This project is done by Anbang Xu and Weizhe Ni

Project3 – CS221(Information Retrival)

Crawl the ics.uci.edu domain

We started by crawling the whole ics.uci.edu domain in the second project. We take the maximum depth of crawler as 20 and store all the URLs, titles, outgoing, texts and htmls in the web pages we crawled. We run the crawler with the filter excluding the question mark('?') and/or the hyphen('-'). We stored all the data collected and compare the result of the most common words and 2-gram words.

Tools: crawler4j

Build index

We build our index by processing all the data and output the number of documents, the number of words , the number of unique words, common words, common 2-grams and so on.

Tools: Lucene

Text processing

1. computer word frequency
2. computer term(2-gram/3-gram) frequency
3. computer inverted index
4. computer word position in each url
5. generate the bold character

Develop page ranking criteria

We developed page ranking criterias mainly based on the following aspects. Anchor text: using Field.setBoost(8.0f) in index time to add weight to anchor text; Title: using Field.setBoost(5.0f) in index time to add weight to title; Non-HTML content(Text): using Field.setBoost(1.0f) in index time to add weight to content; Text Characteristics: get the bold fonts and some special characters from html content and add more weight to these characters in index time; Page Rank: a. transfer outgoing link to matrix; b. use hadoop to implement page rank algorithm; c. generate the top 30 pages; In final step, assigning different weights to the previous technologies. And make use of Lucene to adjust the weight to anchor text, title and content and compare the results in order to generate the best parameters. Additionally, based on the query results from Google and comparing the NDCG of Google result and our search result, we add some heuristic function to adjust lucene parameters in order to get better query results.

Tools: Hadoop, Pregelix, Google Customer Search, Lucene, Json

Build up UI

We build up the user interface mainly by compiling in HTML and call JAVA servlets in JSP. There are mainly two page instances. First is the home page of our search engine. It contain a search box, a submit icon, two links to Google.com and UCI home page respectively, three other links to some explanation

about our UI. Each with some hover in color and shape. Second is the result show page, which contain a text representing area, a search box and some interesting links. Learn more tells you about how we carry out the whole project.

Tools: Tomcat, JSP, servlet, html