Generating Interactive Tag Clouds on Scientific Articles

Visualizing clusters as tag clouds

Having a large collection of scientific articles grouped by topics, it was thought that the best way to visualize the main topics of these groups was through the use of tag clouds. At first glance, one is able to recognize what the articles in a certain group are about.

[img of cluster]   [img of tag cloud]

A system that enabled users to search through topics to find the corresponding group of articles and the involved artists is the **objective**. The <u>key</u> is the interaction between the user and the tag cloud.

Tackling the problem (might be too bulky)

Several aspects were taken in consideration when creating the cloud cloud, such as:

- Stopword filtering. Unimportant words in the given context were discarded. Solution: [Snowball library]
- Word stemming. Words with the same root were grouped together. Solution: String search.
- Language detection. Articles in other languages were to be excluded from the tag clouds. Solution [language-detection library]
- Display the tag cloud. Solution [OpenCloud library]
- Portability. As the intention was to reach as many users as possible, a web environment was chosen. [HTML5, CSS, Javascript, Servlets]

Initial solution

The initial solution made use of OpenCloud, a library in Java that facilitates the creation of tag clouds for the web. Using HTML and CSS, the tag cloud was given the desired styling and presentation.

<img using opencloud>

Evolving approach

Having in mind the **objective** of the project, it was thought that the items in the tag cloud could be better positioned to illustrate the relationship between similar and important topics in the group of articles.

[img of new tagcloud with hmtl5]

Interacting with the Users

The key to get more insight on the groups of articles was to allow for interaction with the tag cloud. An user can click an item in the tag cloud to find out who are the active researchers in that area, as well as to find out in what other topics they are involved. HTML5 and its canvas was used to retrieve information on the clicked item.

[img mock up of idea]

Indexing articles

To successfully deploy a system where topics inside the groups of articles were to be quickly searched, a structure known as an *index* was chosen. Each document is indexed, so that the fields, in other words, the topics of the articles, can be queried.
Solr, a powerful tool to index documents in the web was used. It provided a simple interface between the data stored and the means of returning the desired information.

[potential img of index struct]

Future Work

Although the project focused on the many components involved, there is much work to be done to integrate these parts into a system for use in the web. Future work will focus mostly on deploying an architecture where the user inputs topics as queries, and the server processes these queries to return the tag clouds where such topics have been seen in certain groups of articles.

References