# ASSIGNMENT-2
# DOCUMENT RE-RANKING

## Part-1: Pseudo-relevance Feedback with Rocchio's method

1. Pre-processing the data:

- Parsing: We use data collection and queries from TREC COVID track. The format and the description of the track is given at https://github.com/allenai/cord19/blob/master/README.md.
  Punctuation marks and symbols were removed from the text to clean the data. To increase the computation efficiency, stopwords from nltk.corpus were also weeded out. Stemming of words was done by PorterStemmer() of http://www.tartarus.org/~martin/PorterStemmer.

2. Data Storage:

- Words were stored in a dictionary which carried the data for term frequency, document frequency and document length.
- To calculate the idf of every term, we need to consider the document frequency across the entire collection. To enhance the space efficiency and to avoid parsing over a large data collection, only the top-100 relevant documents were collected; document frequency of a term was thus approximated as df*N/N_top100 where df=Number of times the term appears in the relevant document collection, N=Number of documents in the entire collection, N_top100= Number of relevant documents. It was observed that the above assumptions did not affect the overall performance of the ranking system.

3. Query vector:

- The vector consists of query frequency.

4. Rocchio Re-Ranking:

- The alpha parameter was tuned at 1, beta at 0.75 and gamma at 0.15.

5. BM-25 Modelling:
   - For efficient calculation of scores of every document, the BM-25 model was deployed.
   - w_(i,j)= tf_(i,j)*(1+k1)*w/(k1*((1-b)+b*dl(j)/dl_avg+tf_(i,j)), where w=log(N/df(i)); w_(i,j) represents the weight of a term i in a document j. It was observed that for efficient ranking, b should be set at 0.75 and k1 at 2.

| | nDCG @5 | nDCG@10 | nDCG@50 | MAP |
|---|---|---|---|---|
| After Rocchio Re-Ranking | 0.5374 | 0.5320 | 0.4771 | 0.0599 |
| Original file(Before Re-ranking) | 0.4844 | 0.4833 | 0.4424 | 0.0576 |

<u>Part 2: Relevance Model-based Language Modeling</u>

1. RM1- I.I.D. Sampling:
We assume that the query words and the words in relevant documents are sampled identically and independently from a unigram distribution. We pick a distribution M with probability P(M) and sample from it k+1 times. The probability of observing w with $q_1,q_2,...q_k$ =

$$P(w, q_1 \dots q_k) = \sum_{M \in \mathcal{M}} P(M) P(w, q_1 \dots q_k | M)$$

$$P(w, q_1 \dots q_k) = \sum_{M \in \mathcal{M}} P(M) P(w|M) \prod_{i=1}^{k} P(q_i|M)$$

Assumption: w and $q_1,q_2,...q_k$ are mutually independent once we pick a distribution M.

2. RM2- Conditional Sampling

$$P(w, q_1 \ldots q_k) = P(w) \prod_{i=1}^{k} \sum_{M_i \in \mathcal{M}} P(M_i|w) P(q_i|M_i)$$

Assumption: $q_i$ is independent of w once we pick a distribution M.
In this model, we are free to pick a separate $M_i$ for every $q_i$.

3. Unigram model with Dirichlet smoothing
To ensure the proper functioning of our model, we set the query prior P(q1,...qk)= $\sum_w$ P(w,q1,....qk)

Similarly, we set the P(w) as :

$$\sum_{M \in \mathcal{M}} P(w|M) P(M)$$

Dirichlet smoothing:

$$\hat{P}(t|M) = \frac{f_{t,d} + \mu \hat{P}_C(t)}{|D| + \mu}$$

It was found out that the most appropriate value for μ is dl_avg=average document length.
$f_{t,d}$= frequency of term t in document d.
$P_c(t)$= $f_{t,C}$/ N_top100, where $f_{t,C}$ is the language model built from the entire collection(in our case, collection of relevant documents).

For RM-1, we set P(M)=1/N_top100
For RM-2, we estimate the conditional probabilities of picking a distribution M, based on w:

$$P(M_i|w) = P(w|M_i) P(w) / P(M_i)$$

3. Query Expansion:

Weight $c_i$ was calculated as
log(((s+0.5)/(S-s+0.5)]/[(df-s+0.5)/(N-df-S+s+0.5))).
s= number of relevant docs that contain the word i.
S=Number of relevant docs
df= Number of docs that contain the word
weights[word]=s*$c_i$
Terms are ranked according to their weights.

The above algorithm was used only to rank the first few relevant terms in the collection. After the initial filtering, every query was expanded to include ten more terms according to the RM-1 model.

4. Re-Ranking:
Documents were ranked using KL divergence score:
$\sum_w P(w|R) \log P(w|D)$.

Results:

RM-1 Model

| | nDCG @5 | nDCG@10 | nDCG@50 | MAP |
|---|---|---|---|---|
| After Re-Ranking | 0.4131 | 0.4022 | 0.3843 | 0.0505 |
| Original file(Before Re-ranking) | 0.4844 | 0.4833 | 0.4424 | 0.0576 |

RM-2 Model:

| | nDCG @5 | nDCG@10 | nDCG@50 | MAP |
|---|---|---|---|---|
| After Re-Ranking | 0.4131 | 0.4022 | 0.3841 | 0.0506 |

| | | | | |
|---|---|---|---|---|
| Original file(Before Re-ranking) | 0.4844 | 0.4833 | 0.4424 | 0.0576 |