# Information Retrieval - Fall Semester 2015

Group ID: 11
ollel@student.ethz.ch
eholmer@student.ethz.ch
pmichal @student.ethz.ch

January 16, 2016

## Our retrieval system produced the following statistics:

The mean average precision was calclated using min(TP + FP, 100) in the denominator.

| | Term-based | Language-based | OkapiBM25 | Tf-Idf |
|---|---|---|---|---|
| Mean precision | 0.279 | 0.271 | 0.243 | 0.204 |
| Mean recall | 0.130 | 0.123 | 0.112 | 0.0890 |
| Mean F1 | 0.156 | 0.150 | 0.136 | 0.0538 |
| MAP | 0.286 | 0.268 | 0.223 | 0.195 |

## Execution instructions

The program takes three arguments:

- Path to file containing -separated queries (see *topics_final.csv*)
- Path to directory with zips
- Path to *qrels* file

Example to run the program with added memory:

```
scala -J-Xmx2g retsys.jar topics_final.csv tipster qrels
```

The program will produce two files for each model in the *output* folder, one with the retrieved documents and one with evalution scores. Also the *cache* folder will be used to store the calculated collection and document frequencies.

## Implementation details

- **Parsing documents:** We parse the given document collection in two passes. The first time we parse the whole collection to precompute the Document frequencies and the Collection frequencies. In the second pass, we calculate the Term frequencies for each document and calculate the term based score together with the language-based score. The top 100 results for each model are kept in a priority queue during the second pass.

- **Term-based model:** For the term based model, a scoring that measures the overlap between the query and the document is used. The overlap $O$ of a query $q$ and a document $d$ is defined as $O = |\{w|w \in q, tf(w;d) > 0\}|$, where $tf(w;d)$ is the count of word $w$ in document $d$. We also add a term which measures the normalized frequency of each word in the document. The frequency of the words in a query $f(q)$ is defined as $f(q) = \sum_{w \in q} tf(w;d)$. $f(q)$ is normalized by the square root length of the query $\sqrt{|q|}$ and the euclidian norm of the document $\|d\|^2 = \sum_{w \in d} tf(w;d)^2$. The final score is calculated as: $O + \frac{f(q)}{\sqrt{|q|}\|d\|}$.

- **Language-based model:** The language based model uses Jelinek-Mercer smoothing to interpolate between the estimated probability of a word in the query given a document $P(w|d)$ and the unconditional probability of a word $P(w)$. We need to approximate these probabilities by $\hat{P}(w|d)$ and $\hat{P}(w)$. This is done by setting $\hat{P}(w|d) = \frac{tf(w;d)}{\sum_{v \in d} tf(v;d)}$ and $\hat{P}(w) = \frac{cf(w)}{\sum_v cf(v)}$, where cf(w) denotes the collection frequency of a word. The probability that a word in a query belongs to a particular document is calculated as $P_s(w|d) = (1-\lambda)\hat{P}(w|d) + \lambda\hat{P}(w)$. And the score of a particular document is finally calculated as $\sum_{w \in q} log(P_s(w|d))$. We use $\lambda = 0.2$ as the smoothing parameter.

- **General remarks:** We do not use tf-idf scoring since it produced worse results than our current scoring function. We implemented the OkapiBM25 scoring[1], giving slightly worse results than our current term based scorer, but better than the tf-idf scorer. We also tried treating the title of each document separately, arguing that that a query-to-title overlap should be more relevant but due to time constraints and no significant increase in MAP this method was discarded. Stemming was not used since it provided little to no improvement, but slowed down the computations drastically.

[1] https://en.wikipedia.org/wiki/Okapi_BM25