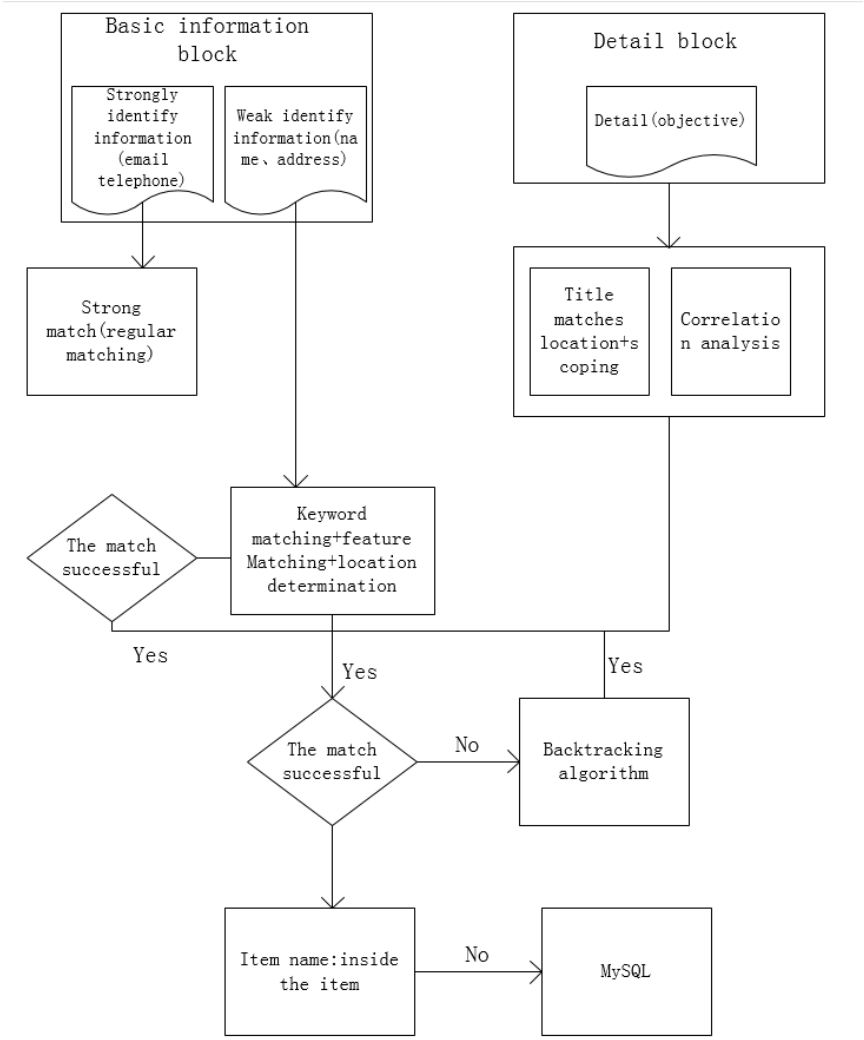

Figure 4. Extraction algorithm design diagram.

(2) Text segmentation

The implementation of segmentation of text segments is divided into the rough segmentation and the fine segmentation. The rough segmentation identification and its rules are mainly the format features and title keywords of the resume information. First of all, according to the above analysis and the segmentation flag of the basic information block and the detailed information block, the first

detailed information keyword in the body is used as the boundary. If the number of rows in which the first keyword is located is less than 10, the boundary is determined according to the location of the keyword. If it is greater than 10, the first 6 lines will be used as the basic information block, and then combined with the rules of the first six lines of the main text, the first 6 lines of the entire text are segmented as the basic information block part, and the rest is detailed information block parts. The detail segmentation stage separates the basic information block from the detail information block. The segmentation identification and rules for the fine segmentation of basic information blocks are as follows: firstly, the segmentation is performed according to the row, and then the information of each row is segmented according to symbols and features such as multiple spaces, newlines, blank lines and so on. The segmentation process of detailed information block is based on the subtitle keyword or blank line feature as the segmentation boundary. In order to ensure the performance of the extraction algorithm, the tasks in the segmentation stage are not completely performed separately, and the segmentation process and extraction stage are often carried out simultaneously.

(3) Target information item extraction algorithm

The extraction algorithm is still implemented in blocks to ensure the differentiated optimal extraction of basic information items and detailed information items.

The extraction of basic information items is mainly the extraction of four basic information items: name, email, telephone, and address. According to the analysis above, for the extraction of the name item, the first line information is positioned according to the location characteristics and judged according to the commonly used English name. In addition, some resume documents are named with the name field, and multiple information comparisons are made, and the name field is finally determined. For the extraction of mailbox and telephone, the method of advanced regular expression matching is directly adopted. The code example is as follows:

Email and telephone extract code samples

```
    pattern_email=re.compile('^0\d{2,3}\d{7,8}$|^1[358]\d{9}$|^147\d{8}|[^\._-][\w\.-]+@(?:[A-Za-z0-9]+\.)+[A-Za-z]+')
  pattern_tele=re.compile('\([0-9]{3,4}\)\s*[0-9]{3,5}-[0-9]{4,6}|[0-9]{3,5}-[0-9]{4,6}')
  email = pattern_email.search(item)
  if email is not None:
      list_email.append(email.group())
  tele=pattern_tele.search(item)
  if tele is not None:
      a=[]
      a=tele.group().split('-')
      try:
          if(int(a[1])-int(a[0])<20):
              continue
      except:
          list_tele.append(tele.group())
  str_email='    '.join(list_email)
  str_tele='    '.join(list_tele)
```

For the extraction of the address field, firstly, other information items in the basic information block are extracted and removed, and after noise and interference items are reduced, the feature information of the address information is analyzed in the remaining information: first of all, on the basis of matching the digital characteristics of the address before and after in combination with the position of the text, together with the judgment of the length of the address information, the address information is finally determined.

The extraction of detailed information items is based on the location of the segmentation boundary, relying on the title of the detailed information item for positioning and classification. In the implementation of the specific algorithm, the keywords of all the titles in the resume are matched and the corresponding positions are recorded, and then the keywords are sorted according to the line number. Then the middle part of the two words is the specific content of the corresponding keywords. After the information location is determined, the correlation between the extracted information and the existing information is analyzed, and finally the extracted information classification is determined.

(4) backtracking algorithm

In the backtracking stage, the result analysis is used to determine whether the information item to be backtracked is empty. If it is empty, the recognition and extraction of relevant information can be conducted again through method call, and the number of backtracking can be controlled by counting: to ensure the balance between the optimal extraction effect and the highest efficiency.

3. Massive CV data storage and query

In this paper, our main work is how to extract a large amount of resume text data, so the framework of the entire massive data processing system will be introduced at first.

3.1. System data storage framework

The data processing process of the massive data processing system proposed in this paper is as follows in Figure 5.

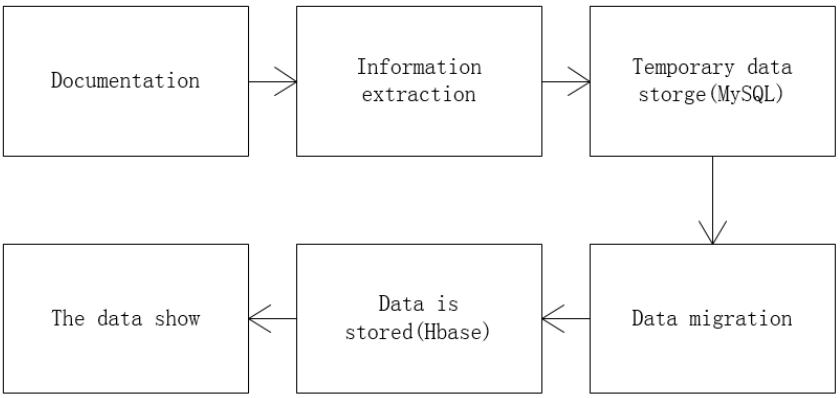

Figure 5. Schematic diagram of data flow.

In order to realize the distributed storage and access of massive CV data, for the sparse feature of the resume extraction results, the whole storage part adopts the combination of MySQL and HBase. In order to speed up the extraction, the extraction results are temporarily stored in the MySQL database. So we can use the rule model of resume information extraction to finish the translate for the CV to the text. The architecture design of the storage query part of the whole system is shown in Figure 6.

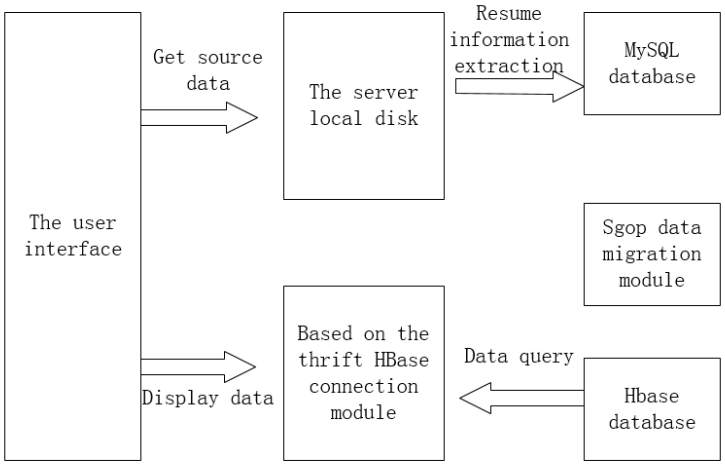

Figure 6. Resume system data storage architecture design.

With the distributed tool Sqoop, the data is finally migrated to HBase for persistent storage, and the HBase database table is designed for the characteristics of the CV data to optimize the efficient retrieval efficiency. According to the requirements analysis of system data storage and efficient query, the source file is directly stored in the server disk as a document without excessive processing. In addition, the data storage processing of the entire system is designed as four modules: MySQL storage module, Sqoop data migration module, HBase storage module, HBase connection module based on Thrift.

3.2. Pre-storage of CV data

As the original data of the system, the resume documents can be obtained via many channels, here is the mainly information such as library document, literature and so on. This paper applies the model of the resume information extraction algorithm to extract and to process the resume document, and the target item information is obtained. In order to facilitate the rapid storage of the extracted results, the storage of the extracted results is first temporarily stored in MySQL to ensure the efficiency of the extraction process and the staged storage of the data. According to the requirements analysis, the data entities included in the entire system including user entities, resume entities, historical data entities, and the like. The system entity relationship diagram is shown in Figure 7.

3.3. Migration of CV data

In order to ultimately reduce the storage, cost of the massive resume data and improve the query efficiency of the data, the data needs to be stored in HBase permanently. Therefore, the data needs to be migrated from MySQL to HBase. Data migration from MySQL to HBase involves two aspects: schema migration and data migration.

In schema migration, you first need to obtain the metadata of the database according to the specified database, and then get the markup and the metadata of each table through the original data, and focusing on the table name, column name, type information, index of the table, the information

of table's primary key and foreign key, according to the need to carry on the corresponding row keys, column family design, and then create the corresponding table in HBase. There is a foreign key information in the data table of the relational database, and there is an association relationship between the tables, but the HBase table is an independent table structure, and there is no foreign key and association relationship between the tables. So when you do schema migration from MySQL to HBase, in the face of how to convert the relationship between tables in MySQL to a table in HBase database, the only way is to store the data of relational tables in an HBase data table so that you can find it quickly and efficiently without losing efficiency. Based on the considerations mentioned above, the schema migration design for data migration is shown in Figure 8.

The specific conversion rule design is the MySQL table name as the table name of the HBase data table, and the table name is used as the column family name of the first column family of the HBase data table, and the table field corresponds to the column name of the column family. The table name of its associated table is used as the column family name of the second column family, and its table field is used as the column name of the second column family, so that the relationship between the tables can be recorded between the column families of HBase. In addition, in order to ensure the efficiency of the query, the primary key or index column of the associated table is combined as the row key of the HBase.

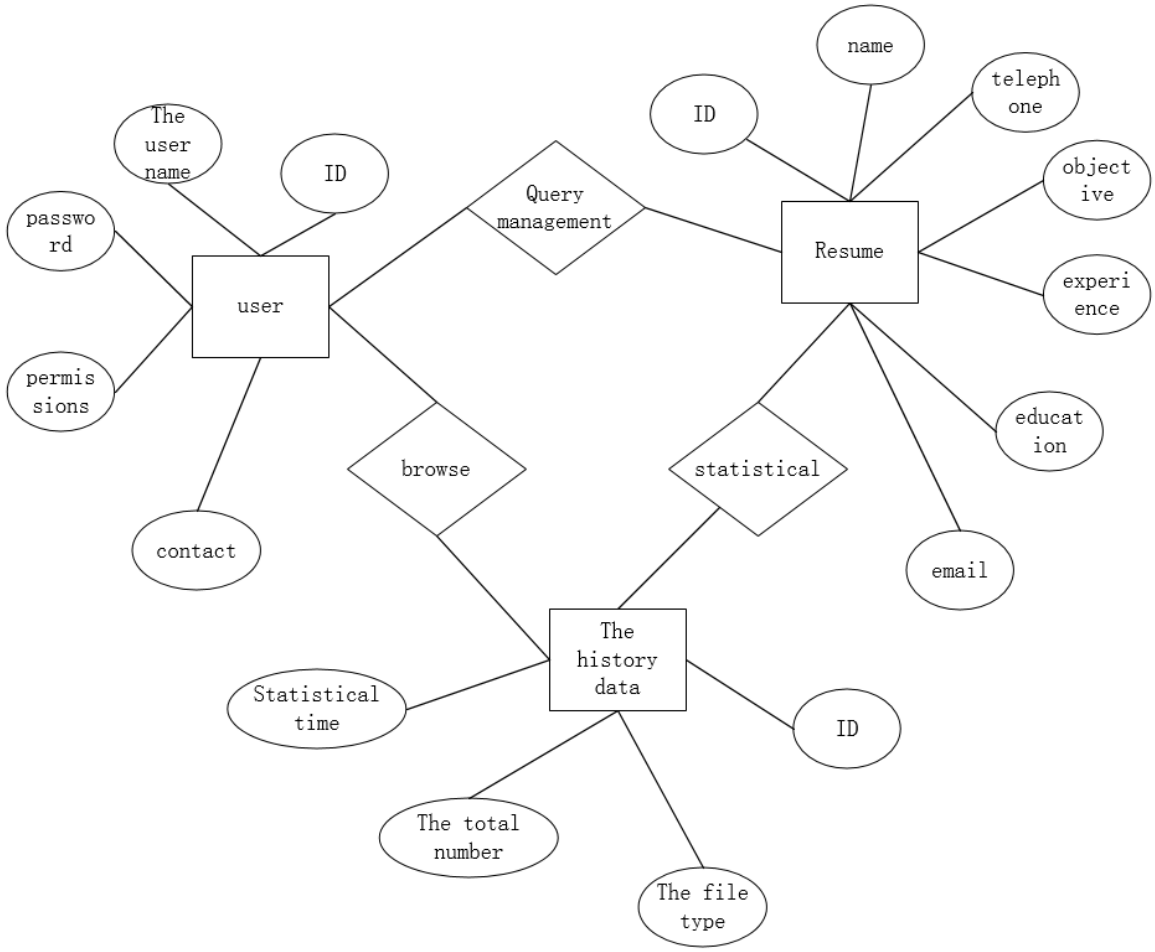

Figure 7. Entity relationship diagram.

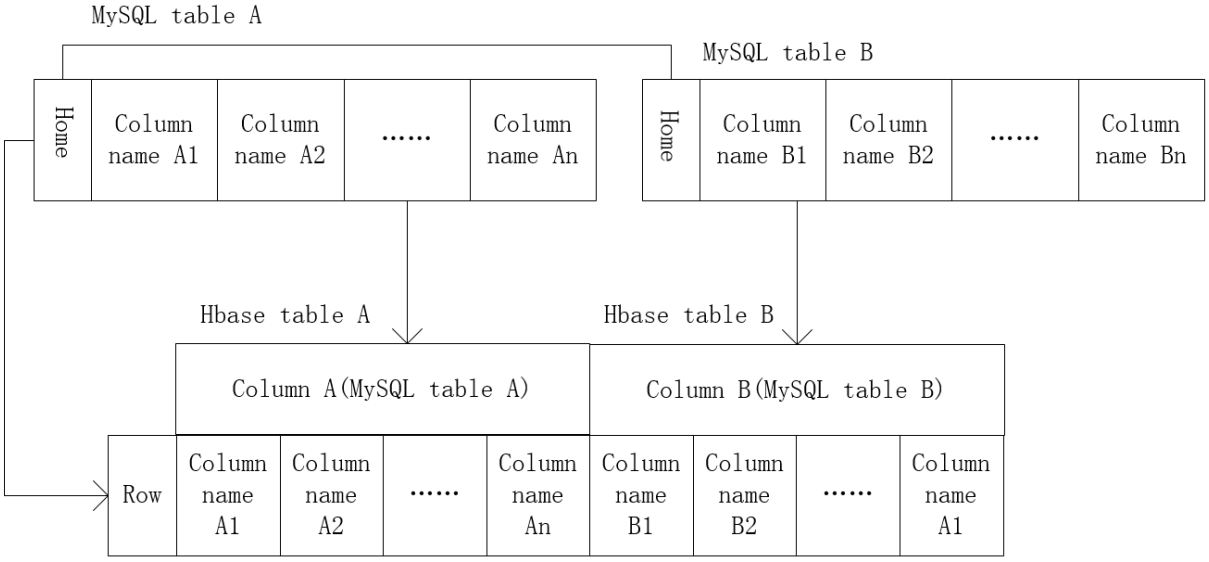

Figure 8. Schematic diagram of schema conversion.

3.4. Storage model and query optimization

The focus of the storage model design is to solve the storage structure and query performance optimization of the data table.

The rowkey field in the HBase data table is similar to the primary key of the relational database table and it is unique. Each rowkey corresponds to a data record. The distributed storage of HBase is sorted by rowkey and divided into different regions to implement distributed data storage. At the same time, HBase provides three query methods: single-point query by rowkey, range query by setting rowkey range or data retrieval that meets the requirements by scanning data table adopted by HBase without row key specified. In summary, HBase uses the class B+ tree model for storage, and the retrieval method based on the primary key is more efficient. Therefore, the design of a reasonable primary key model should fully consider the retrieval efficiency, and also consider the storage performance to avoid reading and writing hotpots.

Therefore, the storage design should be as far as possible to split the data into different regions to improve the ability of parallel queries. According to the above design ideas, the primary key design of HBase table adopts the composite primary key model with the combination of main query fields, and the identification prefix is set before the primary key to ensure the distributed storage performance. The design strategy of the primary key model in this paper is shown in Table 1. In this design strategy, the primary key is composed of the combination prefix time and the basic information field of the resume individual.

Table 1. Composite primary key design strategy.

Identification prefix	Name	Telephone	Email
2017-03	Adam L. Savery	(301) 203-0545	Savery@yahoo.com
2017-06	Chuck Martin	(719) 372-0927	Martin@google.com
2016-08	Steve G.Fishman	(443) 867-4992	Fishman@sina.com
2016-12	P. J. Wimberly	(443) 926-1340	Wim@foxmail.com
2016-01	John D.Venables	(704) 277-3164	John@google.com

4. Experiment and test

In the experiment, 3000 English resumes including word, pdf, ppt and txt formats were used. The information extraction performance of the resume information and the storage and query performance of the massive resume data were tested to verify the automation management efficiency of the resume information provided in this paper. Among them, 2,500 resumes were used as the basic data for feature collection to realize the design of the extraction algorithm. Another 500 resumes were used as test samples to evaluate resume extraction performance. For HBase-based efficient queries, a certain amount of resume data was stored as basic data in MySQL and HBase for performance comparison. The experimental steps:

①500 resumes were processed, and the number of information items in each resume was counted and marked as C_3. The total number of resumes that were unable to be processed normally due to encryption or damage were recorded as Abnormal number.

②For the test resume, the automated extraction program was used for extraction processing, the total number of information items extracted from each resume was recorded as C_2 and the number of correctly extracted information items was recorded as C_1.

③According to the formula, the accuracy rate $P = C_1 / C_2$ was calculated for each non-abnormal resume, and the recall rate was $R = C_1 / C_3$. Because there was not only one test resume, the P and R indicators were averaged separately, but there was no need to consider encryption or damage when averaging. The β value in the comprehensive index F was taken as 1 here, assuming that the recall rate and the accuracy was the same, then $F = (\beta^2 + 1)PR / (\beta^2 P + R)$ was calculated. C_3

(1) Data storage and query performance experiments

In the actual test environment of data storage and query performance experiment, the same data was stored in MySQL and distributed database HBase respectively, and then the size of MySQL data table was counted, and the size of HBase database table was checked with the same command, and finally the two were compared. In the actual environment, the query performance experiment conducted multiple multi-dimensional query experiments, statistically comparing the response time of the query request under MySQL and HBase, and evaluating the actual demand satisfaction.

(2) MySQL temporary storage

MySQL temporary storage part, mainly for resume information extraction procedures. The system information extraction program is developed in Python, so MySQL data storage access is realized by database connection pool.

The specific code is as follows:

Sample database connection pool code

```
import mysql.connector.pooling
MYSQL_HOST = 'localhost'
MYSQL_PORT = 3306
MYSQL_USER = 'root'
MYSQL_PWD = 'root'
MYSQL_DB_NAME = 'jianli'
mysql_pool = mysql.connector.pooling.MySQLConnectionPool(pool_name="mypool", pool_size=10,
                    host=MYSQL_HOST,
                    port=MYSQL_PORT,
                    database=MYSQL_DB_NAME,
                    user=MYSQL_USER,
                    password=MYSQL_PWD,
                    pool_reset_session=True)
try:
    conn = mysql_pool.get_connection()
    cursor = conn.cursor()
    sql = 'select * from model_resume'
    cursor.execute(sql)
    data = cursor.fetchall()
except Exception as e:
    print e
finally:
    cursor.close()
     conn.close()
```

(1) HBase environment setup

Base environment:Linux，Hadoop2.5.2

Installation pack：hbase-0.98.9-hadoop2-bin.tar.gz

Modify configuration file:

[hadoop@master conf]$ vim hbase-site.xml

```
<configuration>
<property>
        <name>hbase.rootdir</name>
        <value>hdfs://master:9000/hbase</value>
</property>
<property>
        <name>hbase:master</name>
        <value>hdfs://master:60000</value>
</property>
<property>
        <name>hbase.cluster.distributed</name>
        <value>true</value>
</property>
<property>
        <name>hbase.zookeeper.property.clientPort</name>
        <value>2181</value>
</property>
<property>
        <name>hbase.zookeeper.quorum</name>
```

<value>master:2181,slave1:2181,slave2:2181,slave3:2181</value>
</property>
<property>
name>hbase.zookeeper.property.dataDir</name>
<value>/home/zkpk/zookeeper-3.4.5/data</value>
</property>
</configuration>

Copy files to other nodes:

[hadoop@master local]$ scp -r hbase hadoop@slave1:/usr/local/
[hadoop@master local]$ scp -r hbase hadoop@slave2:/usr/local/
[hadoop@master local]$ scp -r hbase hadoop@slave3:/usr/local/

Start HBase on master：

[hadoop@master hbase]$ start-hbase.sh

(2) Sqoop data migration

The data migration part of Sqoop realized data migration by combining the migration command script with timed task. The code of this part of data migration script and the core code of timed task command are as follows:

Data migration script
sqoop import –connect jdbc:mysql://master:3306/resume --table model_resume --columes id,objective,experience,activities,education,summary,current,academia,awards,career_highlights, professional_affiliations,affiliations,interests,publications,qualifications,employment,language, skills,certifications,intemships,training,background –hbase-table resumes –column-family detail –hbase-row-key Rowkey --username hadoop --password hadoop --hbase-create-table

In the process of data migration, the HBase database table was created at the same time. In the data migration code, the rowkey field is used as the primary key of the HBase table. Before data migration using the Sqoop command, tables in MySQL database needs to be modified to merge the column items that are the primary keys into rowkey columns, which are then used as row keys in the HBase database table. The timed task of data migration is set at one o'clock every day, and the execution log is written into the file named as Sqoop.log. The timed task command is as follows:

Timed task command
0 1 * * * python /usr/local/sqoop.py >> /usr/local/sqoop.log 2>&1

(3) Thrift database query connection

Python uses Thrift middleware connection to access HBase, first installs Thrift under HBase directory, then installs Python's Thrift library, after successful installation, starts Thrift service on HBase cluster: bi/hbase-daemon sh start thrift, default port 9090.Upon starting the service, you first establish a connection to the HBase database's Thrift Server by:

Thrift client

```
from thrift import Thrift
from thrift.transport import TSocket, TTransport
from thrift.protocol import TBinaryProtocol
from hbase import Hbase
#server side address and port
transport = TSocket.TSocket('localhost', 9090)
#sets timeout
transport.setTimeout(5000)
#set transport
trans = TTransport.TBufferedTransport(transport)
#set transport protocol
protocol = TBinaryProtocol.TBinaryProtocol(trans)
#make sure the client
client = Hbase.Client(protocol)
#open connection
transport.open()
```

The above operation is completed, and the HBase database is accessed through the Client. On the Python website, HBase database operation request was initiated through the Client mentioned above, HBase database was accessed through Thrift plug-in, and then HBase conducted efficient data retrieval and query in a distributed environment.

4.1. Storage model and query optimization

The test environment of this paper consisted of two parts, one was the local server and the other was the distributed HBase storage system built with distributed clusters. HBase was based on Hadoop, and the experimental data was not very large. Therefore, the experimental environment was four nodes virtualized by one physical machine, including one master node and three slave nodes. The node configurations were the same, as shown in Table 2:

Table 2. Virtual Node Configuration.

Project	Configuration
CPU	Intel(R) Core(TM) i7-5500U
memory	1G
Hard disk	30G
system	Centos6.5
JDK	1.7
Hadoop version	2.5.2
Development of language	Java

A single physical machine configuration is shown in Table 3:

4.2. Storage model and query optimization

Combined with the core algorithm for implementation and testing, the test system architecture was implemented as shown in Figure 9.

The system overview was mainly divided into three parts. The first part showed the number of system resumes in the form of charts and numbers according to the document format category. The

second part showed the historical operation data of the resume document processing in the form of a graph. By default, the data processing history in one year was displayed in units of months. In the upper right corner, a drop-down list was provided, allowing the user to view the history of resume processing in days for a month. The third part statistics on the proportion of resume documents in various formats.

Table 3. Physical Machine Configuration.

Project	
Processor	Intel(R) Core(TM) i7-5500U CPU
Memory capacity	8.00G
Graphics card	AMD Radeon R7 M260
Hard disk	ST500LM021-1KJ152 (500GB)
Motherboard	20DCA01PCD (SDK0E50518 STD)
Network card	Intel(R) Ethernet Connection (3) I218-V
Sound card	Conexant SmartAudio HD
Monitor	LEN:a640 Resolution:1920x1080
Current os	Windows 10 64 位

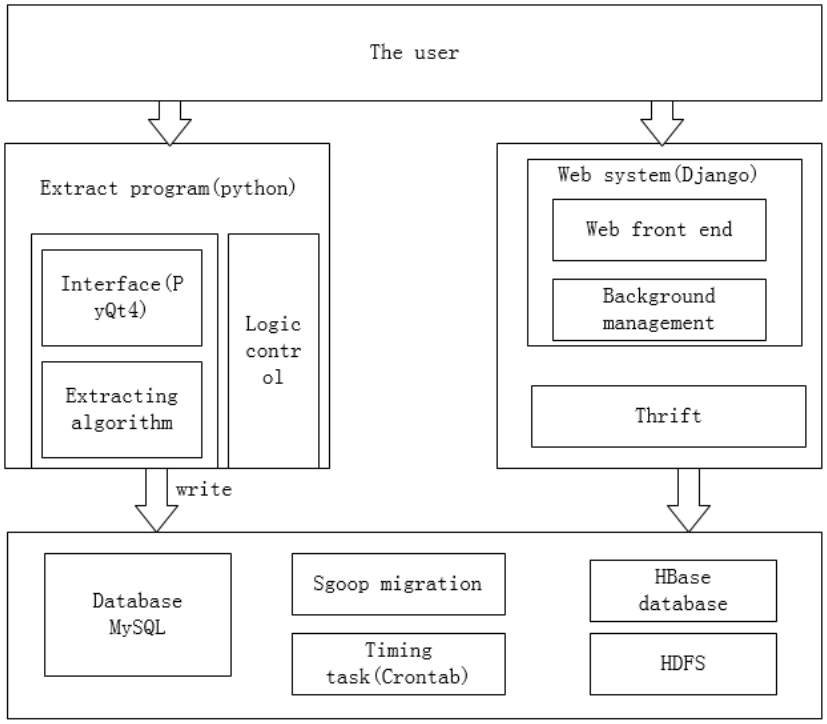

Figure 9. Technical architecture diagram.

The system overview is shown in Figure 10. We can see the system has processed 210,000 resumes, mainly including word and pdf files.

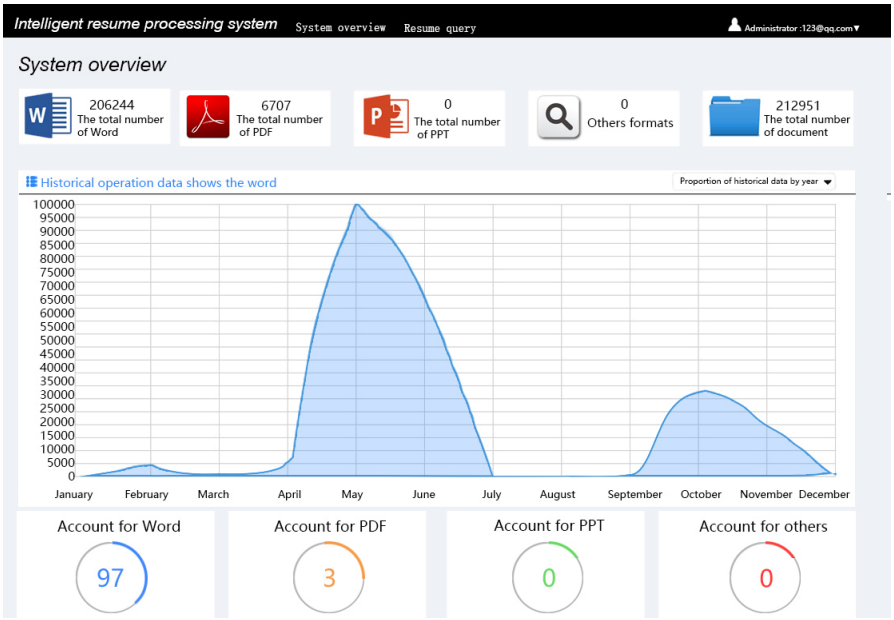

Figure 10. System overview.

4.3. Evaluation criteria and experimental results

(1) Evaluation criteria

This system mainly provides automatic information extraction service and web-based resume information retrieval management for massive resume information. Because the accumulated data level reached tens of millions or even billions and requires the liberation of manpower to achieve automated information extraction. Therefore, the effect of information extraction of the system and the retrieval performance of massive data have a greater impact on user experience. The specific evaluation criteria are as follows:

① Evaluation criteria for the performance of resume information extraction, using the accuracy rate (P), recall rate (R) and comprehensive index (F) mentioned above.

② In the big data environment, the HBase-based distributed storage system query storage performance evaluation criteria are the disk space occupied by the large data volume and the query response time. The disk footprint compares the disk space in the traditional relational database MySQL and the distributed non-relational database HBase with the same amount of data, and evaluates the storage advantages of distributed HBase. The query response time is then queried to test the specific response time and whether the actual demand is met.

(2) Experimental results

① In this experiment, some information and comprehensive indexes in resume documents were counted. 500 resumes were tested, including 10 encrypted or damaged resumes, and 490 actual information extraction resumes. The experimental results of information extraction are shown in Table 4.

The experimental results show that the extraction performance of email and telephone is relatively better, and the accuracy and recall rate are higher. This is because the two kinds of data belong to strong identification information, and the features are easier to identify than other data. The other parts of the data are relatively low, but overall, the comprehensive accuracy and recall rate can

reach more than 86%, and the comprehensive index F also reaches 88.06%, indicating that the extraction model is more effective. Due to the diversity of the resume documents, there are out-of-order situations and picture information in the unified formatted document, which leads to some situations in which the current extraction model still cannot meet the requirements. This needs to be further researched and improved in the future work. In terms of information extraction efficiency, the extraction program has reached a speed of more than 600 extractions per minute and has reached the expected standard.

Table 4. Information extraction experiment results.

	Accuracy rate (P) /%	Recall rate (R)/%	Comprehensive index (F)
Name	94.41	94.89	94.64
Telephone	98.20	91.02	91.99
Email	98.22	91.84	94.95
Address	92.33	87.75	89.98
Others	86.76	89.39	88.06

② In the database storage query performance test, the resume data is stored in MySQL and HBase respectively, and the space size comparison is shown in Figure 11.

As can be seen from the figure, the space for extracting the same amount of resume data into MySQL is significantly larger than HBase, and as the number of resumes increases, the space occupied by HBase storage grows more slowly, so HBase is better than MySQL for sparse resume data storage. Through the query access test, the HBase-based query time delay is basically stable within 1 second, which satisfies the actual query requirements. Through this experiment, we can see that the use of HBase to store large amounts of resume data query to a certain extent has improved the economics of the system and met the requirement of the actual application.

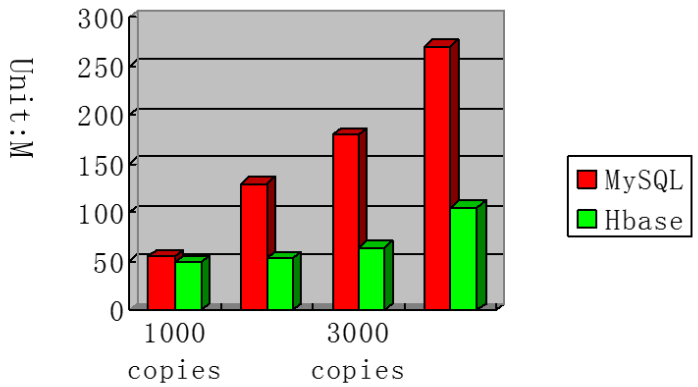

Figure 11. Comparison between MySQL and Hbase storage.

5. Conclusion

In the context of massive information query and intelligent analysis in social large-scale

complex systems, this paper studies the semi-structured features of resume documents and designs the resume extraction rules model. With the help of the distributed database HBase, the data migration is studied; the migration mode and the HBase database table model are designed. Thereby ensuring the accuracy and recall rate of automatic information extraction analysis, and improving the economics of mass data storage and the efficiency of query management. Finally, the experimental results show that the new system architecture has better performance in resume data management and system analysis.

Acknowledgement

This work is partially supported by Science and Technology Project Plan of Heilongjiang Archives Bureau "Big data archives storage and retrieval method" (HDK2018-20), and the National Natural Science Foundation of China (31770768, 31370565).

Conflict of interest

No author of this paper has a conflict of interest.

References

1. B. Li, Y. Chen and S. Yu, Review of information extraction research, *Comput. Eng. Appl.*, **10** (2003), 1–5+66. (in Chinese)
2. Y. Liu, R. Jin and J. Y. Chai, et al., A Maximum coherence model for dictionary-based cross-language information retrieval. Proceedings of the 28th Annual International ACMSIGIR Conference; 2005 August 15–19; Salvador, Brazil. New York: ACM; 536–543.
3. A. L. Berger, V. J. D. Pietra and S. A. D. Pietra, A maximum entropy approach to natural language processing, *Comput. Linguist.*, **22** (1996), 39–71.
4. W. Huang and Y. Sun, Chinese short text sentiment analysis based on maximum entropy, *Comput. Eng. Des.*, **38** (2017), 138–143. (in Chinese)
5. Y. Lin, Y. Liu and S. Zhou, Text information extraction based on maximum entropy of hidden Markov model, *Acta Electronica Sinic*, **33** (2005), 236–240. (in Chinese)
6. K. Seymore, A. Mccallum and R. Rosenfeld, Learning hidden Markov model structure for information extraction, *In Aaai'99 Workshop Machine Learning for Information Extraction*, (1999), 37–42.
7. C. Chi and Y. Zhang, Information extraction from chinese papers based on hidden Markov model, *Adv. Mater. Res.*, **846** (2014), 1291–1294.
8. Y. Liu, Y. Lin and Z. Chen, Text information extraction based on hidden Markov model, *J. Syst. Simulat.*, **16** (2004), 507–510. (in Chinese)
9. S. Zhe, *Research and application of hidden Markov model in web page information extraction*, Ph.D thesis, East China Normal University, 2016. (in Chinese)
10. S. Zhou, Y. Lin and Y. Wang, et al., Text information extraction based on clustered hidden Markov model, *J. Syst. Simulat.*, **19** (2007), 4926–4931.
11. Q. Du, H. Wang and Z. Shao, et al., Research on the extraction method of literature metadata based on hybrid HMM, *Comput. and Digit. Eng.*, **45** (2017), 101–106. (in Chinese)

12. F. Ciravegna and A. Lavelli, Learning Pinocchio: adaptive information extraction for real world applications, *J. Nat. Lang. Eng.*, **10** (2004), 145–165.
13. W. Yu, G. Guan and M. Zhou, et al., CV information extraction based on two-level cascade text classification, *J. Chinese Inform. Process.* **20** (2006), 59–66.
14. K. Yu, G. Guan and M. Zhou, Resume information extraction with Cascaded Hybrid Model. Proceddings of the 43th Annual Meeting of the ACL; 2005 June; Ann Arbor, Michigan. Association for Computational Linguistics; 499–506. (in Chinese)
15. Q. Wang and F. Li, Wikipedia-based resume extraction of personal name information, *Comput. Appl. Softw.*, **28** (2011), 170–174. (in Chinese)
16. N. Ren, *Research on the extraction of character title information in large-scale real texts*, Ph.D thesis, Beijing Language and Culture University, 2008.
17. N. Gu, W. Feng and X. Sun, et al., Chinese resume automatic analysis and recommendation algorithm, *Comput. Eng. Appl.*, **53** (2017), 141–148+270. (in Chinese)

AIMS Press