Information Retrieval Project 2, Group 11

Michael Merki & Markus Greiner - December 2016

# Installation

See README.md how to install and run using sbt.

# Preprocessing / Tokenizing

# Scoring Models

Term-Based Model:

Our term-based model consists of two different scoring approaches. On the one hand we implemented the scoring based on tf-idf weights described in the lecture (lecture 4, slide 14). In order to be more robust against biases towards longer document we extended the standard tf-idf weighting scheme by using the augmented term frequency instead of the raw term frequency. We computed the augmented term frequency with the following formula: atf(w,d) = 0.5 + 0.5 \* tf(w,d) / max(tf(w',d) : t' ϵ d)

On the other hand we used the idea of the vector space model described in the lecture (lecture 4, slides 15 – 17) and computed the cosine similarity between the documents and queries represented as vectors.

Both scoring approaches can be found in the class TermBasedModel.scala.

Language Model:

As our language model we implemented the maximum likelihood estimation presented in the lecture (lecture 6, slides 5-8) and applied the proposed Jelinek-Mercer smoothing to get better estimates for small frequencies. Experimenting with varying the tuning parameter λ showed optimal precision, recall, f1 and MAP measures for λ = 0.2.

The implementation of the language model can be found in the class LanguageModel.scala.

# Challenges during development

…

# Training Data Performance

Evaluating the performance of our three scoring models with 100'000 training documents resulted in the following precision, recall, f1 score and mean average precision (MAP) metrics:

|  |  |  |  |
| --- | --- | --- | --- |
| Metrics | TFIDF | Cosine Similarity | Language Model |
| Precision | 0.40024999999999994 | 0.342 | 0.40474999999999994 |
| Recall | 0.4147382133995037 | 0.3548179374389052 | 0.4181252349800736 |
| F1 Score | 0.40544781382967565 | 0.34663095537529603 | 0.40954392122031474 |
| MAP | 0.27744256788107285 | 0.19819450627710983 | 0.2790647060168729 |

Note: For computation of recall and MAP we bounded the denominator by MIN((TP+FN),100)

# Approaches for Improving Training Data Performance

We implemented several extensions to the above described scoring models with the goal to improve the training data performance. These included:

* N-grams
* Query Expansion through Synonyms
* Query Expansion through terms from relevant documents

# Running Times

The comparison of running the three scoring models with and without the inverted frequency index resulted in the following average running times per query:

|  |  |  |  |
| --- | --- | --- | --- |
| Running Time | TFIDF | Cosine Similarity | Language Model |
| With inverted frequency index |  |  |  |
| Without inverted frequency index |  |  |  |

Note: For these experiments we used a Windows Machine with Intelcore i/ 4500u processor, 1.80 GHz with 8 GB RAM. The maximum heap space was set to 4 GB.