

CS 221 – Project 3 Milestone 1

TEAM: Priyanka Ravi (33246700) and Rahul Sridhar (41608676)

APPROACH: We have taken a leaf out of Google's book to create the inverted indices. Three types of indices have been created – one for the entire text content, one for fancy hits alone (special HTML tags, listed in the table below), and one for plain hits (all other tags). The first index will be used to benchmark and fine-tune the performance of the search engine using the other two indices. TF-IDF scores are stored in a separate index while other information such as token position, font, etc. (more details below) are stored in a different index. Each type of index has been created and stored batch-wise based on the first character of the token; different buckets have been used for this – (a-f), (g-l), (m-r), (s-z) and everything else (numbers, special characters, etc.). The indices are then stored on disk as JSON objects. The sizes and #tokens shown below are the cumulative sum of sizes and #tokens of all the above bucketed sets of indices.

STATISTICS:

Document validity (based on the is_valid() function from our previous project):

Valid – 32,525 (indexed)

Invalid – 4,972 (not indexed)

Note – there was one duplicate link

Document type categorization:

<i>Extension type</i>	<i>Number of documents</i>
html	25,792
php	1,196
txt	1,464
htm	2,040
Miscellaneous	2,033
Total	32,525

<i>Index type</i>	<i>Tags</i>	<i>Information stored</i>	<i># Tokens in index</i>	<i># Tokens in tf-idf index</i>	<i>Size on disk (in KB)</i>	
					<i>Index</i>	<i>tf-idf</i>
Overall	Entire text	Document id, term frequency, token position (and tf-idf separately)	1,721,781	1,721,781	1,150,000	343,400
Fancy	title, h1, h2, h3, h4, table, th, b, a	Document id, font size, tag name, html tree path, case, token position, term frequency (and tf-idf separately)	90,158	1,673,291	325,400	308,000
Plain	All tags except the ones above		1,622,923		2,857,100	

Enhancements to be performed:

Our next few steps with respect to indexing are:

1. Index the content in meta tags
2. Tokenize and index the text in URLs
3. Perform data encoding (compression)