IRTM Assignment 3

Task 2) Jelinek-Mercer Smoothing $P(q|d) \propto \prod_{1 \le k \le |q|} (\lambda P(t_k|M_d) + (1-\lambda) P(t_k|M_c))$

$\lambda = 0{,}3$ $P(q|d_1) = [0{,}3 \cdot \frac{1}{8} + \underbrace{0{,}7 \cdot \frac{2}{16}}_{0{,}0875}] \cdot [0{,}3 \cdot \frac{1}{8} + 0{,}7 \cdot \frac{1}{16}] = 0{,}01056$

$P(q|d_2) = \underbrace{[\underbrace{0{,}3 \cdot \frac{1}{8}}_{0{,}0375} + 0{,}7 \cdot \frac{3}{16}]}_{0{,}125} \cdot [0{,}3 \cdot \frac{0}{8} + \underbrace{0{,}7 \cdot \frac{1}{16}}_{0{,}04375}] = 0{,}00547$

$d_1 > d_2 \Rightarrow$ document d_1 is more likely for the query

higher value of λ: document d_1 will be even more likely because this means documents containing all query words are having a higher probability.

lower value of λ: document d_1 is just a little more likely because the fact, that d_2 doesn't contain all query terms is not as important as with a higher value of λ.

Task 4)

In the vector space model ~~collections~~ documents and queries are represented by an tf-idf vector. To see how close a query is to a document we ~~calc~~ calculate and sort the angles between query and each document. This however only describes how similar a query is to each document but not how relevant they actually are since tf-idf weights ~~are based~~ only use the occurence of the terms (in the documents). On the other side a ~~vec~~ probabilistic ~~model~~ ranking takes into account how relevant a document is for a given query since the terms in the document vectors are weighted by their relevance.

To rank efficiently we could create a inverted index by storing the documents and their probabilities for how likely a document is relevant given a term. (exclude documents with $P(d)=0$)