Search Engines Technology – Assignment
Student: Luiz Ribeiro (n9383298)
Search Engine used: Terrier 4.0 with "open heart surgery" approach.

Initial idea:

- Expanding the original queries using a thesaurus approach (in this case, the Consumer Health Vocabulary)
- Perform a further expansion based on mutual information over the collection
- Then to use using Dirichlet Jelinek-Mercer (or Two-stage) language model to score the documents and use the formulas proposed by its authors to perform tuning according to the collection [1].

Justification

- Since lay people aren't used to medical terms, the CHV expansion may allow the Search Engine to retrieve documents that talk about the query topic but using niche language, increasing the amount of relevant documents retrieved
- Some queries are small and general, so performing a further expansion by adding terms that co-occur on the collection may make them less vague
- The DJM model allows exploring an idea that combines two models seen on class, the Dirichlet and Jelinek-Mercer methods
- This model seems to be effective on this scenario, as Dirichlet performs better on big queries, while JM is better for small queries, so it's versatile
- The two-stage allows easy tuning by using the equations proposed by its authors

Evaluation

The evaluation results are displayed on Table 2. On this table, query expansion corresponds to the policy used to expand the queries, which are listed on Table 1. μ and λ are the tuning parameters and file name corresponds to the file on the repository, under the folder `tools/` containing the retrieved documents under such configuration.

Policy name	Strategy
CHV Expansion	Query expansion is performed only using the Consumer Health Vocabulary.
Long expansion	The amount of allowed expansions terms per query is the minimum between 7 and 14 – k, where k is the amount of words that a query contains.
Short expansion	Same as Long, but the bounds are 5 and 10 – k, respectively.
Log expansion	The amount of expansions terms is controlled by the following formula: $\left\| Ln\left(\frac{1}{1+k}\right) R \right\|$, where the operator $\|\|x\|\|$ indicates rounding. R is a constant that was set to 4. This expression grants that small queries are expanded more than longer queries, but that at least one new term is always added.

Table 1 – Developed policies

Query expansion	μ	λ	P@10	File name
Original queries	2500	0	0.2091	self_2500_0
Original queries	303	0	0.2227	self_303_0
Original queries	303	0.2	0.2227	self_303_50
Original queries	303	0.5	0.2227	self_303_50
Original queries	303	0.9	0.2212	self_303_90
CHV expansion	2500	0	0.1591	self_2500_0_chv
CHV expansion	303	0	0.1636	self_303_0_chv
Short expansion	2500	0	0.0470	self_2500_0_short
Short expansion	303	0	0.0455	self_303_0_short
Long expansion	2500	0	0.0273	self_2500_0_long
Long expansion	303	0	0.0258	self_303_0_long
Log expansion	2500	0	0.1152	self_500_0_lg
Log expansion	303	0	0.1212	self_303_0_lg

Table 2 – Evaluation using *trec_eval*

Performance analysis

- The only implementation that has brought improvement to the results was tuning the μ parameter according to the collection
- Changing the values of λ didn't reflect on the results until a big value, near to the maximum threshold of 1, was used. Still, this configuration decreased the relevance of the retrieved documents. This is probably due to the fact that λ multiplies a background probability extracted from the query logs
- Furthermore the corpus was used to calculate these probabilities due to the lack of a proper query log. Since the collection contains $2.8 \cdot 10^8$ tokens, even the most occurring tokens end up having very occurrence probability
- Also, as log probabilities are used, small numbers end up making no difference on the score calculations, as $log(x+y) \approx log(x)$ for $x \gg y$
- Finally, the query expansion strategies also decreased the P@10 evaluation. By observing the results it's possible to conclude that the more aggressive the expansion, worse the results
- It's also possible to observe that a linear expansions strategy (used on *"long"* and *"short"* policies) performs worse than a logarithmic policy
- Maybe taking in account the mutual information value before expanding a term can lead to better results.

References

[1] Zhai Chengxiang, John Lafferty. 2004 "A Study of Smoothing Methods for Language Models Applied to Information Retrieval". In *ACM Transactions on Information Systems, April 2004.*