Ryan Cedzo

University of Illinois at Urbana-Champaign

Text Information Systems

November 5, 2021

Apache Lucene

When building a website or application, very often a developer needs to add a search function to enable users to search across the site. Many search algorithms can become extremely complicated and hard to optimize. With Apache Lucene, developers are able to add search capabilities directly into their website or application. Apache Lucene is a full-featured text search engine library that is written in Java. The software is open source and free to download (*Apache Lucene Core*). Apache Lucene contains many different ranking algorithms and comes complete with many useful features to a developer that needs to add a full-text search to their application. This short paper aims to discuss the basic workings and features of Apache Lucene as well as some comparisons to other similar software on the market.

In general, the indexing for Lucene is an inverted index. In the inverted index, Lucene indexes specific terms, then maps these terms to specific documents. The scoring function used by Lucene essentially counts the number of documents all the search terms point to and scores the document accordingly. If more search terms all point to the same document, then this document will most likely be seen as more relevant (Md. Rezaul Karim). The scoring function used by Lucene is a combination of two different scoring functions, the Vector Space Model of Information Retrieval and the Boolean model (*Apache Lucene Core*). Taking a deeper dive into the documentation, the breakdown of different segments of the scoring function can be found. In

terms of the VSM model part of the scoring function, it uses cosine similarity to determine the score.

In addition to the basic idea of Lucene as an inverted index that scores documents by their relationship to the specified search query, Lucene offers many other features as well. During querying, Lucene allows many different query types, such as phrases, wildcard, proximity, and range queries (*Apache Lucene Core*). This provides the user with more possibilities to personalize their queries. Lucene also offers simultaneous updating and searching, joining and grouping results, and also pluggable ranking models, including Okapi BM25. However, even though there are many features included within Lucene, it also has its drawbacks. These drawbacks are seen in the fact that Lucene is not a complete application, but rather an API that can be utilized within an application. This leads to some of the main comparisons between Lucene and some other popular search algorithm software.

Solr is another popular searching software. However, Solr is a complete application that can be used as-is. This is different from Lucene, as Lucene is an API that requires to be implemented into an existing application. Because of this, Solr has many more functionalities compared to Lucene and contains a much better UI. Lucene must be used by programmers or those that have enough programming knowledge to implement while Solr allows the use of those without any programming knowledge. Lastly, Solr is also a web application, which differs from Lucene's API. However, Solr actually implements Lucene for its searching and indexing capabilities. Therefore, it makes sense that Solr is more extensive, as it builds upon the foundation laid down by Lucene (*Lucenetutorial.com*).

Another similar searching software is Elasticsearch. Elasticsearch is similar to Solr in that it uses Lucene and builds upon its foundation. Elasticsearch also relates to Lucene in that it keeps

the idea of being an open-sourced searching software. However, Elasticsearch is still a web server that assists in searching and indexing capabilities while Lucene is essentially a Java library with a much more direct usability (*What is the difference…*). Therefore, Elasticsearch provides a much larger functionality than Lucene as it is a distributive system that implements Lucene directly.

Apache Lucene is a very well-used software that can be extremely helpful in implementing search and indexing capabilities to an application. Some software, such as Solr and Elasticsearch are built upon Lucene to allow the functions of Lucene to be used with better UI and added capabilities. On its own, Lucene can only be used by those with programming knowledge, as it is a base Java Library. It uses an inverted index and VSM to score documents and return a ranking of those documents in relation to a query. For a programmer that wants to implement a search and ranking function in their application, Lucene would be a good choice, as they can use the Java library directly with a wide range of possibilities for implementation. However, for a non-programmer or someone with less programming experience that wants an easier UI and implementation, Solr or Elasticsearch would provide an easier experience for searching, indexing, and ranking capabilities.

References

Apache Lucene Core. Apache Lucene. (n.d.). Retrieved November 4, 2021, from https://lucene.apache.org/core/.

Lucenetutorial.com. Lucene vs Solr - Lucene Tutorial.com. (n.d.). Retrieved November 4, 2021, from http://www.lucenetutorial.com/lucene-vs-solr.html.

Md. Rezaul Karim ·, Karim, M. R., 18, J., & Like (8) Comment . (2017, January 18). *Searching and indexing with Apache Lucene - DZone database*. dzone.com. Retrieved November 4, 2021, from https://dzone.com/articles/apache-lucene-a-high-performance-and-full-featured.

What is the difference between Lucene and Elasticsearch. NewbeDEV. (n.d.). Retrieved November 4, 2021, from https://newbedev.com/what-is-the-difference-between-lucene-and-elasticsearch.