Vinay Anantharaman

vanantha@calpoly.edu

9/25/2007

Senior Project Proposal

Problem

The local search engine for calpoly.edu, powered by Google, uses the Page Rank algorithm for determining the quality of page. It works quite well on the global internet because, of the massive amounts of data they can store and effectively retrieve. However, within site specific searches new pages have very poor initial rankings and rarely come to the top of a search result. This causes problems for pages which are time-sensitive such as, a course website intended for the current quarter. When a student searches for the course web site they are presented with an older page, and then will have to find the correct course web site themselves. If the local search engine used the web traffic logs then, it could determine the newer course website has higher traffic.

Solution

To provide accurate results for time-sensitive queries on a single-site search engine requires the combination of traditional page ranking mechanisms combined with web traffic log analysis. The search engine requires a site crawler, log miner, information retrieval systems, and a front-end with a query processor.

Site Crawler

The site crawler constructs a connectivity graph of the website, and stores the HTML documents for indexing.

Log Miner

The log miner performs the three functions:

- Cleans the logs of web robots from other search engines, image requests, and failed page requests.
- Adds weights to the connectivity graph created by the site crawler.
- Computes the LPageRank (Log Page Rank).

The weights for the site graph are determined by analyzing the sessions for the users. A session consists of a user entering the site and making subsequent request in a fixed set of time. Each request from one page to the next adds a weight to edge between the pages. After all sessions are analyzed, edge normalization occurs by the total sum of all the weighs on outgoing edges of each node. Finally the computation of LPageRank executes.

Information Retrieval

The pages retrieved from the site crawler require indexing for quicker retrieval. The log miner's result must be paired with the indexes so it can be used to determine which links to show.

Front-end with Query Processor

The front-end presents a search bar for users and the query processor parses the request and executes the search. The results are displayed by the front-end.

Schedule

I intend in the first half of fall quarter to write requirements for the web crawler, and log miner. The implementation will last till the end of winter quarter. If another student works on the project, then in spring the other student and I will work on testing and integration of the search engine. Otherwise, I will work on the information retrieval and the front end for the web search engine.

Meeting minimum Criterion

The project meets the minimum criterion of independence, background research, and creativity.

Independence

I will write the requirements for the two components of the search engine and implement the requirements.

Background Research

A fair amount of research is required for implementing the log miner and page rank algorithm. For instance, I must find out what the general pattern of logs look like for the web traffic logs at Cal Poly. The algorithms to analyze the web traffic logs require research to provide the most accurate metrics of how users traverse the site.

Creative

The implementation is open-ended, and for an efficient and elegant solution will require my creativity and effort. The idea of combining a web traffic log analyzer, which exists as standalone apps, with a page ranking of algorithm is unique and novel.