CS298 Progress Report

Topic: Full-Text Indexing for Heritrix

Prepared By: Darshan Karia (006960200), darshan.karia@students.sjsu.edu

Introduction:

I am working on my master's project, Full-text Indexing for Heritrix since Fall 2010. In Fall 2010 semester, I analyzed working of Heritrix, and performed some sample crawls to understand crawl parameters and retrieve sample archive files. I also made modifications to the code of Heritrix to understand the work flow of the code. I read chapters regarding Inverted Index from Information Retrieval Book.[1]

In Spring 2011 semester, my plan was to give page rank to documents, create inverted index, create front-end to use this index for searching through crawled documents and performing some benchmark comparison for speed and relevance of result.

But, due to some difficulties during implementation phase, I could not finish the project as I planned. There was difficulty with getting to read archive files at initial stage and then issues with html parser towards middle of the semester.

Actual work done during Spring 2011 semester:

In the first stage towards creating inverted index, I needed to use a reader for the internet archive files. After trying to use java's implementation for GZIPInputStream to read compressed archive files for three weeks, I figured out with help of Dr. Chris Pollett, my project advisor, during one of our project meetings that there is a bug in implementation of GZIPInputStream class in java. It could not handle concatenated documents in gzip file very well. So, I decided to use Internet Archive's implementation of ARCReader class in Heritrix project as my utility for reading ARC files. I retrieve all documents from archive file one by one using iterator and check for type of the file. If it is html file, I parse the document using SAX Parser and extract words from html tags like title, h1, h2 and anchor tag and store that in following format:

doc_id –> word1, word2

doc_id –> word1, word2, word3

After parsing all html files in whole archive file, I use this initial phase result to produce word dictionary stored in Hash Map with following format:

Word1 –> doc1, doc2, doc3…

Word2 –> doc1, doc2

I transform the Hash Map data into regular text file once I build whole dictionary in Hash Map. After processing the whole archive file, I begin processing other archive file in the job list.

There is one more stage while building inverted index. There should be page rank associated with all documents to generate relevant result upon search query from user. I am in middle of implementation of page ranking algorithm.

Conclusion:

During this semester, I worked on archive reading, html parsing, word dictionary building and part of page ranking. Due to certain difficulties during project, I could not complete all my deliverables for this semester.

Roadmap Towards Finishing the Project:

To proceed towards completion of project, I need to finish the remaining implementation of page ranking for documents and finish with inverted index and helper class to access this index for producing results. Once, inverted index and helper functions are ready to use, I will develop JSP search interface. This search interface will take search query from user and search through the index for producing search results with help of the helper class containing access methods for inverted index. I also need to compare the search results with search results of other popular search engines for speed and relevance.

References:

[1] Büttcher, S., Clarke, C. L., & Cormack, G. V. (2010). Information Retrieval: Implementing and Evaluating Search Engines. MIT Press.

[2] Bug ID: 4691425 – Retrieved from Sun Developer Network Web Site: http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4691425