Notes

Matthew Morgan, May 2018

# Resources

The following is a list of written resources referenced for the rest of the documentation/research. Resources are listed by the date of publication, the title, author(s), and the location of the document. If a hyperlink is available, it will be provided; otherwise, files will be made available with the titles of the documents as their filename.

|  |  |  |  |
| --- | --- | --- | --- |
| Research References | | | |
| Date | Document Title | Document Author(s) | Location |
| 30 Apr. 2018 | *Information Retrieval: Libraries and Frameworks* | William Smith | Resources |

## Research References

### Information Retrieval: Libraries and Frameworks, William Smith

This 11-page PDF document explores Apache Solr, ElasticSearch with Kibana, and Indri to evaluate their IR effectiveness on the Ohsumed corpus. It investigates the interfaces of these three sets of software as well as what input each accepts.

# Software and Corpa

## Corpa

### Ohsumed Collection

The Ohsumed collection is available at <https://trec.nist.gov/data/t9_filtering.html>. This version includes approximately 348,000 documents, all from the MEDLINE database. The content of the collection is from 270 medical journals over the timeframe of 1987-1991. *(Some abstracts are truncated at 250 words, and some have no abstracts.)* The following restrictions apply:

* The data will not be used in any non-experimental clinical, library, or other setting
* Any human users of the data will explicitly be told that the data is incomplete and out-of-date

In the analysis of IR implementations, a partial of the Ohsumed collection was used. It consisted of approximately 50,000 abstracts from the year of 1991 on the subject of cardiovascular disease. The abstracts are all plain-text documents, where the first line is the **title**, and the rest of the file was the **content** itself. (Smith)

### Cranfield Collection

The Cranfield Collection is available at <http://ir.dcs.gla.ac.uk/resources/test_collections/cran/>. The Cranfield Collection is a series of 1,400 documents, specifically formatted to isolate document titles, identifiers, authors, bibliography, and abstract content. The documents are all scientific abstracts, and all documents are contained in a single file. The Collection also includes a series of query results and relevancy judgements. *(There is a copy of the collection with only 200 documents.)*

The following notes apply:

* The Collection has a few entries that are blank – that is, they are only skeletons of the keywords meant to assist in parsing the collection. Known IDs are: 471 and 995.
* A few of the documents, in the corpus, have duplicate keywords, such as ‘.W’ or ‘.A’, confusing which is the content of a document ID. This is known to occur in four documents.

### [WIP] Reuter’s

…

<http://ir.dcs.gla.ac.uk/resources/test_collections/>

<http://disi.unitn.it/moschitti/corpora.htm>

### Gutenberg Mini-Corpus

The Gutenberg Mini-Corpus was a corpus provided by Dr. Venkat Gudivada to the participants of the Information Retrieval course at East Carolina University for Spring of 2017. This corpus consisted of 18 textual documents with no particular formatting, and was utilized for a programming assignment. *(Due to the nature of the corpus, minor modifications have been made to it to conform to the Cranfield Collection’s file format.)*

## Software Programs (Overview)

(A more thorough investigation, or set of notes, for these software programs is available in the [“Software Programs (Full)”](#Software Programs (Full)) section.) The following pieces of software were explored, or otherwise utilized, during accumulation of research content:

1. **Apache Lucene**: A set of packages that provides the fundamental components of an IR system: tokenization, indexing, and searching (term, phrase, and Boolean)
2. **ElasticSearch**: RESTful server application that indexes JSON file(s) for searching
   1. **Kibana**: Provides a powerful user interface for ElasticSearch to configure and interact with indices

## Software Programs (Full)

### Apache Lucene

#### About

Apache Lucene is an open-source API. It provides an array of features, via Java classes, that include different methods of an IR system implementation. The API is constructed of a few different components, and only accepts **plain text input**. Information about Lucene can be located here: <https://lucene.apache.org/>. *(Lucene has more than a Java implementation, such as Apache Lucy for C, but the most well-known implementation is the Java one.)*

Lucene provides the following components in a package structure:

* **Analysis**: Performs the tokenization process, converting plain text into tokens that are index
* **Codecs**: Allows manipulation of index encoding and storage (postings list, document values, term vectors, etc.)
* **Document**: Represents a document for indexing and searching
* **Index**: Reads, and writes, documents to and from storage
* **Search**: Provides different search methodologies, such as term, phrase, and Boolean queries
* **Util**: Data structures and utility classes, such as PriorityQueues

Lucene also provides various, extraneous utilities, including but not limited to: query parser implementations, codecs, and query implementations. Lucene does **not** provide functionality for parsing documents into plain-text format though.

### ElasticSearch (Kibana)

#### About

Elasticsearch is software that implements the Lucene API. It processes **JSON** **file** **input** to create indexes, and allows querying of those indexes. Kibana allows the manipulation and querying of those indexes, using Lucene’s query syntax, in a uniform interface. ElasticSearch is available at <http://www.elastic.co/products/elasticsearch> with Kibana available at <http://www.elastic.co/products/kibana>.

#### [WIP] Installation & Execution

**Pre**-**requisites**

* The Java Runtime Environment must be installed on the machine that will run ElasticSearch, with the JRE appended either to the PATH or set to the JAVA\_HOME environment variable
* The corpa/files that will be utilized with ElasticSearch should be in JSON format. *(ElasticSearch does not provide utilities to generate JSON files for you.)*

**Execution**

1. ElasticSearch
   1. Download and unzip ElasticSearch. Then, open a terminal (or Command Prompt) at that directory
   2. Run **bin/elasticsearch** (Unix) or **bin\elasticsearch.bat** (Windows)
   3. Run **curl http://localhost:9200/** (Unix) or **Invoke-RestMethod http://localhost:9200/** (PowerShell on Windows)
2. Kibana
   1. Download and unzip Kibana. Then, open a terminal (or Command Prompt) at that directory
   2. Open **config/kibana.yml** in an editor, and set **elasticsearch.url** to point at the ES instance
   3. Run **bin/kibana** (Unix) or **bin\kibana.bat** (Windows)
   4. Open a web-browser and go to the URL **http://localhost:5601**
3. … (Execution commands for adding corpa)

## [WIP] srcML

…

## [WIP] Luke (LukeFX)

…

## [WIP] Indri

…

# Research Questions

## [WIP] Heaps’ Law

**Notes:**

* The typical equation for Heaps’ Law is as follows:
  + M The vocabulary size that will be found
  + k A constant factor, usually assumed to be between 30 and 100, inclusive
  + T The number of tokens in the collection
  + b A constant factor, usually assumed to be around 0.5
* Least squares fitting
  + …

**Questions:**

* What proportion of a text collection must be read for 90% of its vocabulary to be found? *(Beta may be assumed to be equivalent to 0.5. To solve this problem, the value of k is not necessary.)*
  + …
* Does Heaps’ Law hold for the Gutenberg Minicorpus? *(Estimate this by gathering data points (x,y), where x is the number of tokens looked at, and y is the number of unique tokens looked at, for each word processed. Remember to transform the data (log-log) and use least squares to determine k and beta.)*
  + …