

Diversify Search Results with Facet Indexing

Xiaoliang Tian

Abstract—Based on the novel academic system, AceMap, which can present the big scholarly data through "map" approach. However, for the traditional search engine design, some optimizing methods are still needed to improve the user experience. In this report, I will describe the additional designs, diversifying search results with indexed attributes, on the search result page which I have done in this semester, with Solr platform and Ajax technique.

I. INTRODUCTION

ACEMAP[1] is a novel academic search and visualization system where informations are displayed through "map" approach. For example, users can get an intuitive impression of the paper and its relationship with others, just like checking a place on a geographic map. As for the search results, we have applied Solr[2], an open source enterprise platform from the Apache Lucene project, to implement the query function. Solr has three important methods: indexing, searching and faceting. Solr supports indexing structured content in a variety of incoming formats, such as XML, JSON, CSV, etc. It can be queried via REST clients, cURL, wget, etc, and it is able to perform a filter query limiting the response to particular fields. The most popular feature, faceting, allows the search results to be arranged into subsets (or buckets or categories), providing a count for each subset.

With such powerful tools, we can easily diversify the search results into different fields and present the faceting results with statistical charts. However, the faceting result brings poor user experience since it takes lots of time, especially those have large number of docs returned. What I mostly cared about during my current research is this issue. The rest of the report is organized as follows. First, the solution to this issue is given in Section 2. Then I introduce the additional improvements in the search results in Section 3. Finally, it is the conclusion part and the future work under consideration.

II. ASYNCHRONOUS REQUEST WITH AJAX

A well-known solution to the problem mentioned above is to perform an asynchronous request. Because searching without faceting brings fast response, we can begin the faceting request after the response without faceting returned. In this way, users can read the result lists in the left part of the search page, and at the same time waiting for the faceting results. To implement this, we should apply Ajax(Asynchronous Javascript And XML)[3], a technique used to implement a web application that communicates with a server in background, without interfering current page.

It is easy to insert an Ajax function in a single HTML file, however, for the whole AceMap project written in PHP,

we need to figure out the working process first and then we can modify the codes. It is obvious that the project uses the popular design pattern, MVC(model-view-controller)[4], decoupling these major components allowing for efficient code reuse and parallel development. Figure 1 shows the interaction between users and the web application based on MVC. The model is the central component which manages data, logic and rules of the application, and a view refers to a web page representing information loaded from the controller, whose task is to send commands to update its associated view. In the AceMap system, users see the view entity, which is usually a web page with contents provided from the model. In fact, users send inputs or commands through view and create a new request, and then controller will parse the request so as to manipulate the model to get the retrieved data. After that, the controller loads the corresponding view and fills it with the data from the model, which forms a completed response. Finally, the response reaches the browser so that users can see the updated page.

Another issue is that the same-origin policy prevents Ajax from using across domain functions, thus we should use JSONP(JSON with padding) to support cross-domain requests. In Figure 2, we represent a sequence diagram to specify the whole process of the asynchronous request. After the user entered a query in the view entity, the view sent a request without faceting to the 1st controller, returned with a result list, and this view started to render the list into the current page. Once the page is ready, an Ajax request is sent to another controller(2nd) which would perform the faceting search with the same query, and the response containing the faceting results was sent back to the view so that the user can see faceting results in different charts.

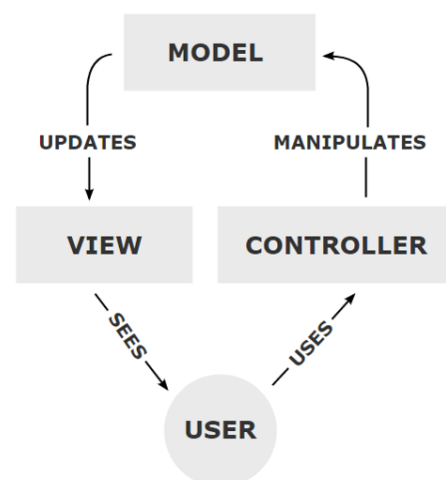


Fig. 1: Diagram of interactions within the MVC pattern

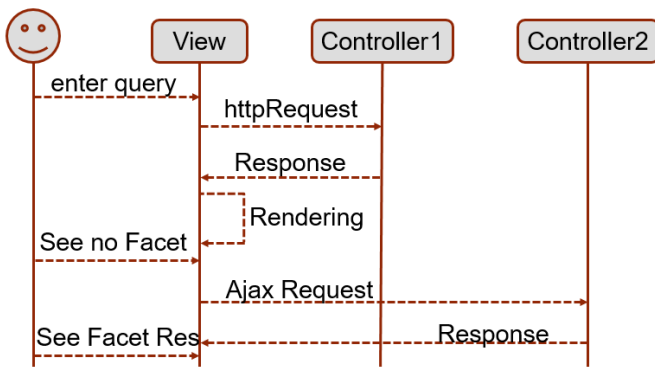


Fig. 2: Sequence diagram of the asynchronous request process

III. FACETING RESULTS

I have selected five important fields that users are interested in, that is, *paper published year*, *keywords*, *related authors*, *corresponding conferences* and *journals*. After loading the faceting results, five statistical charts, 4 pie charts and 1 line chart, are represented in the right part of the result page, providing intuitive graphical interface and helping users to have a better understanding about the result distribution. Figure 3 is the statistical charts when querying 'wireless communication'. As we can see, users can easily figure out those top relevant keywords, active related authors, important conferences and journals, by the percentage of different items. Besides, hovering over the item of the chart will display a small box with item name and corresponding paper number. If you click on this item, it will jump to a new result page, starting the same query with this item value as constraints. For example, once you click the item named 'INFOCOM' in the pie chart of conference, the browser will send a request to search papers only from the 'INFOCOM' conference so as to narrow down the result sets.

It is quite easy for developers to add those fancy charts into web pages with the help of various JavaScript libraries or toolkits. The difficult part is to index those data into the Solr platform. First we should find out the tables corresponding to those fields of data in the database, and then we recorded the whole data into some structured documents, such as XML or JSON, since Solr supports indexing structured content in a variety of incoming formats. Next, we just posted those documents to Solr's endpoint and began indexing with well-defined commands. Finally, we can browse the documents indexed at the server by the */browse* UI.

It is also interesting to observe the tendency of the amount of published papers through years, and it seems that the amount of publications has a giant leap between 21th century and before. Thanks for the Internet, people now can access a great deal of research papers. However, the completion is everywhere and your new idea will be discovered by others soon if you have not taken your action in time. Thus the only thing we can do is to keep on learning, so as to rise to the challenges.

IV. CONCLUSION

During this semester, I have finished three tasks on the improvement of AceMap system.

- Partially solved the delay response due to faceting by performing a asynchronous request with Ajax.
- Designed the statistical charts to display the faceting results.
- Indexed important fields into Solr platform so that it can provide those facet results.

However, I cannot claim that those faceting results are useful before users agree so. We need to ask some researchers to use it and let them provide insightful feedback. Only in this way can we correctly evaluate our improvements on AceMap system.

The future work should be focused on improving the Solr performance. The Solr platform still takes plenty of time to complete statistical analysis. What's more, we may develop some algorithms to display the relationship among different research fields.

In the end, I want to thank *Qianyang Peng* and *Wei Sun*, my seniors in the AceMap team, for helping me with the project. And thank Professor *Xinbing Wang* for bringing us to do research on AceMap, an academic system filled with excellent jobs done by my teammates and I learned a lot from them.

REFERENCES

- [1] Z. Tan, C. Liu, and Y. Mao. AceMap: A Novel Approach towards Displaying Relationship among Academic Literatures *WWW '16 Companion*, Montral, Quebec, Canada, pp. 437-442, April 2016. <http://acemap.sjtu.edu.cn>
- [2] Grainger T, Potter T, Seeley Y. *Solr in action[M]*. Cherry Hill: Manning, 2014.
- [3] Garrett J J. Ajax: A new approach to web applications[J]. 2005.
- [4] Krasner G E, Pope S T. A description of the model-view-controller user interface paradigm in the smalltalk-80 system[J]. *Journal of object oriented programming*, 1988, 1(3): 26-49.

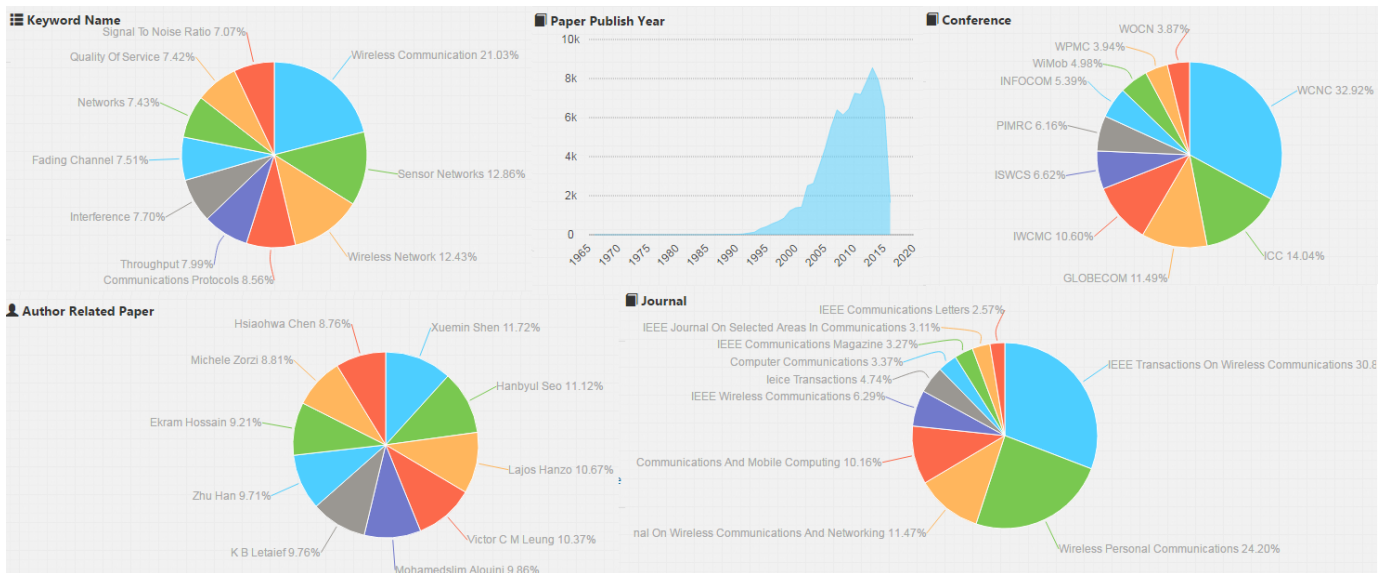


Fig. 3: Statistical Charts by querying 'wireless communication'