**Algorithm 1**

**TF-IDF based model**

In this model I use the term frequency of different terms in the documents and inverse document frequency of the terms in the overall corpus of document(s). Then I normalize the two tf and idf scores using log base 20, and apply the formula;

Score = tf\*idf

The goal is to create one score for a query per document. So applying the formula to the training set documents yields the folder TF-IDF in the root folder of the project

**Algorithm 2**

**Dirichlet based model**

In this model I use the dirichlet smoothing technique to find scores of queries in documents. This is a probabilistic model to find scores. As it uses the following formula;

Score = N/(N+u) {\* probability of term in document} + u/(u+N) {\* probability of term in corpus}

The goal is to create one score for a query per document. So applying the formula to the training set documents yields the folder DIRICHLET in the root folder of the project.

Moreover, the scores are not very optimal because dirichlet is more commonly used for documents with smaller lengths and a large number of files, considering the average well over a large set of documents, Since the model adds a background probability for the terms to exist in the overall corpus of documents.