1 system

All code is in the file `WebIndexing.scala`, this scala code runs in 4 steps (seperated by comments in the code), the 4 steps are as described below:

1.1 Preparing candidate terms from queries

First we aggregate all queries, tokenize them and (after stopwording) we get a list of unique terms in all queries. This list of query terms is about 100 in size.

1.2 Reading through documents and getting statistics

The second phase of the programme consist in passing through all documents, and for each document d, we calculate the term frequency of this document and update our statistics.

The statics include:

- `tf(w,d)`: the term frequency, notice we are only interested in terms `w` *in query terms*, in scala this is represented by a `HashMap[(String, String), Int]`
- `df(w)`: document frequency for term `w` *(in query terms)*, `HashMap[String, Int]` in scala
- `cf(w)`: collection frequency for term `w` *(in query terms)*, `HashMap[String, Int]` in scala
- `len(d)`: mapping from document to its length, `HashMap[String, Int]` in scala
- `N`: number of document in total
- `totalLen`: length of total document length

We will store there statistics in memory so that we can performe calcluations later on.

1.3 Indexing of documents

In this phase we do the indexing.

For each query, we go over all documents again (but this time only to look up hashmaps in memory instead of going through zip files), and calculate the term-based score and language-model score according to the equations described above. Then we maintain minimum priority queues of maixmum size 100 to store the revalant documents, so that at last we get 100 highest-scored documents.

We keep these lists in memeory for evaluation later on. Also in this step we have outputed the results to files.

1.4 Evaluation

In this step we do the evaluation.

We first read the content in `qrel.txt`, and construct the ground truth hashmap.

For each query we then go over our result, compare it with the ground truth, and calculate the evaluation mertics.

2 Training performance

	precision	recall	MAP
term-based	0.134750	0.058697	0.067742
language-model	0.087750	0.047216	0.035881

Table 1. Performance of our models on training data