Design and Implementation of Personalized Learning Platform Based on Big Data

Xueming Bai

School of Information Science and Technology, Taishan University, Taian 271021, China

Keywords: Personalized learning, Knowledge graph, Search engine, Big data

Abstract: The paper designs and implements a personalized learning platform for search. Based on big data technology, users search keywords on the platform which excavates and filters information from the huge cloud resources and pushes it to users. Through the mining of related web data, valuable information such as users' behavior, habits, preferences, etc. can be provided. The platform then integrates and reverses relevant recommendations for users to achieve private determination. The crawler system uses the scrapy crawler framework, the search engine service uses the Elasticsearch framework, and the data collection engine runs in the following environment: cloud server, scrapyd official deployment tool, and scrapydweb visual crawler monitoring framework. The website platform is based on the Python Django framework development. The project is based on uWSGI and Nginx as the production environment for the project, and built on the Docker container's service environment. The effect of the system is to realize the learning of artificial intelligence.

1. Introduction

Internet technology is booming now, and there are countless educational learning products. Portals, such as Sohu, NetEase, Tencent, and Sina, tend to lead people to access online information and solve information shortages. The superiority of portals lies in the comprehensiveness of information content rather than the depth and quality of content [1-2].

The interactive question and answer platform, such as Baidu knows, knows about such typical products. Baidu knows that positioning is a platform for popularization and popularization. Its positioning is based on the search platform, which determines the information of the platform is broad, which is also its advantage. But the answer tends to be subjective. Knowing that positioning is a real-name network question-and-answer community, the real-name network community determines the depth of content. Some people who answer questions are experts who have in-depth research on this issue or have been engaged in the industry for many years to ensure the answer. objectivity. In general, knowing the topic level is the most detailed and the most convenient to find. The only downside is that the topic is not strictly attributed, and it is difficult for ordinary users to sort out the search structure. The main advantage of Baidu is that it is easy to search and has a high degree of matching. Therefore, Baidu knows that information is more wide than knowing it, and it knows that it is more profound than Baidu.

The information resources of the Internet are huge. These huge data contain many unknown and valuable information and knowledge. What we need to do is to filter and filter their information data and send it to users who need this knowledge. The realization of the above mechanism is based on the search engine, and it is the core idea of this project to sort out the knowledge they need for users with different needs to make the system deeper [3-4].

2. System Analysis

This platform is a search-based personalized learning platform based on search engine technology, data analysis technology, the core of which is search engine technology, and recommendations based on user-searched content.

Core technology principle is below:

DOI: 10.25236/isetbm.2019.055

2.1 Web Crawling

The web crawler is the web crawler (Spider). Each search engine has its own web crawler Spider. According to certain rules, the spider crawls the webpage continuously along the URL in the webpage.

2.2 Data Processing

After the crawler crawls the webpage, it usually captures the html source code of the webpage. At this time, the data is preprocessed. The preprocessing of the data includes extracting the content in the webpage, filtering the spam, and removing the duplicate webpage.

2.3 Search Engine

After the data is preprocessed, the search engine builds an index file and stores the data in an index file. After the data is completed, when the user performs a keyword search, the search engine performs keyword matching and returns the corresponding data engine.

2.4 User Behavior Analysis

As a search-based platform, the core user behavior is the user's keyword, which analyzes the type of knowledge of the keyword, so as to analyze the keyword, establish the entry, recommend the user to the corresponding search content, and realize personalized learning.

3. Project Design

The design of this project is based on the design idea of software engineering. The design process runs through the process of feasibility study, demand analysis, overall design, detailed design and test deployment. Determine the system implementation: Python3.7, Scrapy crawler framework, Django framework, ElasticSearch full-text search engine framework, MySQL core database, Redis NoSQL storage system, redis cache middleware.

The superiority of Python in data analysis scientific computing and its short and easy development cycle is the root cause of the project using Python as the core programming language.

Django is a powerful and very mature MVC framework. It has an ORM framework for processing databases.ES is suitable for emerging real-time search applications, and its own ability to retrieve data is very powerful. Github and Wikipedia use ES as their search service. The data in this project is the data crawled in real time. It is usually necessary to index the data while searching for it, so this feature of ES meets the technical requirements of this project [5].

The reason for using Redis in this project is to reduce the read operation of io and reduce the pressure of io. Secondly, it solves the CPU and memory pressure of the application server. In addition, the scalability of the relational database is not strong, and it is difficult to change the table structure. Many functional modules need to be Borrow redis as a cache middleware to optimize.

The front-end uses Bootstrap, which is an open source toolkit for front-end development launched by Twitter. It is simple and flexible, making Web development faster.

3.1 Data Acquisition Engine

The data processing engine is divided into a web crawler engine and a full-text search engine service. The web crawler crawls the data of the website and processes the data, and stores the accurate and complete data in the index of the search engine. The web-side integration and retrieval engine retrieval service allows the user to execute the search service, thereby realizing the data collection end. Data interaction with the web [6].

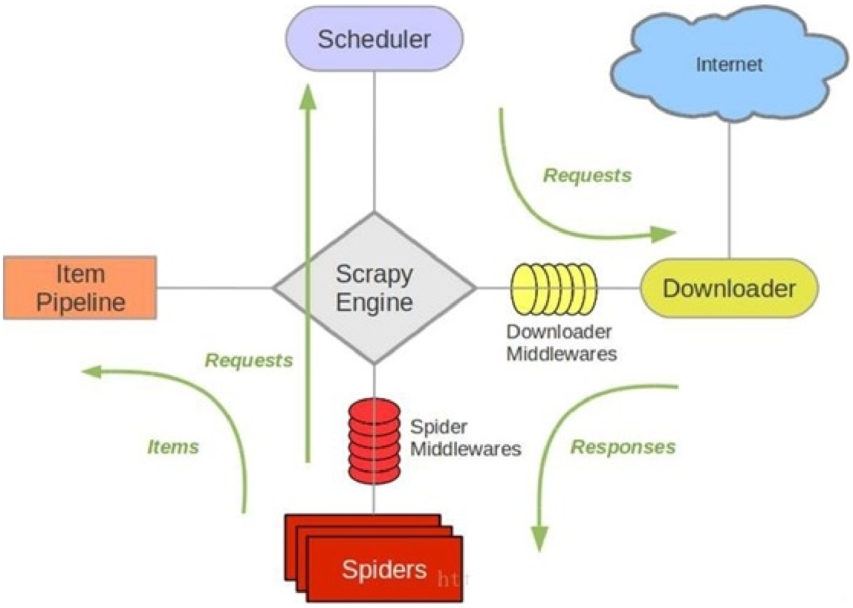
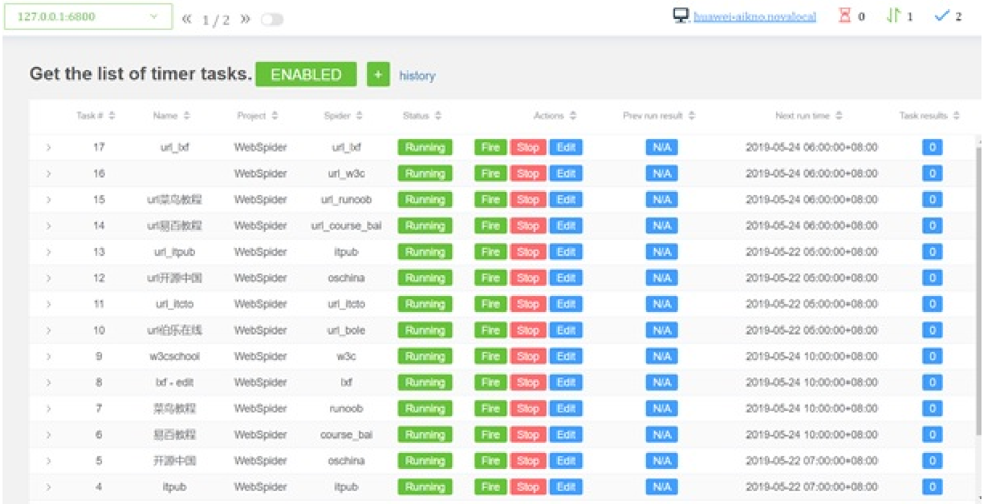


Figure 1 Scrapy Architecture

The crawler system uses the scrapy crawler framework. The main function is to download and extract data and deduplicate the data. The crawler extracts the data from the url and stores the url link into the Redis cache, and implements the deduplication of the data through the redis set data structure. By writing the extraction rules, the semi-structured extraction of the data is realized and the data is stored in the es search engine. It can be distributed across multiple servers, crawling data into es, or structuring data.



Get the list of timer tasks. ENABLED + history

	Task #	Name	Project	Spider	Status	Actions	Prev run result	Next run time	Task results
›	17	url_lxf	WebSpider	url_lxf	Running	Fire Stop Edit	N/A	2019-05-24 06:00:00+08:00	0
›	16		WebSpider	url_w3c	Running	Fire Stop Edit	N/A	2019-05-24 06:00:00+08:00	0
›	15	url菜鸟教程	WebSpider	url_runoob	Running	Fire Stop Edit	N/A	2019-05-24 06:00:00+08:00	0
›	14	url易百教程	WebSpider	url_course_bai	Running	Fire Stop Edit	N/A	2019-05-24 06:00:00+08:00	0
›	13	url_itpub	WebSpider	itpub	Running	Fire Stop Edit	N/A	2019-05-22 06:00:00+08:00	0
›	12	url开源中国	WebSpider	oschina	Running	Fire Stop Edit	N/A	2019-05-22 06:00:00+08:00	0
›	11	url_itcto	WebSpider	url_itcto	Running	Fire Stop Edit	N/A	2019-05-22 06:00:00+08:00	0
›	10	url伯乐在线	WebSpider	url_bole	Running	Fire Stop Edit	N/A	2019-05-22 06:00:00+08:00	0
›	9	w3cschool	WebSpider	w3c	Running	Fire Stop Edit	N/A	2019-05-24 10:00:00+08:00	0
›	8	lxf - edit	WebSpider	lxf	Running	Fire Stop Edit	N/A	2019-05-24 10:00:00+08:00	0
›	7	菜鸟教程	WebSpider	runoob	Running	Fire Stop Edit	N/A	2019-05-24 10:00:00+08:00	0
›	6	易百教程	WebSpider	course_bai	Running	Fire Stop Edit	N/A	2019-05-24 10:00:00+08:00	0
›	5	开源中国	WebSpider	oschina	Running	Fire Stop Edit	N/A	2019-05-22 07:00:00+08:00	0
›	4	itpub	WebSpider	itpub	Running	Fire Stop Edit	N/A	2019-05-22 07:00:00+08:00	0

Figure 2 Scrapydweb Cloud Monitoring

The operating environment of the data acquisition engine includes Cloud Server + scrapyd official deployment tool + scrapydweb visual crawler monitoring framework. The crawler's monitoring system essentially calls the scrapyd api to implement the crawler's scheduling. Since the scrapyd work will output a large number of log files, it will take up a lot of storage space of the server, so be sure to write a script file to clean the log regularly. Otherwise it will affect system performance.

3.2 Search Engine Website

The search engine website is the core project of the learning platform, providing users with the search service. When the user logs in, searching for the corresponding keyword will return the corresponding data, record the keyword search record, the user's keyword record, and the historical search record. Sorted by the number of keyword search records, the search page is formed to form a search hotspot, and the keyword part of speech is analyzed. The recommended keyword information is presented to the right sidebar in the search result interface to form a keyword classification encyclopedia, based on the user keyword record. The platform implements corresponding content recommendations for different users. When the user is not logged in, no search history and corresponding content recommendations are provided.

The project is based on Python's Django framework development. The project's system architecture diagram is shown in the figure. Based on the design of Django model, Django ORM implements database operations, and Django manage.py+ related commands enable model to generate data table structure. The Django framework decomposes the functional modules of the project in the form of an app.

4. Conclusion

The project is based on uwsgi + nginx as the production environment deployment, and based on the service environment construction of docker container.

The platform uses a series of technologies to provide people with a platform to learn the more systematically and deeply knowledge they need, and then to achieve on-demand learning and personalized learning. Through our learning platform, learners can acquire extensive and effective knowledge and enjoy an interesting learning process. The needs of learners provide a broad promotional value for our platform.

Acknowledgement

This work was supported by Taian Science and Technology development project, Taishan University Teaching reform project (201714) and Horizontal research project (2018HX098).

References

[1] Jun Xiao, Minjuan Wang, Bingqian Jiang, Junli Li, “A personalized recommendation system with combinational algorithm for online learning,” Journal of Ambient Intelligence and Humanized Computing, 2018, pp. 667-677.

[2] Priscilla M. Regan, Jolene Jesse, “Ethical challenges of edtech, big data and personalized learning: twenty-first century student sorting and tracking,” Ethics and Information Technology, 2019, pp. 167–179.

[3] Soulef Benhamdi, Abdesselam Babouri, Raja Chiky, “Personalized recommender system for e-Learning environment,” Education and Information Technologies pp 1455–1477.

[4] Nan Jing, Tao Jiang, Juan Du, Vijayan Sugumaran, “Personalized recommendation based on customer preference mining and sentiment assessment from a Chinese e-commerce website,” Electronic Commerce Research, 2018, pp. 159–179.

[5] Vohra D, “Using Elasticsearch: Pro Couchbase Development,” Apress, Berkeley, 2015, p.175-196.

[6] S.Bhaskaran, B.Santhi, “An efficient personalized trust based hybrid recommendation (TBHR) strategy for e-learning system in cloud computing,” Cluster Computing, 2019, pp. 1137–1149.