CS221 Project 3 Milestone 3

JUN GUO, ID: 83322637
ZHEN CHEN, ID: 85282960

March 18, 2017

1 Search Engine Improvement

In this stage, we improve the search engine by three aspects: (1) We improve the relevance of the top 5 results returned by our search engine; (2) The search engine is refined to be faster; (3) The user interface is redesigned to be more beautiful. In what following, we introduce the strategies utilized in our search engine.

1.1 Relevance Improvement

It goes without saying that the relevance of the returned results is the key issue in search engine. To improvement the relevance of the returned results, we revise the indexer and save more information into the index. At first, we read the given web pages and store the full-capital keywords, such as REST. The reason why we add this new function is that a full-capital word can be an abbreviation of multiple words and users may enter the abbreviation as a query. Secondly, we find that Google's search engine returns some web pages in which the query keywords only appear in the urls. Therefore, we read the the urls in the web pages and extract the terms from the corresponding urls. Third, we calculate the page rank according to the directed graph generated by the given web pages. Moreover, we check the positions that the query keywords appear in the top-100 related web pages and justify if the keywords appear in order. Totaly, we take 7 measures, **cosin**, **isHead**, **internalLink**, **outLink**, **pageRank**, **capitalCheck**, and **positionCheck** into the scoring. A linear function is applied to combine these 7 measure, i.e.,

$$\begin{aligned} \textbf{score} =& a * \textbf{cosin} + b * \textbf{isHead} + c * \textbf{internalLink} + d * \textbf{outLink} + e * \textbf{pageRank} \\ &+ f * \textbf{capitalCheck} + g * \textbf{positionCheck}. \end{aligned} \tag{1}$$

Observed that there are a great number of duplications in the given web pages, we remove the duplicated pages according to the head of the page. To evaluate the search engine's relevance performance, we calculate the DCG@5 after searching a query.

1.2 Speed Improvement

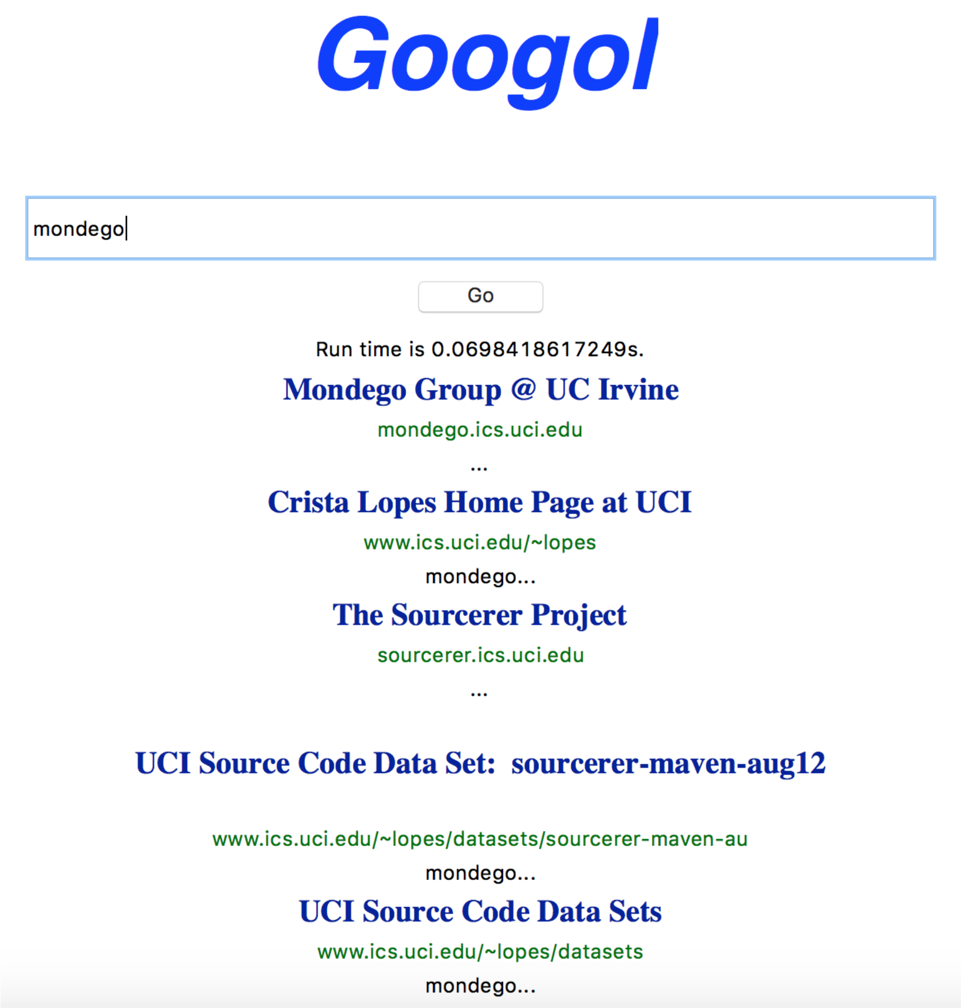
To reduce the time spend on searching, we decease the index file size that the search engine need to read when it receives a query. To achieve this goal, we **divide the index according to its first three alphanumeric characters and save them into different files**. Thus, the search engine only need to read several small index fils that related to the query. Moreover, **our scoring process has two steps**. In the first step, we retract the top-100 web pages according to 5 measures. In the second step, we sort the top-100 web pages according to all 7 measures. Thus, only 100 web pages requires to check the position which consume much time. And the time spend on sorting (whose time complexity is $O(nlogn)$, where n is the number of web page candidates) is also decreased because of the small web page candidates set. According to out observation, the search speed is greatly improved after applying this method.

1.3 UI Improvement

To make the user interface more beautiful and readable, we use different fronts and colors to show the head of the result, the url of the result, and the snips of the result. The head which can be clicked is denoted by bold blue. The corresponding url and the snip are, respectively, denoted by green and black.

2 Results

Similar to Google's search engine, we return not only top-k relevant urls but also corresponding heads and snippets with different colors and fronts. Fig. 1 illustrates the interface when we search the query *mondego*. To evaluate the search engine's performance, we also show the running time, which can be found in Fig. 1.



Run time is 0.0698418617249s.

Mondego Group @ UC Irvine

mondego.ics.uci.edu

...

Crista Lopes Home Page at UCI

www.ics.uci.edu/~lopes

mondego...

The Sourcerer Project

sourcerer.ics.uci.edu

...

UCI Source Code Data Set: sourcerer-maven-aug12

www.ics.uci.edu/~lopes/datasets/sourcerer-maven-au

mondego...

UCI Source Code Data Sets

www.ics.uci.edu/~lopes/datasets

mondego...

Figure 1: Search Engine Interface

Fig. 2 illustrates the normalized **NDCG@5** scores before and after the improvement. The blue line shows the normalized **NDCG@5** at the different positions when only cosin is used to calculate the score. The red line shows the normalized **NDCG@5** at the different positions when only cosin and isHead is used to calculate the score. The green line shows the normalized **NDCG@5** at the different positions when all 7 measures are used to calculate the score. After adding the improvement strategies, the total **NDCG@5** for the 10 given queries is increased from **0.048** to **0.575**. Consequently, the new search engine definitely outperforms the old one.

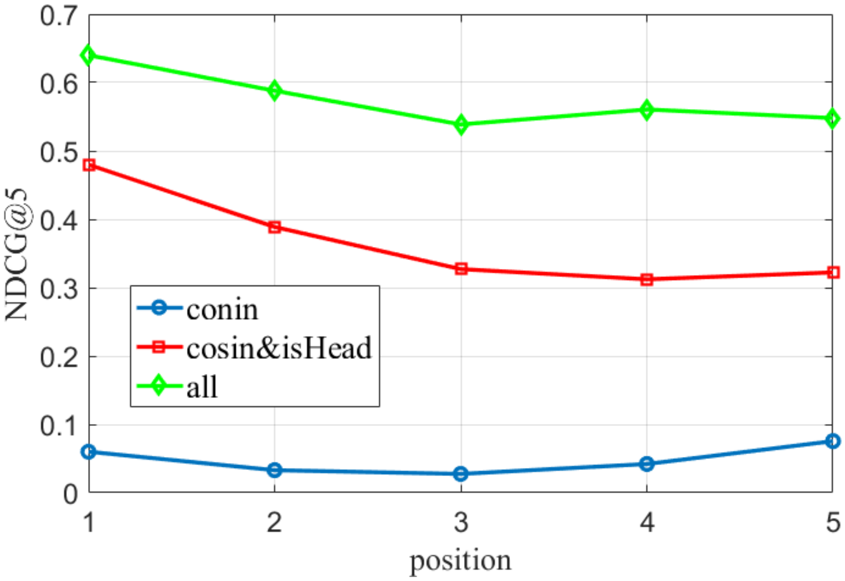


Figure 2: NDCG@5