FinalIR

Nasrin Baratalipour, Christopher Kahn

November 2014

1 Building a Information Retrieval System

In this project we used Apache Nutch for crawling the web pages and Apache Solr for indexing the crawled pages.
Apache Nutch is an open source web crawler that by using it, we can find web page hyperlinks in an automated manner. It creates a copy of all the visited pages for indexing and searching. That's where Apache Solr comes in. Solr is an open source full text search framework, with which we can search the visited pages crawled by Nutch.

1.1 Crawling

Apache Nutch provides a complete set of features you commonly need from a crawler. Nutch offers features like politeness, robustness and scalability (Nutch runs on hadoop), quality (you can bias the crawling to fetch "important" pages first).
Apache Nutch is very easy to use and also it reduces lots of maintenance work, like checking broken links.

1.2 Indexing

We used Solr for indexing the crawled Web pages. Solr is powered by Lucene, a powerful open-source full-text search library.The relationship between Solr and Lucene, is like that of the relationship between a car and its engine, which means Solr is based on Lucene plus an interface extra than it.

When a document is indexed, its individual fields are subject to the analyzing and tokenizing filters that can transform and normalize the data in the fields(content, url, ...). For example — removing blank spaces, removing html code, stemming, removing a particular character and replacing it with another.

Our used tokenization performs as follow:

1. Tokenizing with whitespace
2. Discarding stop words

3. Spliting words into subwords with delimiters

4. Case folding

1.3 Searching

At query time as well as indexing time we need to transform the query text, we did similar operations as above.
For retrieving and ranking the Web pages, Lucene combines Boolean model (BM) of Information Retrieval with Vector Space Model (VSM) of Information Retrieval. In VSM, documents and queries are represented as weighted vectors in a multi-dimensional space, where each distinct index term is a dimension, and weights are Tf-idf values. documents "approved" by BM are scored by VSM.
VSM score of document d for query q is the Cosine Similarity of the weighted query vectors V(q) and V(d):

$$cosine - similarity(q, d) = V(q) \times V(d) / \mid V(q) \mid\mid V(d) \mid$$

Where $V(q) \times V(d)$ is the dot product of the weighted vectors, and $\mid V(q) \mid$ and $\mid V(d) \mid$ are their Euclidean norms.

2 Program design, Interface, and Results Presentation

We chose to implement a simple web interface to allow users to make queries against our index, and to present our results. The program is organized as a combined Java and Scala program using the simple build tool (SBT). This allowed us to write the core parts of our search engine in Java while taking advantage of the lightweight and performant Scala web server library called Spray.

When the server starts up, it opens a connection to Solr and then starts listening on port 12345 for HTTP connections. It serves a web page with our search interface, which sends queries via background javascript requests to the server. Results are returned back to the browser client in Javascript object notation (JSON) where they are converted into HTML. We took advantage of Facebook's "react.js" view library to render the page and Twitter's "Bootstrap" css library to give it a clean look.

See the build.sbt and project/plugins.sbt files in the project archive for our complete external dependencies.

3 Questions

3.1 Which is the most positive Department in ENCS at Concordia?

We found that two departments were both very positive with close sentiment scores. Which one comes out on top of the list depends on the method of comparison. The top two departments are the department of Building, Civil and Environmental Engineering, and the department of Mechanical and Industrial Engineering.

Considering solely the average sentiment score, then Mechanical's score of 1.2738 makes it the most positive department, over BCEE's 1.2682. BCEE has a sigificantly higher number of positive documents, however, with 56.5% of its results being positive over Mechanical's 48.7%. BCEE also had no negative documents while Mechanical cannot claim the same. Thus we can also consider BCEE to be more positive. The remaining four departments all had very neutral sentiment scores with far fewer positive documents.

3.2 Is Computer Science and Software Engineering more positive or less positive than Electrical and Computer Engineering?

The Computer Science and Software Engineering department is less positive than the Electrical and Computer Engineering department. Computer science has an average sentiment score of 0.719, with 15.9% of its documents being positive, the remaining 84% are considered neutral. Electrical, on the other hand, had an average score of 0.7973, with a full quarter (exactly 25% of crawled pages) of its pages are positive, the remaining 75% neutral.

3.3 Rank the departments in ENCS by sentiment of their web documents

The departments, ranked in order of sentiment are presented here. In parentheses follows the average sentiment score, followed by the proportion of documents that are positive, neutral and negative.

- Building, Civil and Environmental Engineering (1.2682, 56.5/43.5/0)
- Mechanical and Industrial Engineering (1.2739, 48.7/50.63/0.63)
- Institute for Information Systems Engineering (1.0689, 39.24/58.86/1.9)
- Electrical and Computer Engineering (0.7973, 25/75/0)
- Computer Science and Software Engineering (0.719, 16/84/0)
- Centre for Engineering and Society (0.5867, 33/66/0)

3.4 Classify the departments in ENCS with a three way classifier into positive, negative, and neutral

Out of 1087 documents crawled from the starting point (http://www.concordia.ca/encs), we found that the average sentiment score is 1.1494 and the classification as follows:

- Positive: 47.65%
- Neutral: 51.98%
- Negative: 0.37%

3.5 Classify the additional mystery page results the same way and compare its dominant sentiment

McGill's Electrical and Computer Engineering department presented a stark contrast from Concordia's writing. Its pages are considerably less positive than its counterpart at Concordia. ECE had an average sentiment score of 0.4088, contrasting with Concordia ECE's 0.7973. Of McGill's ECE department's pages, only 10% were classified as positive, and a surprising 20.3% are classified as negative, giving it more negative pages than all of Concordia's ENCS faculty.

We were not be able to crawl http://www.cs.mcgill.ca/. We are suspect that it is because of robot.txt, which forbids crawlers from crawling Mcgill Computer Science pages. It is worth mentioning that apache Nutch respects to this probhibition from the Web pages.

4 Additional Questions

4.1 What was the hardest step

We found that the project as a whole was both challenging and rewarding, in particular the step of adding the sentiment scores to our results.

4.2 Index size

The Concordia index, starting from ENCS page, has a total of 1087 documents.

4.3 Observations and Learning

In addition to learning the theoretical aspects of search, indexing and classification, as also learned a great deal about how this is put into practice in online systems. Using these tools–Nutch, Lucene, Solr, and a fairly simple web server– it is possible for anyone to setup a small but effective search engine to index and search some subset of documents on the Internet.