Design and Implementation of Analyzer Management System Based on Elasticsearch

Jingqi Sun, Peng Nie, Licheng Xu and Haiwei Zhang

October 17, 2021

Design and Implementation of Analyzer Management System Based on Elasticsearch

Jingqi Sun[1,3], Peng Nie[1,3(✉)], Lingcheng Xu[2,3], and Haiwei Zhang[1,3]

[1] College of Cyber Science, Nankai University, Tianjin 300350, China
sunjingqi@dbis.nankai.edu.cn
{niepeng, zhhaiwei}@nankai.edu.cn
[2] College of Computer Science, Nankai University, Tianjin 300350, China
xulicheng@dbis.nankai.edu.cn
[3] Tianjin Key Laboratory of Network and Data Security Technology

Abstract. Elasticsearch is a distributed search and analytics engine for all types of data, and its analyzer plays a crucial role in the construction of data index. An appropriate analyzer can significantly improve the accuracy of the index. However, for some languages, especially Asian languages, the default analyzer of Elasticsearch is not enough to fulfill the retrieval requirements, so it is necessary to configure the analyzer and thesaurus manually. Therefore, it is inefficient and means the waste of human resource. In this paper, we design and implement an automatic configuration management system for the analyzer in Elasticsearch, which provides a visual interface, enables users to select analyzer individually, install and configure them automatically. In addition, we integrate Elasticsearch's Restful API into the system to better manage the cluster. The experiments show that The system can simplify the configuration process of the analyzer to a great extent and significantly improve the development efficiency.

Keywords: Analyzer configuration · Elasticsearch management · Information management system.

1 Introduction

With the development of Internet technology, information retrieval technology is constantly evolving. As a lightweight full-text search engine with complete community ecology, Elasticsearch has been widely used in various enterprises. However, when dealing with Asian languages, such as Chinese, Elasticsearch does not provide a default analyzer but adopts the maximum granularity segmentation scheme, which leads to the decline of query accuracy. In order to solve this problem, an Analyzer for Chinese is usually used to segment Chinese document instead of using the default configuration. After word segmentation, the semantic of the text can be considered to improve the accuracy of the query. However, Elasticsearch has only backend service but no visual operation interface, which has many inconveniences in operation and use. Firstly, It needs to send HTTP requests to the cluster to configure the Chinese analyzer, which is cumbersome and reduces development efficiency. Moreover, the system lacks login authentication and permission management mechanism like MySQL database, so the security of the system is low, which is easy to cause data leakage. Finally, it requires a large number of native requests to test the interface and manage the index. Therefore, it is very cumbersome, resulting in low development efficiency.

In order to solve this problem, we propose and implement an analyzer management platform based on Elasticsearch. The system can help users quickly configure and test the word segmentation and realize the automatic update of the analyzer and the automatic reconstruction of the index without downtime. At the same time, the system also provides the management function of the Elasticsearch cluster, which can quickly obtain the real-time status of the cluster, dynamically change the configuration of the cluster, and help developers better manage the cluster environment. Finally, the running results of the system show that the system can run normally and can help users to configure and manage word segmentation quickly and conveniently.

2 Related Work

An apparent difference between Elasticsearch and traditional relational databases is that the former is an unstructured NoSQL database, so many concepts such as indexes, analyzers, documents, etc. are not intuitive enough for us, while Elasticsearch's visualization tools solved this problem. It can intuitively display the structure and content of various data, which is convenient for our understanding. At present, the mainstream cluster management tools include Elasticsearch-Head, Dejavu, Kibana, etc. Elasticsearch-Head is a plugin used to manage Elasticsearch. It obtains data from the cluster and displays it through the HTTP RestfulAPI provided by Elasticsearch, but it has simple functions and a simple interface. Kibana is an officially provided data visualization tool. It can visualize vast amounts of data in Elasticsearch using charts and graphs. It often appears in big data analysis scenarios. However, Kibana does not have many functions in cluster management, index management, etc., which is challenging to meet the daily development needs. Dejavu is the missing web UI for Elasticsearch. It has made many optimizations for the visualization of cluster data and supports online preview and update of data, as well as data import and export. But the feature of Dejavu is quite simple for it only provides the feature of data operation, and can only operate the data of one index at a time.

In summary, the above tools are still basic cluster management tools. For some specific features, such as the configuration of the analyzer described in this article, there is no good support, and you need to follow the original steps base on Elasticsearch. At the same time, these tools rely on Elasticsearch's HTTP service, so the relevant permissions must be opened before they can be used, which may lead to some security risks.

3 Concept of Design

3.1 Overview of System

The system provides a function to configure the analyzer automatically. The target group of the system is developers, for it can significantly improve the development efficiency. After logging into the system, the user can select an appropriate analyzer and submit it. The system will automatically generate the analyzer configuration and upload it to the Elasticsearch cluster. Then, rebuild the index of the relevant document and ensure that the system is always available during the rebuilding process.

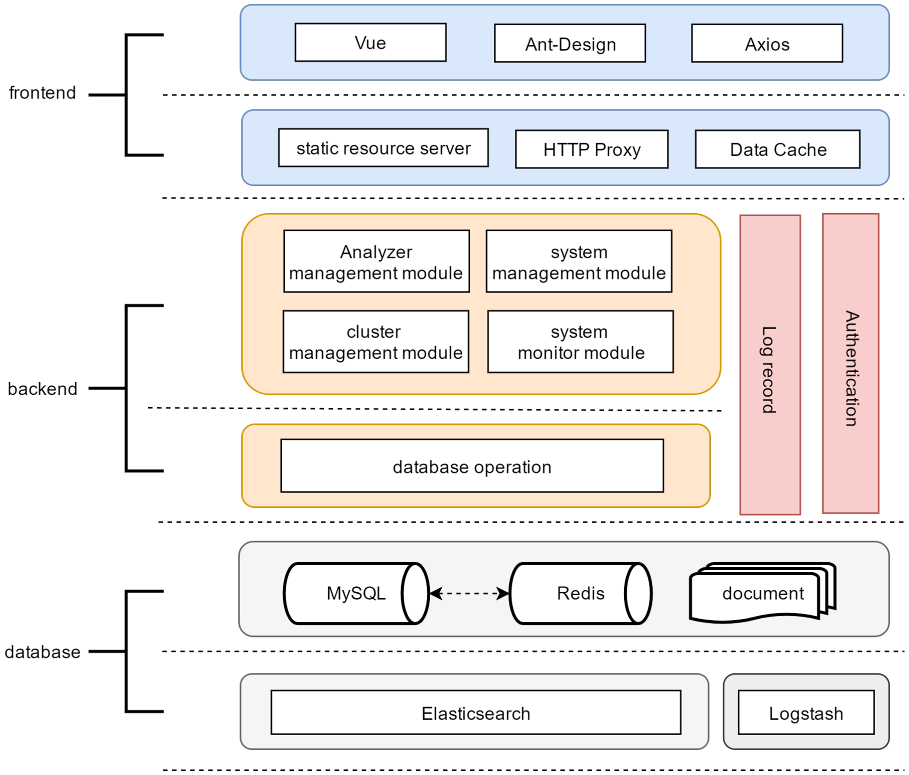


Fig. 1. the design of system logical architecture.

The system adopts the classic B/S architecture. The most significant advantage of B/S architecture is convenience and cross-platform. Users can use the system as long as they have installed a browser. From the development perspective, once the development is completed, it can run on anywhere, which significantly reduces the development cost. The architecture design of the system adopts hierarchical model(see Fig. 1). In hierarchical model, each layer performs its duties. The upper layer depends on the lower layer, and the upper layer does not need to care about the implementation of the lower layer. It only needs to call the interface provided by it to decouple the system functions and improve the maintainability of the system.

3.2 System Design

This system can be divided into four modules: analyzer management module, cluster management module, system monitor module, system management module(see fig2). The analyzer management module is the core function module of the system, which completes the selection, installation and configuration process of the analyzer in Elasticsearch. The system management module plays a significant role in the system, mainly accomplishing the functions of authentication, user management to guarantee the security of the system. The remaining modules are mainly auxiliary to the operation of the system. The cluster management module and the system monitoring module are responsible for monitoring the Elasticsearch cluster environment, monitoring the system's running status and system operation environment.

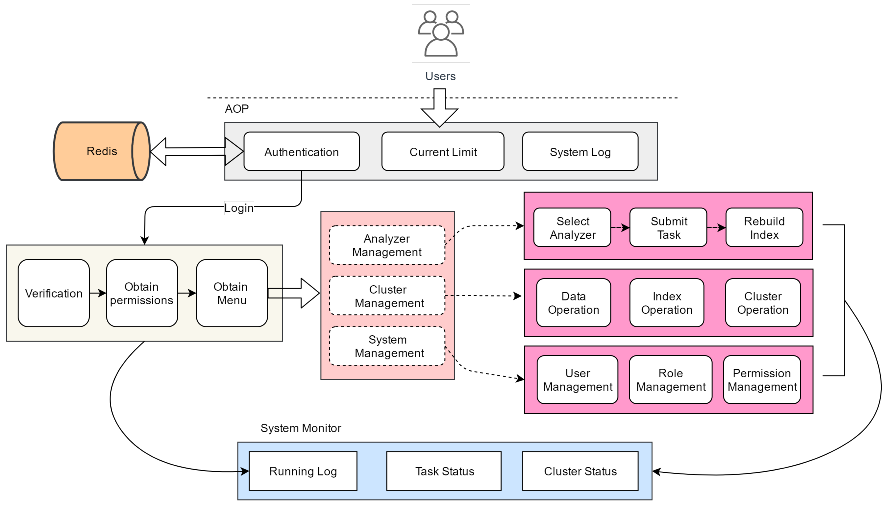


Fig. 2. the design of system logical architecture.

Analyzer Management Module The analyzer management module supports automatically install an analyzer or generate an analyzer based on a custom configuration. The system can automatically install analyzers include analyzer plugins currently certified by Elasticsearch officially. Users can apply analyzers to clusters by selecting, downloading, and installing them directly online. User-defined analyzers are also supported, and configurations are generated to join the cluster. To achieve the automatic configuration process, we used an asynchronous task processing method (see fig. 3).

After the user selects the analyzer and confirms, an asynchronous task is automatically generated, and the task information is submitted to the asynchronous task manager and saved in a queue. At the same time, the parameter information of the task is communicated to the task executor, which initiates the specific task process. Although there are some differences between the two methods, they are generally the same in the configuration process. We abstracted the entire process into four steps: downloading the analyzer, unpacking the package, installing it in the Elasticsearch cluster, and finally rebuilding the index for the changes to take effect.

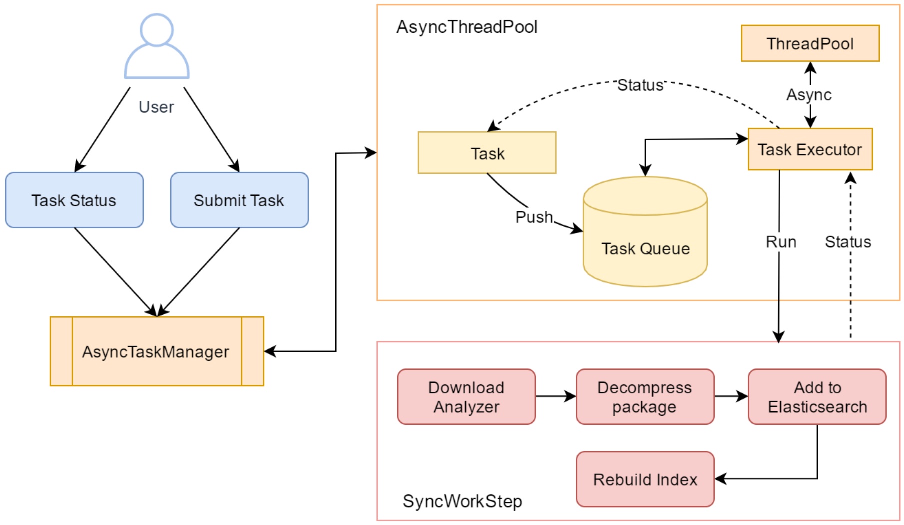


Fig. 3. Asynchronous Task Management Model

After installation, users often configure the analyzer's dictionary, and so on. We provide an online dictionary configuration for IK-Analyzer. Users can directly visualize the operation dictionary to improve the accuracy of the query.

Cluster Management Module Automation can greatly affect the user experience. Therefore, we want to provide as many automated processes as possible so that users can have a better experience using the system. The cluster management module is for this purpose. This module can be divided into three sub-modules: index management, data operation, cluster monitoring. The index management module supports index operations of the Elasticsearch, such as index deletion, index creation and other functions. The data operation module provides a visible interface for user to catch the data in Elasticsearch.

In a real application scenario, when configuration updates, normal services should also be provided to avoid problems while using the system. The easiest way to solve this problem is to set a timed task, such as updating the configuration of the cluster in the middle of the night. However, this can cause data lag. Because configuration updates often do not take effect immediately. To avoid these problems, we designed a non-downtime index rebuilding method(see fig.4). After the user initiates an index rebuild request, we first alias the source index and generate a new index that uses the updated analyzer. When the new index is built, migrate the data from the old index through the reindex API. Then alias the new index, and delete the old. The entire process is not visible to the user, and the way the user's query requests are handled remains the same, that is, to access the data through aliases.

System Management Module The system management module can be divided into two sub-modules, the authentication module and the permissions management module. The authentication module allows users to sign in to the system through their accounts and authorizes users of the system. At the same time, to prevent the same account from being reused, the system restricts users from using the system online at the same time. After the user logs into the system, query the database for the appropriate permissions and display pages that match the user's permissions. The logon status is cached in the in-memory database Redis so that when the user closes the browser, the session will not fail and will be saved in Redis for some time. Users can directly enter the system without having to sign in again to improve their experience.

The permission management section uses the RBAC(Role-Based Access Control) model, where the user-role-permission relationship is many-to-many, each user has multiple roles, each role has multiple permissions, and the permissions are the sum of all the permissions of all its roles. Moreover, the permissions are mainly reflected in data access, page access and operation. All operations in the system are for authorized users.

Monitor Module In order to make the system run stably for a long time, we realize a monitor module for the system. The monitor module is invisible for users but very important for the system. We provide tow main function in this module. They are log record and environment monitor. Logging is very important for any information system. When there is an exception to the system, we can find problems in the log in time. At the same time, we can record the action of users using the system, from what we can find much valuable information. In our system, we mainly record the sign info and the operation log of users. These are benefit for the maintenance and management of the system.

The environment monitor is for the developers. They can quickly obtain the parameters of the runnable environment. It provides a way to observe the lower

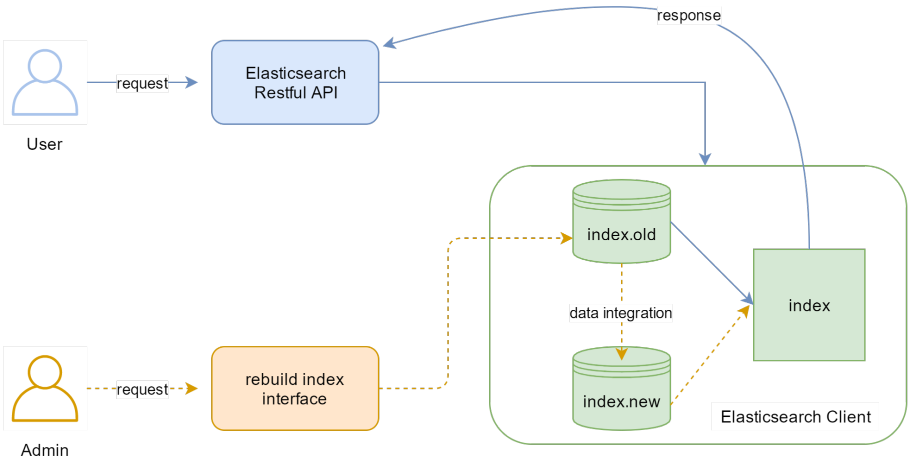


Fig. 4. the process of rebuilding the index

environment from above services. We realize it based on Spring-Boot-Actuator, which provides a series API to get the running status and environmental parameters of the current system.

4 System implementation

In the previous section, we focus on the architecture and design of the system. The content of this section mainly contains three aspects: 1.Platform development environment: IDE (Integrated Development Environment) used for system development, development environment and its configuration, etc. 2. The implementation of Analyzer Management Module: the module is the core of the system. 3. The final effect.

4.1 Development Environment

The system adopts the development method of separation of front and back ends. The front end is developed using Vue+Less+axios+nginx. Among them, the most important role is Node.js. Node.js's package manager NPM can help us quickly download and install the expansion packages and configuration items needed for front-end development.

The back end uses the technology stack of Java+Spring Boot+MySQL+Redis+ Elasticsearch. The back-end development environment needs to install the necessary environment. For example, Java needs the support of JDK (Java Development Toolkit), and Spring Boot needs Maven and Spring CLI to complete package management and project creation. The specific development environment and development tools are shown in Table 1.

Table 1. Development Environment and Tools.

Item	Content
Operation System	Windwos 10
Language	Java, Vue, Less, JavaScript, Painless
Development tools	IntelliJ IDEA, WebStorm
DataBase	MySQL5.7, Redis
Middleware	Elasticsearch 6.6.2
Web Server	Apache Tomcat8.0, Nginx
Package Management	Apache Maven 3.5.2

4.2 Implementation

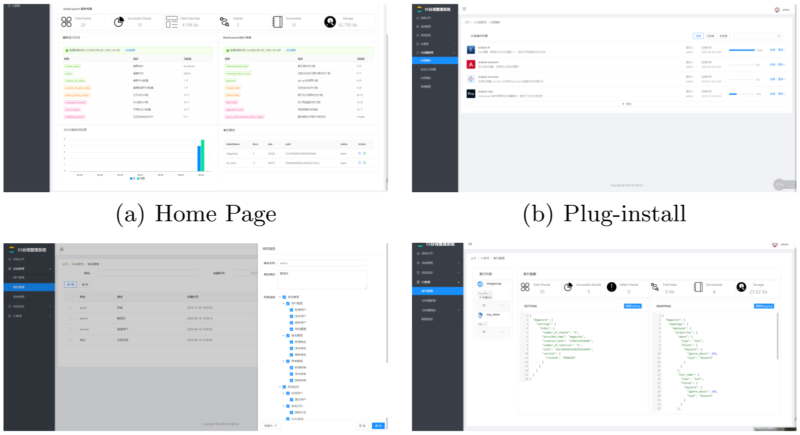
According to the system design, the analyzer management module is the core of the system. It mainly consists of four parts: plug-install, custom analyzer, analyzer testing, and dictionary configuration. This module does not need to use a database for data storage but interacts directly with the cluster through the Java API, which Elasticsearch provides.

Plug-install Plug-install feature provides the ability to install the online ananlyzer plug-ins to users. At present, the system supports analysis-ik, analysis-synonym, analysis-angj, analysis-kuromoji and other tokenizers. Users can directly search for word segmentation plugins online and download them. The system will automatically install the plug-in to the Elasticsearch plug-in directory and perform automatic configuration.

Custom Analyzer The custom analyzer function supports users to configure personalized tokenizers. In Elasticsearch, the analysis process will actually go through three steps: character filtering (char_filter), tokenize (tokenizer), and token filtering (token_filter). In the character filtering stage, specific characters in the text will be removed, such as tags in HTML, XML and other texts; in the tokenize stage, the text after the filtered characters will be segmented, and different tokenizers can be selected during tokenizing. The standard tokenizer will be used by default ; At the token filtering stage, the token generated is processed again, such as stop words, synonyms, etc. This page provides a form, users can complete the content of the form, configure the corresponding parameters, and finally generate a complete Json sentence of the custom tokenizer.

Analyzer Testing and Dictionary Configuration In order to test the effect of the analyzer, the system implements the analyzer testing module. The function of this module is relatively simple. The page is divided into upper and lower parts, with input boxes and test buttons at the top. Users can select the index and the analyzer in the index, enter the text, and then click the "test" button to obtain the text information after segmentation.

The screenshots of system operation are shown below(see fig5).



(a) Home Page

(b) Plug-install

(c) System Management

(d) Cluster Management

Fig. 5. The Implementation of the system

4.3 Final Effect

We implement the system according to the specific design scheme of the previous chapter. Through a period of application, we find that the introduction and implementation of this system has a significant effect on improving the development efficiency, optimizing the management of the Elasticsearch cluster, and improving the security of the Elasticsearch cluster. This is mainly reflected in the following aspects:

1. The system allows users to install, configure, and test analyzers directly on the page. If one does not use the system, he needs to re-index each time configure it for the analyzer to take effect. In situations with large amounts of data, this process is very inefficient.

2. The cluster management module provided by the system allows users to monitor indexes in the Elasticsearch cluster and perform some everyday operations.

3. Relative to some of the popular Elasticsearch management tools, the most significant feature of our system is that we no longer use the HTTP port of Elasticsearch but use the Java API to implement various related functions. Java API ports can be shielded entirely in the intranet environment, which maximizes data security without causing data leakage problems.

Table 2. Comparison with other tools

Features	Our System	dejavu	ES-head	Kibana
Port	9300	9200	9200	9200
Modern UI	Ant-Design V2.1	React 16.6	JQuery 1.6.1, slightly stodgy	Node.JS, Hapi, Jade
Browser features	CRUD, data filters, full-text search	CRUD, data filters	Read Data, full-text search	Read View, visualizations, charting
Data Import/export	support for JSON	support for JSON,CSV	No	Only Export
Analyzer Installation	support auto installation	No	No	No
Custom Configuration for Analyzer	Visually build and test	No	No	No
Authority system	Yes	No	No	Need X-Path

5 Summary

Elasticsearch is gaining more and more recognition in the industry in terms of performance and convenience. Systems using Elasticsearch as a search engine are also increasing. However, Elasticsearch's support for Asian languages, such as Chinese, is not perfect. A specific analyzer is needed to improve the accuracy of queries. However, it only provides only background services rather than a visual application interface, making the configuration process tedious. In addition, it also appears cumbersome in index management, data operation, interface testing and other functions.

To solve these difficulties, this paper designs and implements the Elasticsearch analyzer management system. The system is divided into four modules: the system management module and the analyzer management module responsible for the user and permission management of the system and the analyzer configuration. Two auxiliary modules, the system monitoring module and the cluster management module, are responsible for monitoring the system operation and operating of the Elasticsearch cluster.

By designing and implementing this system, developers can quickly manipulate the underlying APIs in the Elasticsearch cluster, create indexes, view index status, modify index configuration, and so on. At the same time, for Asian languages, developers can directly select and install analyzer in a clustered environment through page operation.

6 Future Work

This paper designs and implements a system for analyzer management based on Elasticsearch. Using this system, users can directly operate the Elasticsearch cluster, configure indexes, view data, and configure the analyzer online, facilitating the development process. However, due to limited time, there are still some deficiencies in the system that need improvement. In the future, the system can be improved from the following aspects:

Firstly, for the analyzer management module, only some Chinese analyzers, such as IK-Analyzer, can be supported in the system at present, while others are not compatible. If the user expects to use another language, it cannot be installed and configured through the system.

Secondly, the permission scheme is still not detailed enough. At present, if one has the permission of data operation, he can get all types of data in the system. This coarse-grained permission scheme puts the data in the system at risk.

In future research work, we will further improve the system based on the above deficiencies and improve the system's applicability.

References

1. A Kanagasundaram, D Dean, S Sridharan, C Fookes. DNN based Speaker Recognition on Short Utterances. Speaker Language Recognition Workshop, 2016.

2. A Graves. Long Short Term Memory. Springer Berlin Heidelberg, 2012, 9 (8): 1735-1780.
3. Z Huang, J Tang, S Xue, el at. Speaker adaptation OF RNN-BLSTM for speech recognition based on speaker code. IEEE International Conference on Acoustics, 2016: 5305-5309
4. D Wang, X Zhang. THCHS-30 : A Free Chinese Speech Corpus. Computer Science, 2015.
5. C Li, X Ma, B Jiang, el at. Deep Speaker: an End-to-End Neural Speaker Embedding System. Computation and Language,2015
6. Praveen M Dhulavvagol,Vijayakumar H Bhajantri,S G Totad. Performance Analysis of Distributed Processing System using Shard Selection Techniques on Elasticsearch[J]. Procedia Computer Science,2020,167.
7. DA Reynolds. An overview of automatic speaker recognition technology. IEEE International Conference on Acoustics, 2011, 4: IV-4072-IV-4075.
8. S Bagnasco,D Berzano,A Guarise,S Lusso,M Masera,S Vallero. Monitoring of IaaS and scientific applications on the Cloud using the Elasticsearch ecosystem[J]. Journal of Physics: Conference Series,2015,608(1).
9. L Muda, M Begam, I Elamvazuthi. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. Ttps, 2010.
10. Towards user-oriented RBAC model[J] . Haibing Lu,Yuan Hong,Yanjiang Yang,Lian Duan,Nazia Badar. Journal of Computer Security . 2015
11. A Graves, AR Mohamed, G Hinton. Speech recognition with deep recurrent neural networks. IEEE International Conference on Acoustics, 2013, 38 (2003): 6645-6649
12. Zhou Shiying. Research and Implementation of Chinese Word Segmentation Method Based on Maximum Probability[A]. Institute of Management Science and Industrial Engineering.Proceedings of 2019 8th International Conference on Advanced Materials and Computer Science(ICAMCS 2019)[C].Institute of Management Science and Industrial Engineering:Computer Science and Electronic Technology International Society,2019:5.
13. V Tiwari. MFCC and its applications in speaker recognition. International Journal on Emerging Technologies Issn , 2010.
14. W Han, CF Chan, CS Choy, el at. An efficient MFCC extraction method in speech recognition. IEEE International Symposium on Circuits Systems, 2006: 4 pp.
15. Yadanar Oo,Khin Mar Soe. Joint Word Segmentation and Stemming with Neural Sequence Labeling for Myanmar Language[A]. SCIence and Engineering Institute (SCIEI)University of Computer Studies, Yangon.Proceedings of 2019 the 11th International Conference on Future Computer and Communication (ICFCC 2019)[C].SCIence and Engineering Institute (SCIEI)University of Computer Studies, Yangon:SCIence and Engineering Institute(SCIEI),2019:6.
16. WM Campbell, DE Sturim, DA Reynolds. Support Vector Machines using GMM Supervectors for Speaker Verification. IEEE Signal Processing Letters, 2006 , 13 (5): 308-311.
17. Y Xu, I Mcloughlin, Y Song, el at. Improved i-vector representation for speaker diarization. Circuits Systems Signal Processing, 2016 , 35 (9): 3393-3404.
18. Lei Lei, She Kun. Speaker Recognition Using Wavelet Packet Entropy, I-Vector, and Cosine Distance Scoring. Hindawi Publishing Corp, 2016, 2016: 1-11.