The Framework of context extraction and ranking from web text data

Wonjoo PARK*, Kyong-Ha Lee*, Cho Kee Seong*
*Smart Media Platform Research Section ETRI, Republic of Korea
E-mail : wjpark@etri.re.kr, kyongha@etri.re.kr, chokis@etri.re.kr

1. Introduction

Traditionally, nations and states have recorded the government data, information and others. They are a large amount of datasets that contain diverse range of government activities, national cultures, regional statistics, and so on. In recent years, numerous nations and public-sector bodies release own data to share and reuse on Web portals. Also, public datasets can be utilized a variety of applications and value-added services by interlinking with web data semantically.

In this paper, we focus on context extraction and ranking framework in the web data according to context similarity. Recently, we develop a platform to collect and process public data and study to practice the platform. To verify the benefits from the availability of the public data, we develop the service scenario in a variety of ways. Such examples are health care service, diet control service and life log service. However, there is a limit if a service is made with only public data. Therefore, it is important to interlink with web data, social data and so on.

In this paper, we propose the framework context extraction and ranking from web text data. In this work, we have made three corpuses - landmarks of Seoul, outdoor activities, events in Seoul - to extract context about outdoor, event in Seoul. The contents of web text data are collected by utilizing open API of Naver and Daum.

2. Architecture and Development

Our approach is basically to develop a framework that it context extracts from web text data ranks the top K by calculating the context score. A large collection of web text data is indexed by Lucen and it is classified documents including outdoor activities and events. It calculate the occurrence frequency of each among the documents. The occurrence frequency is key value to rank context.

Overall system architecture is as follows. : a distributed data store(MongoDB/GridFS/Lucene), Web Collectors, Data refiner and Client Server.

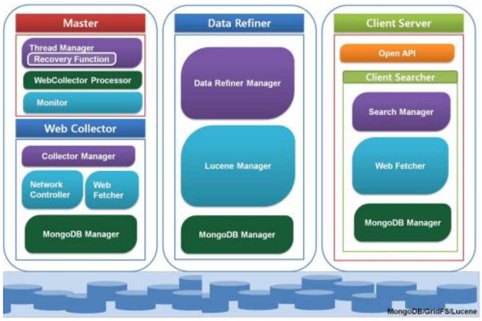

Figure 1. System architecture

Web collector gets the web text data and is composed of two modules, the Web Collector module and Master module. The Web Collector module performs the actual collection. Web Collector Master module can be performed in a multi-threaded, ensures that the monitoring of conflict, deadlock detection and recovery process to provide support. Thus, it performs to collect of web text data steadily.

Master module provide generation, destruction, conflict control, deadlock detection, recovery, treatment and support of Web Collector modules. A number of Web Collectors actually gathers web text data(html) from Daum blog, Daum café, Naver blog, Naver café.

Data Refiner is composed of a Data Refiner Manager module and Lucene Manager module. It performs indexing and classifies documents by using the corpuses (Landmarks, Outdoor Activities, Events). Also it calculates scores and ranks documents on their scores of correspondence. The flowchart of the Data Refiner is as shown in the following Figure 2.

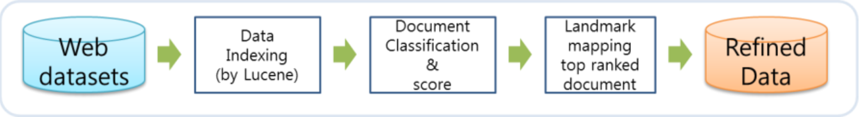
Web datasets
Data Indexing (by Lucene)
Document Classification & score
Landmark mapping top ranked document
Refined Data

Figure 2. Data Refiner module

A large datasets of web text data are indexed by Lucene and searched documents including outdoor activity and event keywords. Here, search fields are title and contents. In this paper, we use the Lucene score algorithm and is as follows.

$$\mathrm{score}(q,d) = \sum_{t \in q} (tf(t \in d) * idf(t)^2 * getBoost(t \in q) * getBoost(t.field \in d) * lengthNorm(t.field)) * coord(q,d) * queryNorm(sumOfSquaredWeights)$$

$$sumOfSquaredWeights = \sum_{t \in q} (idf(t) * getBoost(t \in q))^2$$

Lucene scoring uses a combination of the Vector Space Model (VSM) of Information Retrieval and the Boolean model to determine how relevant a given Document is to a User's query. In general, the idea behind the VSM is the more times a query term appears in a document relative to the number of times the term appears in all the documents in the collection, the more relevant that document is to the query. It uses the Boolean model to first narrow down the documents that need to be scored based on the use of boolean logic in the Query specification. Lucene also adds some capabilities and refinements onto this model to support boolean and fuzzy searching, but it essentially remains a VSM based system at the heart[2]. After calculating score of each document, Landmark is mapped top ranked documents.

Client Server is end point for users and mobile applications to get the refined datasets. If mobile applications with the location information query by using Open API, Client Server searches the corresponding web text data and return top K web page URL in JSON format. We develop the mobile application to verify the framework and algorithm as shown Figure 3.

Figure 3. Mobile application with public and web data.

3. Future works and conclusion

We developed and tested the framework of context extraction and ranking from web gathering text dataset. It is interlinked public datasets semantically and used for mobile application. It is scheduled to extend to a variety of corpuses and diverse web dataset.

4. Acknowledgements

This research was funded by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2014

5. References

[1] Li D., Vs P., Mi H., “Linked Open Government Data, IEEE Intellignet Systems magazine, May/June, pp 25-31, 2012

[2] W. Park, K.H. Lee, Cho K.S. “Development and experiment of semantic similarity between public linked data and web data ”, International Conference on Computers, Communications, and System(ICCC), October, 2013

[3] http://lucene.apache.org/core/362/scoring.html