Information Retrieval: Intermediate Report

Elise Kuylen
Robin Verachtert

May 6, 2016

1 Progress

1.1 Data set

We had some problems with the datasets we specified in the project proposal. The first dataset (Jester) had no texts of the jokes, so it was impossible to use.
We still had the books dataset, which we would intersect with the data from Gutenberg. However the Gutenberg download did only contain text and titles, no ISBN numbers which we would use to intersect the datasets. So we had no data, we decided, rather than try to fix the Gutenberg dataset, that we would write a simple scraper to get information from Goodreads. This is a sort of social network for books. It contains a lot of books, with reviews and ratings of thousands of users. The parser is written in python, and has several flaws, most importantly, it is extremely slow, takes about 1h30 to parse 5 users and their books (100-300). The second problem is that after at most 1h30 the connection is refused by Goodreads. So the data we have is about 6 users. But we do have about 3000 books and their abstracts, which is a decent amount. It also has the advantage that Robin is one of the users in the dataset, so when we do recommendations he can judge the precision. Of this big dataset we use about 20 books to test the system.

1.2 Tokenization and indexing

When we had the dataset, we started on the basis of the project, namely tokenizing the data and creating an index. For this we used Lucene. Next we used the query analyzer from Lucene to test the index, and make sure that stemming was done.

2 Functionality to be added

2.1 Recommendation

The main part of this project is the recommendation, we will take a user and give a set of recommended books. There are several ways we might do this, the simplest way is to use every book the user has read as a query and return a subset of the results. Another way would be to concatenate all the texts of the books, and use that as a query.

2.2 Dimensionality reduction

We would like to use this as a way of learning latent factors in the data, and hopefully finding more relevant books. However, we have not yet found if this is supported by Lucene.

2.3 Validation

This is going to be the hardest part of the project, because we are not using a validated data set, and the hope is to return new books which could be of interest to the user. Precision can be done, by running the program for user Robin Verachtert, and let him decide which of the returned books are relevant. Recall is much harder, it is impossible to go through all 3000 books and decide which are relevant, so for that part we will have to research how other recommendation systems have done it.