ICIC Express Letters
Part B: Applications
Volume **8**, Number **2**, February **2017**

ICIC International ©2017 ISSN 2185-2766
pp. **305–312**

DESIGN AND IMPLEMENTATION OF A WEB-BASED SIMILARITY CALCULATION AUXILIARY SYSTEM

Aijing Jia[1], Yao Liu[1,*] and Yanbing Yang[2]

[1]Institute of Scientific and Technical Information of China
No. 15, Fuxing Road, Beijing 100038, P. R. China
*Corresponding author: Liuy@istic.ac.cn

[2]Department of Mathematics
Dalian Maritime University
No. 1, Linghai Road, Dalian 116026, P. R. China
yyb9480@hotmail.com

Received July 2016; accepted October 2016

Abstract. *With the in-depth development of data-driven research method, the demand for the depth Natural Language Processing technology is increasing in library and information industry. The way of data-driven research is becoming a mainstream way. However, the existing limited research tools are inconvenient for the library and information researchers who cannot understand the engineering technology well. To solve the problem effectively, the paper designs and implements a web based similarity calculation auxiliary system. We achieved the online similarity calculation and comparison through the plug-in technology and customized pipeline processing technology. Users can realize their own similarity algorithm, and customize the individual components of the processing flow, so as to implement a more logical and targeted similarity calculation process. Taking patent text for example, integrate and test the four similarity algorithms. From the results, the system can achieve the online similarity calculation and comparison for the intelligence researchers.*

Keywords: Similarity calculation, Web, Plug-in technology, Pipeline processing technology

1. **Introduction.** In the library and information industry, the way of data-driven research method is becoming a mainstream way, and the demand for the depth natural language processing technology is increasing. The intelligence researchers need similarity calculation tools for their research about the variety of resources. However, the existing tools are inflexible and homogeneous, which cannot satisfy the needs of the intelligence researchers. A survey found that Da described a framework of the plug-in technology [1] and Zhang described a customized pipeline processing technology [2], so integrating their research with some similarity calculation algorithms, this paper designed and implemented a web-based similarity calculation auxiliary system.

With the plug-in technology and customized pipeline processing technology, users can integrate their own algorithm into the system by plug-in. According to the type of resource they studied, the users can select the appropriate processing unit, such as word segmentation tool, similarity algorithm, so as to customize their personalized similarity calculation process. The system has strong expansibility, pertinence and high efficiency for using the plug-in and pipeline technology. For a type of resource that researchers studied, the system can be a complete process formed by rapid combination of components, which greatly improves the efficiency of research.

In Chapter 2, we described the design of the system, including the framework of the system, the structure of the file and the plug-in interface; Chapter 3 described the function and implementation of the system, including system module and function; in Chapter 4,

some experiments were conducted to test the platform; Chapter 5 gives the conclusions of the research.

2. System Design.

2.1. **Frame design.** The system is based on the B/S (Browser/Server) 4-layer structure, as shown in Figure 1. It is a plug-in framework system divided into four layers: presentation layer, processing layer, service components layer and data layer [3].

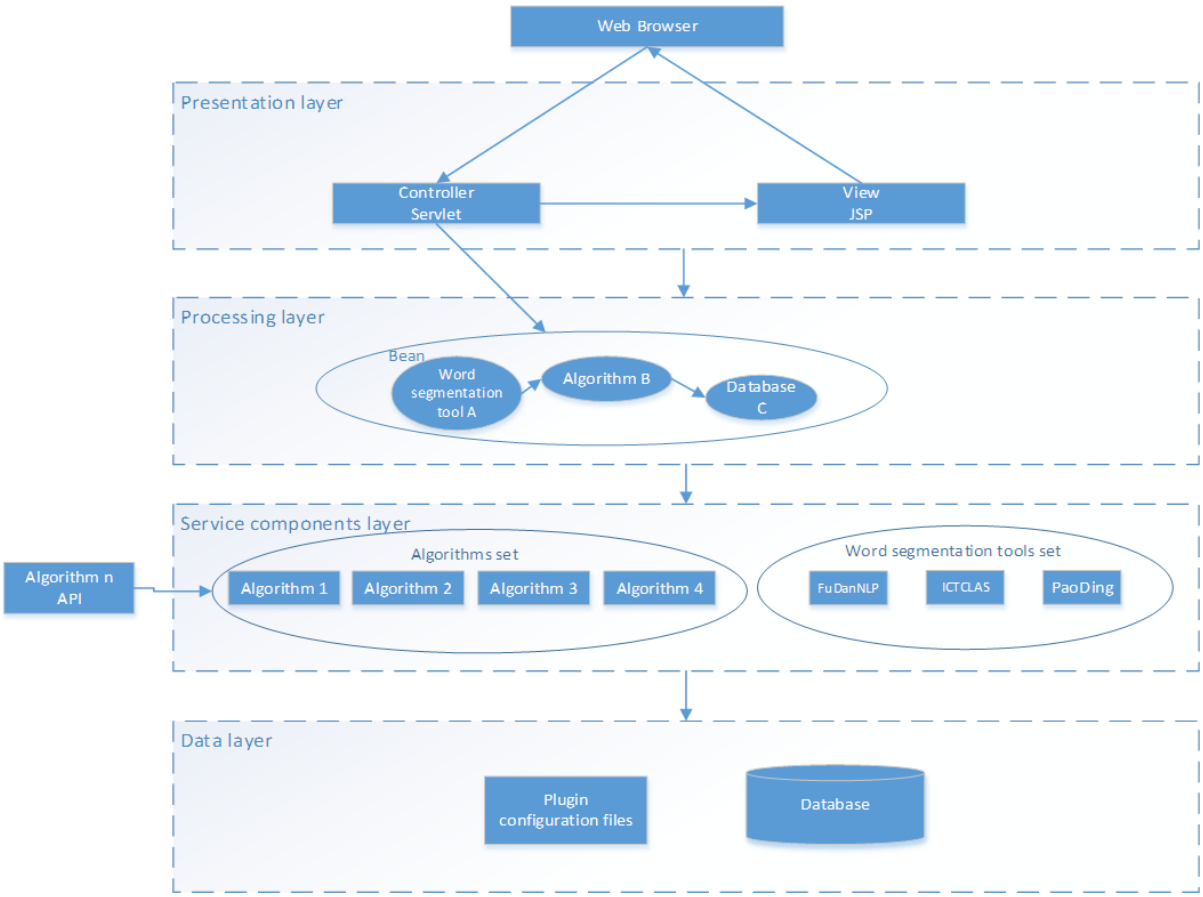

FIGURE 1. Four layer framework of the system

Combined with the MVC design model, we constructed the framework of the similarity calculation auxiliary system by using the J2EE (Java 2 Platform Enterprise Edition technology). When users operate according to the page, the HTML or JSP web pages of the Web client submit applications and sent to the server. The servlet controller from the server-side sends the operation to different Bean according to different operations, including word segmentation operation, stop words removing operation, the similarity calculation operation, etc. Then the processed information is sent back to the view JSP and the view JSP returned it to the page displayed by HTML after the information is processed at the server side [4].

The pipeline applications can simply execute a set of customized selected components in proper order. In the processing layer, there is a series of process flow assembled by the pipeline technology including the word segmentation tool, stop words removing tool and the similarity calculation algorithm. We use plug-in technology in the service components layer, and provide a unified interface, and the users can realize their similarity algorithm through this interface [5].

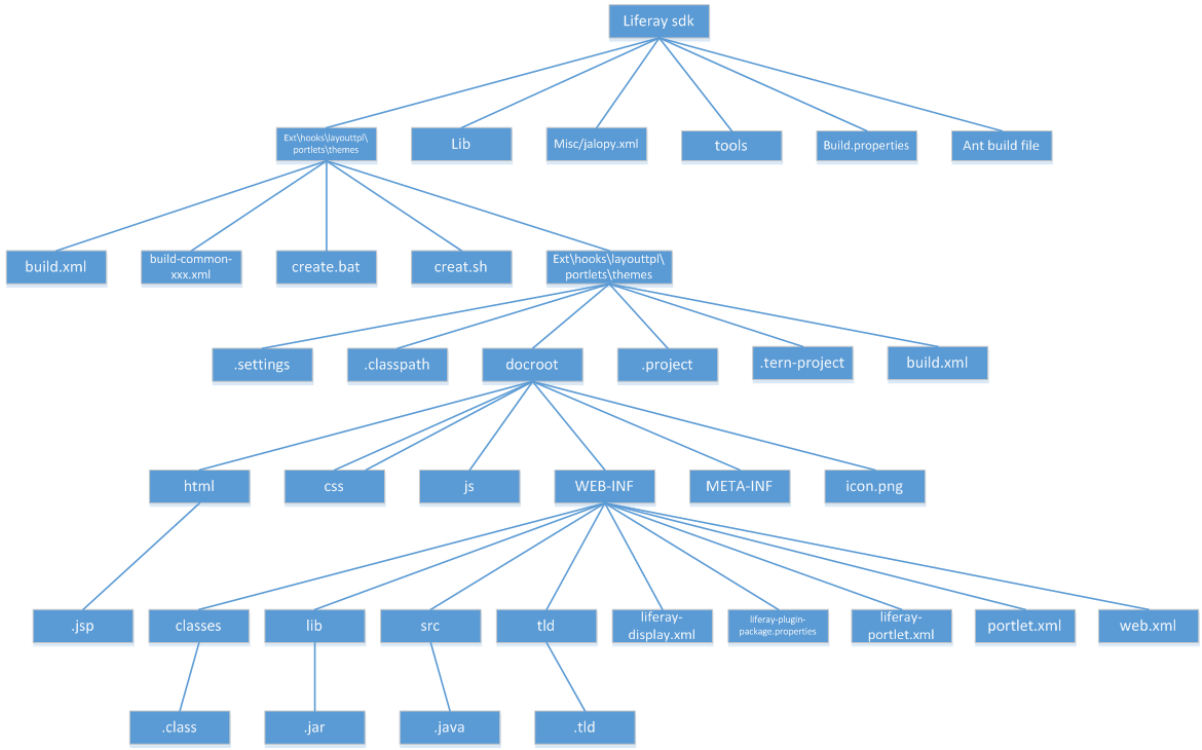

FIGURE 2. The file structure of the system

2.2. **The file structure of the system.** The system is built on the Liferay platform [6,7]. Figure 2 shows the file structure.

Analyzing the files, the plug-in code is written in liferay-sdk\portlets*-portlet\docroot \WEB-INF\src*.java, the page displaying code is written in liferay-sdk\portlets*-portlet\docroot\html*.jsp, and the two files are corresponding one to one. The result of the similarity calculation in the background of the system can return to the page displayed by the render method.

2.3. **The plug-ins interface design.** The system embedded the new algorithms by the plug-in technology. Using the structure of the main frame and extension. The interface of the main frame is called by the plug-ins to implement the function of the plug-ins. We constructed an AddInTree as the main frame.

There is the pseudo code of constructing the AddInTree:

```
AddInTree.CreateAddInTree () {
        AddInTree = new AddInTree (); //create a new default AddInTree object
        AddInFiles fileUtilityService.SearchDirectory
(defaultCoreAddInDirectory, "*.war"); //select the war extension
        InsertAddIns (AddInFiles); //insert a plug-in to the tree
}
```

There is the pseudo code of inserting a plug-in into the AddInTree:

```
Static void InsertAddIns (String AddInFiles) {
        AddIn addIn = new AddIn (); //create a plug-in object
        addInTree.InsertAddIn (addIn); //insert into the AddInTree
        }
```

3. System Function and Implementation.

3.1. **System module.** It is mainly divided into two modules: the similarity calculation module and the management module, as shown in Figure 3.

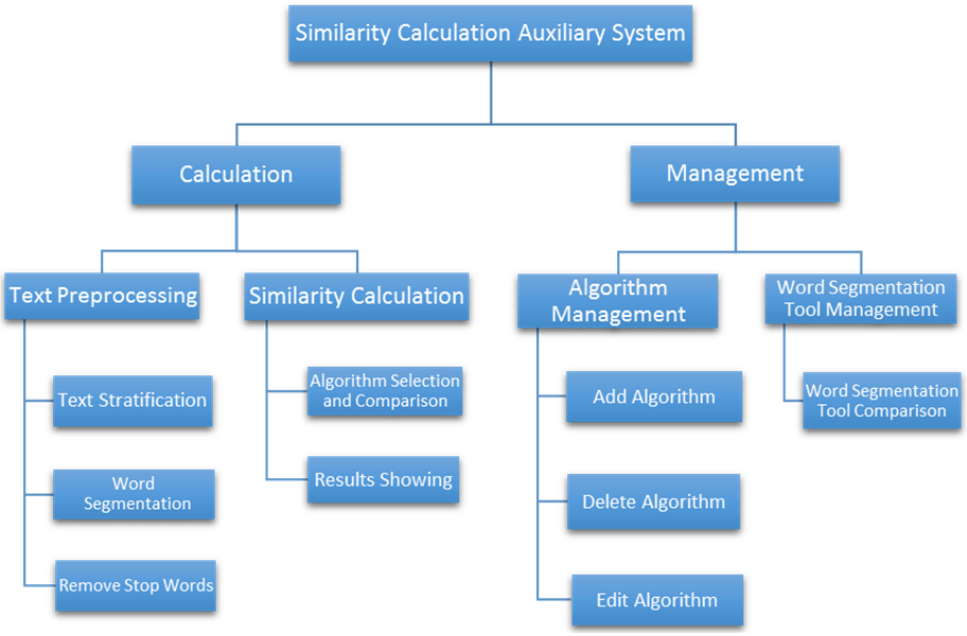

FIGURE 3. System function module

3.1.1. *The similarity calculation module.* It includes text preprocessing module and similarity calculation module.

The text preprocessing module includes the text stratification module, word segmentation module and remove stop words module. The text preprocessing is to structure process documents, which provides a necessary condition for the similarity coefficient calculation.

The similarity calculation module consists of the algorithm selection and comparison module and results showing module, which can give users intuitive results of the similarity calculation and comparison.

3.1.2. *Management module.* It includes algorithm management module and word segmentation tools management module.

The algorithm management module includes add algorithm, delete algorithm and edit algorithm. Users can add their own similarity algorithm into the system, delete the useless algorithm as well as edit the weight value of each layer in the algorithm.

The word segmentation tool management module consists of the word segmentation tool comparison module. The results of different word segmentation tools for various resources are different, and users can compare the results online, and choose the appropriate one in determining a customized pipeline process.

3.2. **The system function.**

3.2.1. *Similarity calculation and comparison.* The core function of the system is to calculate the similarity coefficient, which reflects the concept of customized pipeline technology. Figure 4 shows the similarity calculation page.

There are two parts in the page, algorithm selecting and text accessing.

In the algorithm selecting, users can choose up to 4 similarity algorithms in the multi boxes and then calculate the similarity coefficient. Four algorithms are integrated in the system already: VSM algorithm, VSM based on hierarchical text algorithm, VSM based on hierarchical text and word co-occurrence with the sentence as the unit form algorithm, VSM based on hierarchical text and word co-occurrence with the adjacent words as the unit form algorithm. Users can select their own similarity algorithms here, and then implement the similarity calculation.

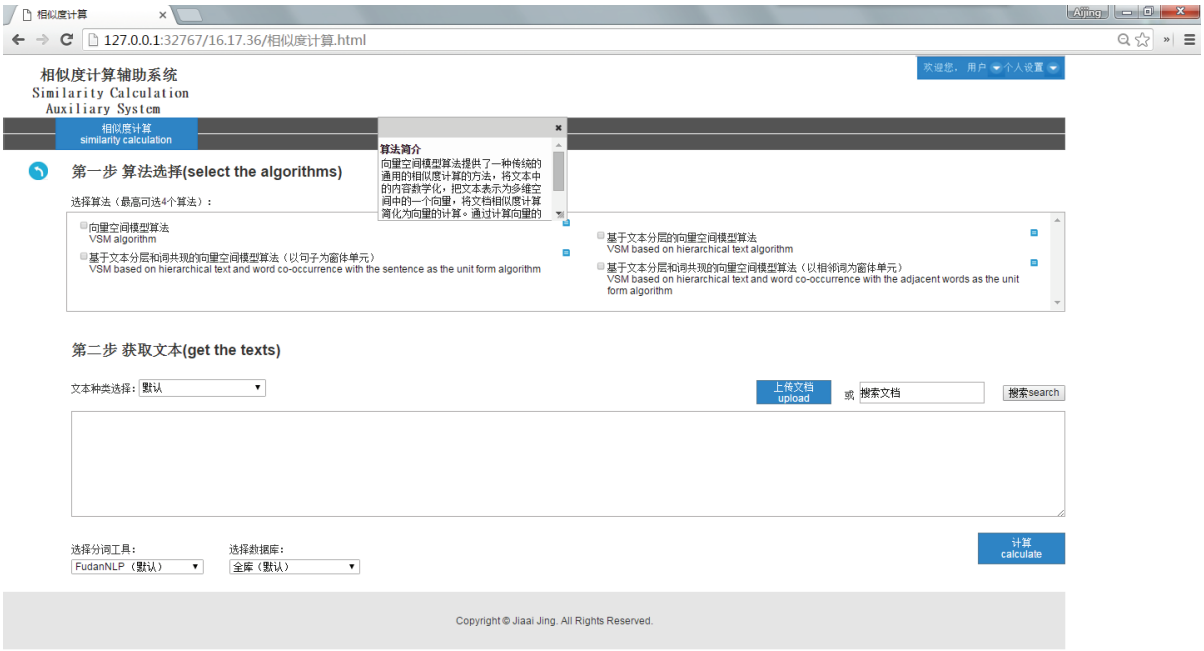

FIGURE 4. Similarity calculation page

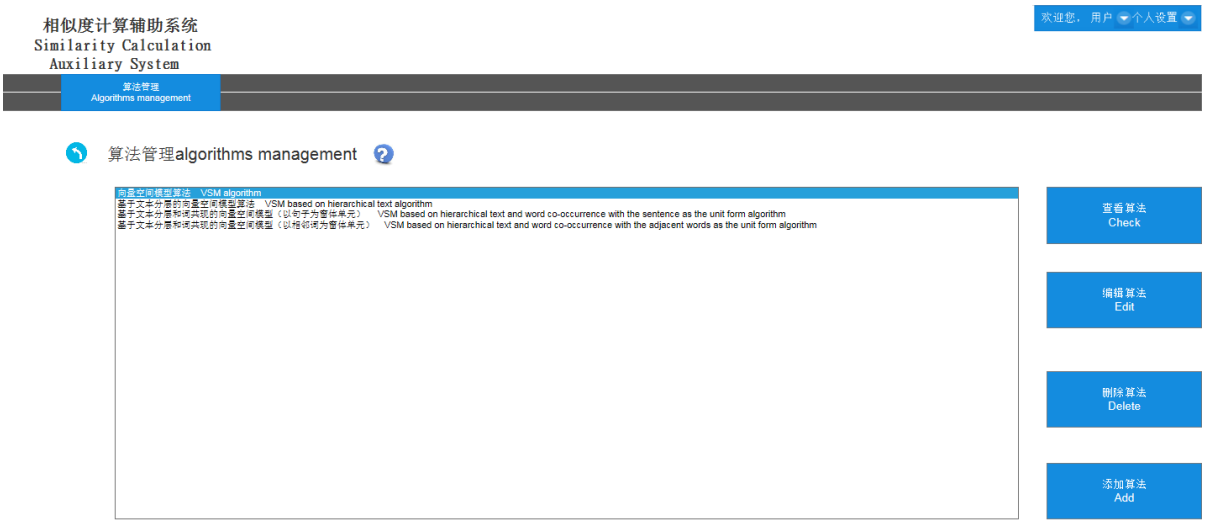

FIGURE 5. Algorithm management page

In the text accessing, users can retrieve document in the system or upload document. They need to choose the type of the uploaded document, which has three types provided by us: default, patent and paper. After uploading the document or selecting the document, the full text of the document is in the text box below. Then choosing the Chinese word segmentation tool, we provided three types here: FuDanNLP, ICTCLAS and PaoDing, default the FuDanNLP. The next step is to choose the compared database, and we provided just one chemical patent database. Click the calculation button.

The system provides a unified interface, users can integrate their algorithms implemented of a unified model to the system by plug-ins, achieving their own similarity algorithms for online testing. It can compare the advantages and disadvantages of their own algorithms with other's on a certain resource.

3.2.2. *Algorithm management.* Algorithm management is an important function of this system, users can operate the algorithms in the system according to their needs.

The algorithm management page, as shown in Figure 5, can view, edit, delete and add algorithms in the system.

In the algorithm editing function, design an interface for editing the algorithm. Taking the VSM based on hierarchical text algorithm for an example, we divided the patent into 3 parts: title, abstract and claim in text preprocessing, and each part has a weight value. We can change the weight value through the interface for a more appropriate result.

In the algorithm adding page, as shown in Figure 6, users can embed their own similarity algorithms into the system as plug-ins. Click the upload document button, select the file, click install button, and the system prompts. Users can see and use their own similarity algorithm in the similarity calculation page.

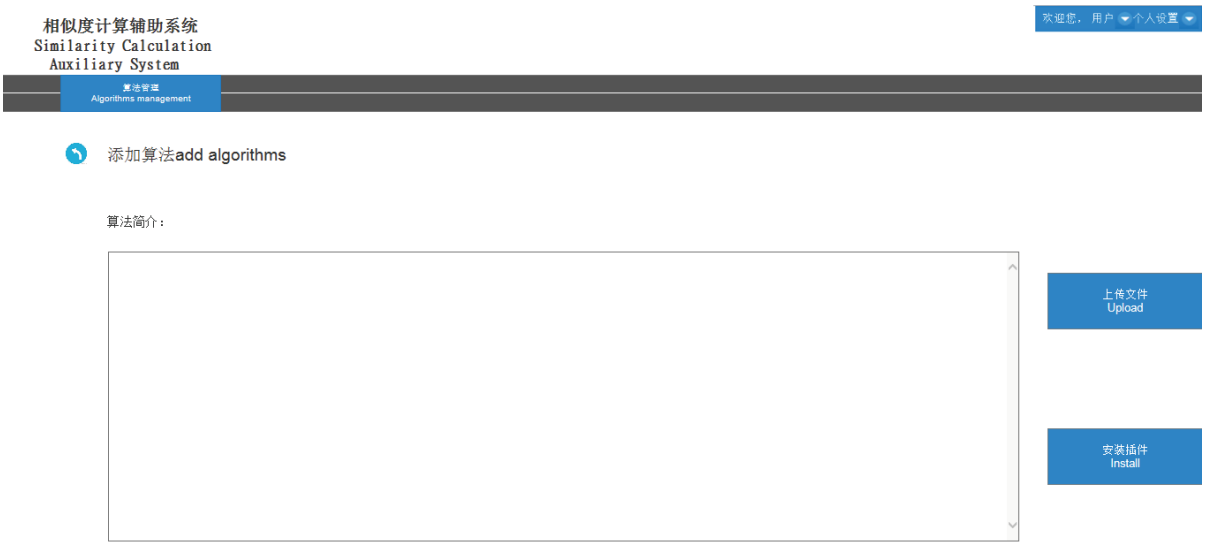

FIGURE 6. Algorithm adding page

3.2.3. *Word segmentation tools management.* Word segmentation is an important part in the similarity calculation, and the result of the word segmentation greatly affects the result of the similarity calculation. Three provided Chinese word segmentation tools will have different results for different resources. This function provides an online segmentation compared tool for users and enables the customized technical process more efficient.

4. Experiment and Result Analysis.

4.1. **The basic flow.** The basic flow of the similarity calculation is shown in Figure 7.

4.2. **Results analysis.** Select the chemical patent database as the contrast database, and the patent text "Safe and efficient compound food preservative" as the measured text. We used the four similarity algorithms existing in the system for the experiment. The result is shown in Figure 8.

The experimental result reflected that the system can calculate the similarity coefficient between the measured document "Safe and efficient compound food preservative" and the documents in the database with different algorithms, and show the results intuitively. The four algorithms in the system are suitable for calculating the patent documents.

5. Conclusions.

This paper introduced a web-based similarity calculation auxiliary system. We analyzed and discussed the framework of the system design, file structure design, plug-in interface design, the system functions and the key technologies, etc. Realize the similarity calculation, the similarity comparison, the new algorithm insertion and the online word segmentation tool comparison, etc. Furthermore, we took the patent text and the four similarity algorithms as an example, and tested the effect of the platform. The

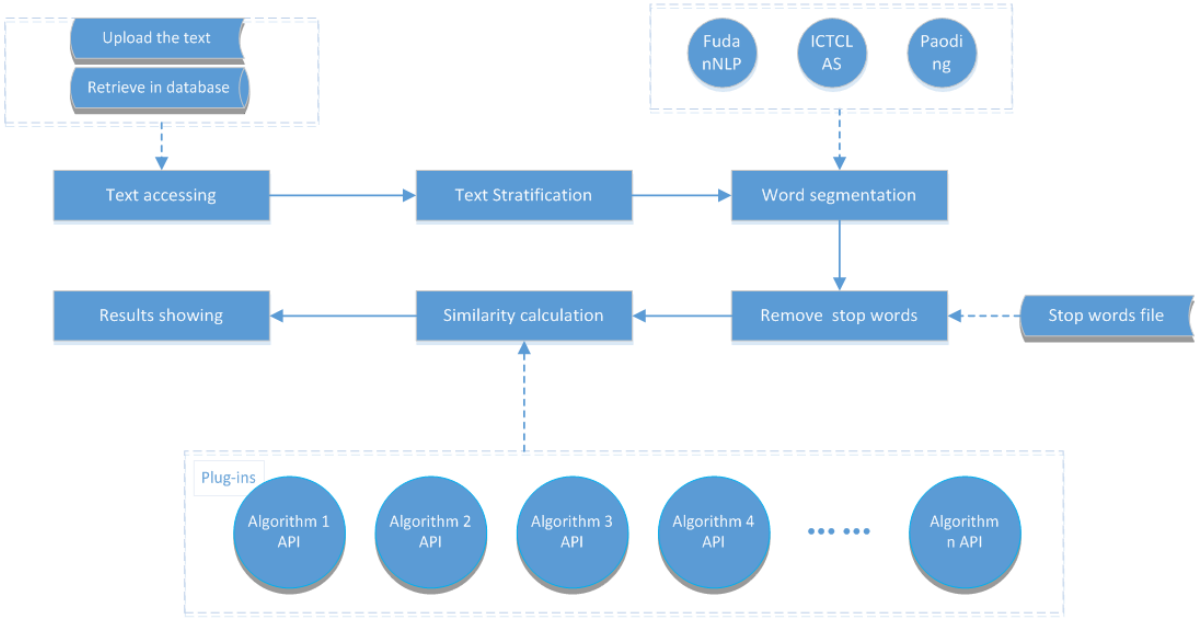

FIGURE 7. The basic flow of the similarity calculation

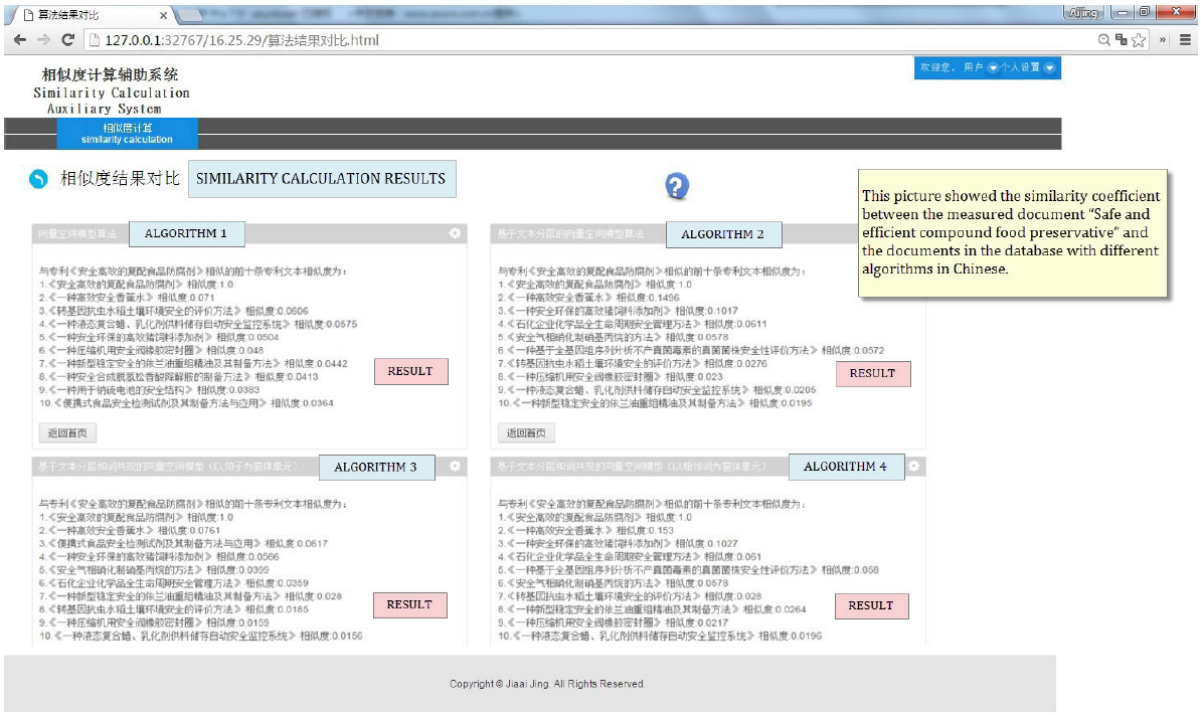

FIGURE 8. Similarity calculation results

results show that this system can be used as a platform for online similarity calculation for library and information researchers. It is much more convenient and can improve their research level efficiency.

However, this experiment is only conducted on the patent documents, the results of other documents are still unknown, and it needs a large amount of experimental data to support. In addition, this system can only combine the tools that already exist. The trend is to make a knowledge-based system which the tools do not exist in the future, and the system can automatically analyze new tools and process for all kinds of texts.

Acknowledgment. This work is supported by Scientific Research Project of Liaoning Provincial Education Department (No. L2015072).

REFERENCES

[1] W. Da, *Research on Addin Framework Development*, Jiangsu University, 2013.
[2] Z. Zhang, *Research and Develop the Semantic Annotation Platform of Scientific Literature*, Beijing University, 2014.
[3] Y. Wei, On the three-tier structure system of browser/server, *Journal of Huangshi Institute of Technology*, vol.23, no.1, pp.53-55,59, 2007.
[4] R. Lu, Z. Yu, Y. Ruan and Z. Wang, Study and implementation of MVC design pattern on J2EE platform, *Application Research of Computers*, vol.20, no.3, pp.144-146, 2003.
[5] X. Feng, K. Cui and J. Shen, Research and implementation of plug-in oriented application framework, *Computer Engineering and Applications*, vol.45, no.10, pp.89-91, 2009.
[6] Y. Xia, *The Design and Implementation of Agriculture Information Portal Based on Liferay*, Master Thesis, Harbin Institute of Technology, 2010.
[7] X. Li, *The Design and Realization of Campus Information Portal System Based on Liferay Portal*, Tianjin University, 2013.
[8] X. Chi, *Design and Implementation of a Recommender System Framework Based on Collaborative Filtering Algorithm*, Master Thesis, Shanghai Jiao Tong University, 2013.