CS221: Information Retrieval

Report: Project 3 Milestone 1

Joel Fuentes (95686428) and Han Ke (62087924)

February 21, 2015

Inverted Index Implementation

The implementation of the index module (inverted index) was built by using a B-Tree data structure stored on disk. The Key value used in this index were the unique *terms* retrieve from the websites in the previous part of the project.
It is important to remark that the B-Tree structure is not built in main memory due to the big amount of memory that the set of website implies. To solve this problem a set of files on disk was used to create a pre-processing with the information needed to build the B-Tree in a second step. These file store all the information needed to create the inverted index by terms and documents (URLs).
In short, the task of indexing consists in:

1. Reading the information from crawled websites by parsing it and storing the result in files on disk. The general structure for each term on file is:

   $term_1|document_1\ position_1\ position_2|document_2\ position_1\ position_2$
   ...
   $term_i|document_j\ position_1\ ...\ position_k$
   To avoid dealing with one big file with the whole information and the likely out of memory problem, the files are split according to the first letter of each term. For instance, the first file stores terms stating with a until d in the alphabet.

2. Creating the inverted index on disk whose keys are the terms and values the set of documents. This task is done by reading the different files, one term at a time and inserting it into the B-Tree.

To store the B-Tree on disk, the same database was used as the previous part of the project and it is BerkeleyDB.
Different classes and methods were created for this part of the project:

- *InvertedIndexDB* (persistence package): This class has the objective of maintaining the inverted index on disk using BerkeleyDB. It provides methods to put and get terms from the index.

- method *Utilities.computeWordWithPositions*: This processes a list of strings and returns a hash map with unique terms and their positions in the list.

- *DocInvertedIndex*: This class represents a document (URL) that belongs to a term in the inverted index. It also has a list of positions where the term is located in the document.
- *TermInvertedIndex*: This class represents a term in the inverted index. It has a list of documents where the term is found.
- IndexBuilder: This class has a static method whose goal is to build the inverted index on disk.
- IndexController: This class has the main method and provides a menu with the following options:
  - Build Index for ics.uci.edu (It may take a long time)
  - Compute Deliverables
  - Execute query (beta)

From the menu presented in the program, it can be seen that the query engine was implemented but in a beta version. It works but with only one term at a time, which is a good advantage to the future part of this project.

Deliverables

The set of deliverables requested for this part of the project can be obtained by running the program. The following section presents an extraction of the index characteristics:

```
Index Details
Total number of documents: 51138
Total number of unique words: 199932
Total space of index on disk
  File    Size (KB)  % Used
--------  ---------  ------
0000005f       9202       0
00000060       9586       0
00000061       9758       0
00000062       9762       0
00000091       9153       0
00000092       9415       0
00000093       9018       0
00000094       9730       0
00000095       9765       0
00000096       9765       0
00000097       9762       0
00000098       9707       0
00000099       9454       0
0000009a       8975       0
0000009b       9757       0
0000009c       9591       0
0000009d       9762       0
0000009e       9746       0
0000009f       9740       0
000000a0       9739       0
000000a1       9765       0
000000a2       9759       0
```

```
000000a3       9763      53
000000cb       5749      98
000000a4       9760     100
000000a5       9756     100
000000a6       9627     100
000000a7       9198     100
000000a8       9397     100
000000a9       9752     100
000000aa       9765     100
000000ab       9763     100
000000ac       9595     100
000000ad       9748     100
000000ae       9712     100
000000af       8120     100
000000b0       9765     100
000000b1       9745     100
000000b2       9115     100
000000b3       9757     100
000000b4       9499     100
000000b5       9765     100
000000b6       9758     100
000000b7       9751     100
000000b8       9711     100
000000b9       9640     100
000000ba       9720     100
000000bb       9761     100
000000bc       9745     100
000000bd       9702     100
000000be       9686     100
000000bf       9748     100
000000c0       9719     100
000000c1       9720     100
000000c2       9764     100
000000c3       9478     100
000000c4       9257     100
000000c5       9745     100
000000c6       9759     100
000000c7       8863     100
000000c8       6592     100
000000c9       9568     100
000000ca       9763     100
 TOTALS      597741      64
(LN size correction factor: NaN)
```

In addition, an example of the query engine is presented in the following segment. Each one of the results presents its URL, number of matches in the document, TF-IDF and the term position in the document. The term used for this demonstration is **mondego** by using the entire information retrieved from ics.uci.edu:

```
***********************************
****      Query     ****
***********************************

Enter the word to search: mondego
mondego: 19 results
  http://mondego.ics.uci.edu/  3 matches TF-IDF=5.94 [ 35 39 296 ]
  http://mondego.ics.uci.edu/datasets/  1 matches TF-IDF=4.02 [ 30 ]
  http://mondego.ics.uci.edu/datasets/wikipedia-events/  1 matches TF-IDF=4.02 [ 14 ]
  http://mondego.ics.uci.edu/datasets/wikipedia-events/files/  1 matches TF-IDF=4.02 [ 22 ]
  http://sdcl.ics.uci.edu/2012/05/calico-for-the-mondego-group/  1 matches TF-IDF=4.02 [ 10 ]
  http://www.ics.uci.edu/community/news/notes/  2 matches TF-IDF=5.23 [ 340 5043 ]
  http://www.ics.uci.edu/community/news/notes/index  2 matches TF-IDF=5.23 [ 340 5043 ]
  http://www.ics.uci.edu/community/news/notes/index.php  2 matches TF-IDF=5.23 [ 340 5043 ]
  http://www.ics.uci.edu/community/news/notes/notes_2013.php  1 matches TF-IDF=4.02 [ 3973 ]
  http://www.ics.uci.edu/~djp3/classes/2006_03_30_ICS105/  1 matches TF-IDF=4.02 [ 649 ]
  http://www.ics.uci.edu/~djp3/classes/2006_03_30_ICS105/Resources/AnteaterIdol.html  2 matches TF-IDF
  http://www.ics.uci.edu/~djp3/classes/2006_03_30_ICS105/index.html  1 matches TF-IDF=4.02 [ 649 ]
  http://www.ics.uci.edu/~kay/courses/i141/hw/asst3.html  1 matches TF-IDF=4.02 [ 47 ]
  http://www.ics.uci.edu/~lopes/  2 matches TF-IDF=5.23 [ 31 125 ]
  http://www.ics.uci.edu/~lopes/datasets/  1 matches TF-IDF=4.02 [ 270 ]
  http://www.ics.uci.edu/~lopes/datasets/Koders-log-2007.html  1 matches TF-IDF=4.02 [ 265 ]
  http://www.ics.uci.edu/~lopes/datasets/SDS_source-repo-18k.html  1 matches TF-IDF=4.02 [ 335 ]
  http://www.ics.uci.edu/~lopes/datasets/index.html  1 matches TF-IDF=4.02 [ 270 ]
  http://www.ics.uci.edu/~lopes/datasets/sourcerer-maven-aug12.html  1 matches TF-IDF=4.02 [ 348 ]
```