Implementing State of the Art Ranking for Lucene

Google Summer of Code Project Proposal

David Nemeskey
Eötvös Loránd University
nemeskey.david@sztaki.hu

1 Problem Statement

Apache Lucene[1] is probably the most popular information retrieval library. It is commonly used as a baseline system at IR evaluation forums (see [2]). There are several open source projects based on Lucene, such as Solr[2] and elasticsearch[3].

Lucene employs the Vector Space Model (VSM) to rank documents, which compares unfavorably to state of the art algorithms, such as BM25. Moreover, the architecture is tailored specifically to VSM, which makes the addition of new ranking functions a non-trivial task. This also hinders wider adaption of Lucene by the IR community.

2 Project Proposal

This project aims to bring state of the art ranking methods to Lucene and to implement a query architecture with pluggable ranking functions. It shall consists of three main components.

Architecture Lucene already has a sophisticated query class hierarchy. The implementation will be based on the existing classes. However, the interfaces will be altered and made agnostic with regard to the ranking function used. The scoring formulas will be implemented in a new class hierarchy following the strategy design pattern.
For this to work, all information necessary for at least bag-of-words (BOW) ranking models will be made available to all implementations. These include tf, idf, document length and the average values thereof.

Methods The flexibility of the architecture will be proved by implementing three probabilistic ranking functions: BM25 [3], BM25F [4] and the DFR framework [1]. The current Vector Space Model will be adopted to the new interface as well.

Configuration An API shall be published that allows the user of the library to select the ranking method and configure its parameters. To maintain compatibility, the system will default to VSM.

Javadoc and wiki documentation shall be written for new and altered components.

I am working at the Data Mining and Search Research Group at the Computer and Automation Institute of the Hungarian Academy of Sciences. I have participated in last year's TREC and CLEF competitions. I am also a Sun Certified Java Developer.

[1] http://lucene.apache.org/

[2] http://lucene.apache.org/solr/

[3] http://www.elasticsearch.org/

References

[1] Gianni Amati, Cornelis Joost, and Van Rijsbergen. Probabilistic models for information retrieval based on divergence from randomness. *ACM Transactions on Information Systems*, 20:357–389, 2002.

[2] Anselmo Peñas, Pamela Forner, Álvaro Rodrigo, Richard Sutcliffe, Corina Forăscu, and Cristina Mota. Overview of respubliqa 2010: Question answering evaluation over european legislation, 2010.

[3] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. pages 109–126, 1996.

[4] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple bm25 extension to multiple weighted fields, 2004.