Relevance models

1. The basic relevance model follows this formula:

$$p(M_d|q) = \frac{p(q|M_d)p(M_d)}{\sum_{d \in D_{init}} p(M_d)p(q|M_d)}$$

The relevance model is basically a weighted average of all documents in the collection where each document is weighted by its probability given the query. In other words, it is basically a weighted average of the query likelihood approach where we take the user's initial query, get a score (pseudo-relevance feedback), weight by query likelihood and normalize by the top documents that were originally retrieved (retrieved from the original query). Indeed, in the formula, when $p(M_d)$ is treated as uniform across all documents, what's left in the formula is the query likelihood.

2. We first obtain the conditional probabilities that produce the original ranking:

- $p(cat, dog|d1) = p(cat|d1) * p(dog|d1) = \frac{1+\frac{8}{19}}{5} \times \frac{1+\frac{2}{19}}{5} = 0{,}0628$
- $p(cat, dog|d2) = p(cat|d2) * p(dog|d2) = \frac{1+\frac{8}{19}}{8} \times \frac{1+\frac{2}{19}}{8} = 0{,}0245$
- $p(cat, dog|d3) = p(cat|d3) * p(dog|d3) = \frac{6+\frac{8}{19}}{6} \times \frac{0+\frac{2}{19}}{6} = 0{,}0188$

We consider the three documents to construct the relevance model:

- $p(cow|R) = p(pig|R) = p(horse|R) = \frac{1}{5} \times \frac{0{,}0628}{0{,}0628+0{,}0245+0{,}0188} = 0{,}1184$
- $p(the|R) = \frac{2}{8} \times \frac{0{,}0245}{0{,}0628+0{,}0245+0{,}0188} = 0{,}0577$
- $p(and|R) = p(are|R) = p(playing|R) = p(together|R)$
$$= \frac{1}{8} \times \frac{0{,}0245}{0{,}0628 + 0{,}0245 + 0{,}0188} = 0{,}0289$$
- $p(cat|R) = \frac{1}{5} \times \frac{0{,}0628}{0{,}0628+0{,}0245+0{,}0188} + \frac{1}{8} \times \frac{0{,}0245}{0{,}0628+0{,}0245+0{,}0188} + \frac{6}{6} \times \frac{0{,}0188}{0{,}0628+0{,}0245+0{,}0188} =$ $0{,}3244$
- $p(dog|R) = \frac{1}{5} \times \frac{0{,}0628}{0{,}0628+0{,}0245+0{,}0188} + \frac{1}{8} \times \frac{0{,}0245}{0{,}0628+0{,}0245+0{,}0188} = 0{,}1472$

When adding all the obtained probabilities for each word together we obtain 1.

3. For RM3, we incorporate some of the weight of the original query to improve retrieval quality.

$$\mathrm{p(w|R)} = \beta \times \mathrm{p_{MLE}(w|q)} + (1-\beta) \times \sum_{d \in D_{init}} p(\mathrm{w}|M_d)p(M_d|\mathrm{q})$$

Our query consists of only two words, so $\mathrm{p_{MLE}(w|q)} = 0{,}5$ for "cat" and "dog" and 0 for all other words. Calculating the probability of each word given the relevance model using RM3:

- $p(cow|R) = p(pig|R) = p(horse|R) = 0{,}3 \times 0 + 0{,}7 \times 0{,}1184 = 0{,}08288$
- $p(the|R) = 0{,}3 \times 0 + 0{,}7 \times 0{,}0577 = 0{,}04039$
- $p(and|R) = p(are|R) = p(playing|R) = p(together|R)$
$$= 0{,}3 \times 0 + 0{,}7 \times 0{,}0289 = 0{,}02023$$
- $p(cat|R) = 0{,}3 \times 0{,}5 + 0{,}7 \times 0{,}3244 = 0{,}37708$

- $p(dog|R) = 0{,}3 \times 0{,}5 + 0{,}7 \times 0{,}1472 = 0{,}25304$

When adding all the obtained probabilities for each word together we obtain 1.

Passage retrieval

- First approach: Maximum Entropy Divergence Minimization Model (MEDMM)

Proposed by Tao T. and Zhai C. (2006) the MEDMM is a robust method for pseudo-feedback based on statistical language models. The method integrates the original query with feedback documents in a probabilistic mixture model and regularizes the estimation of the language model parameters in the model so that the information feedback documents are added gradually to the original query.

MEDMM takes the original query model as a prior and applies it on the feedback language model to be estimated. Hence, the interpolation is reparametrized with more meaningful parameters and useful information from the feedback documents is gradually incorporated into the original query language model.

- Second approach: Latent Semantic Indexing

As proposed by Pawde K., Purbey N., Gargan S. and Kurup L. in their paper "Latent Semantic Analysis for Information Retrieval" (2014), latent semantic analysis is a technique used to mimic human understanding of language. It applies Singular Value Decomposition (SVD) on the term-document matrix with the goal of finding latent topics in the document collection. The model alleviates the vocabulary mismatch problem between the words used in the query and in a short relevant passage. Indeed, the methodology takes documents that are semantically similar but not analogous in the vector space and represents them in a reduced vector space which produces an increase in the cosine similarity. The probabilities are normalized to yield a valid language model. The process is as follows:

- User's query is represented as a vector in k dimensional space and compared to each of the documents.
- SVD is computed in the term-document matrix.
- Space and computation representation are reduced.
- Query is mapped onto the reduced space.
- Cosine similarity is calculated between the modified query and all the reduced documents in the reduced vector space.

- Third approach: knowing the fraction of the document that is relevant to the query

We consider the problem that relevant documents might also contain non-relevant information. For example, it is sufficient for a document to have only one sentence of relevant information to deem the document as relevant even though that the majority of the document is not relevant to the information need. Hence, we are interested in retrieving relevant passages from the documents. The following are three approaches for passage retrieval that aim to alleviate the stated problem:

Each document is split into fractions, that will be known as "passages". We compute the query likelihood on each passage and weight it by the fraction of relevant passages in the document. Hence, we won't consider each document to have a uniform probability. On the contrary, each document will be weighted by the fraction of it that is deemed relevant. This approach is intended for the cases in which we know what that fraction of relevance is for each document. The formula for this language model would be (taking the top k passages from the document):

$$\mathrm{P}(\mathrm{M_d}|\mathrm{q}) = \frac{\mathrm{p}(\mathrm{q}|\mathrm{M_d})\mathrm{p}(\mathrm{M_d})}{\sum_{\mathrm{d}\in\mathrm{D_{init}}}\mathrm{p}(\mathrm{M_d})\mathrm{p}(\mathrm{q}|\mathrm{M_d})}$$

Where:

$\mathrm{p}(\mathrm{M_d})$: fraction of the document that is considered relevant to the query.

For the model to be valid, it must be normalized so that the sum of the probabilities will equal to 1. In this case, the denominator contains the total sum, that includes the fraction for all selected documents.

Positional Language Models

Term	onion	vegetable	corn	soup	potato
Term frequency	3	2	1	1	1
P_MLE	$\frac{3}{8}$	$\frac{2}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$

Query: "onion soup"

We analyze the three documents according to the relevant positions:

- Document 2, position 1: "corn onion soup"

$$\mathrm{Z}_1 = 1k(1{,}1) + 1k(1{,}2) + 1k(1{,}3)$$

$$\mathrm{Z}_1 = \exp[0] + \exp\left[\frac{-(1\text{-}2)^2}{2\times 0.5^2}\right] + \exp\left[\frac{-(1\text{-}3)^2}{2\times 0.5^2}\right] = 1{,}1357$$

$$\mathrm{P_{Dir}}(\mathrm{onion}|\mathrm{M_{D2,1}}) = \frac{1\mathrm{k}(1{,}2) + 100\times\frac{3}{8}}{1{,}1357 + 100} = \frac{\exp\left[\frac{-(1\text{-}2)^2}{2\times 0.5^2}\right] + 100\times\frac{3}{8}}{1{,}1357+100} = 0{,}3721$$

$$\mathrm{P_{Dir}}(\mathrm{soup}|\mathrm{M_{D2,1}}) = \frac{1k(1{,}3) + 100\times\frac{1}{8}}{1{,}1357 + 100} = \frac{\exp\left[\frac{-(1\text{-}3)^2}{2\times 0.5^2}\right] + 100\times\frac{1}{8}}{1{,}1357+100} = 0{,}1236$$

$$\mathrm{S}(\text{"onion soup"}, \mathrm{M_{D2,1}}) = -\mathrm{p}(\mathrm{onion}|\mathrm{q})\ln\left(\frac{\mathrm{p}(\mathrm{onion}|\mathrm{q})}{\mathrm{p}(\mathrm{onion}|\mathrm{M_{D2,1}})}\right) - \mathrm{p}(\mathrm{soup}|\mathrm{q})\ln\left(\frac{\mathrm{p}(\mathrm{soup}|\mathrm{q})}{\mathrm{p}(\mathrm{soup}|\mathrm{M_{D2,1}})}\right)$$

$$= -\frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}3721}\right) - \frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}1236}\right) = -0{,}8465$$

- Document 2, position 2: "corn onion soup"

$$\mathrm{Z}_2 = 1k(2{,}1) + 1k(2{,}2) + 1k(2{,}3)$$

$$Z_2 = \exp[\frac{-(2-1)^2}{2 \times 0.5^2}] + \exp[0] + \exp\left[\frac{-(2-3)^2}{2 \times 0.5^2}\right] = 1{,}2707$$

$$P_{Dir}(\text{onion}|M_{D2,2}) = \frac{1k(2{,}2) + 100 \times \frac{3}{8}}{1{,}2707 + 100} = \frac{\exp[0] + 100 \times \frac{3}{8}}{1{,}2707 + 100} = 0{,}3802$$

$$P_{Dir}(\text{soup}|M_{D2,2}) = \frac{1k(2{,}3) + 100 \times \frac{1}{8}}{1{,}2707 + 100} = \frac{\exp\left[\frac{-(2-3)^2}{2 \times 0.5^2}\right] + 100 \times \frac{1}{8}}{1{,}2707 + 100} = 0{,}1248$$

$$S(\text{"onion soup"}, M_{D2,2}) = -\frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}3802}\right) - \frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}1248}\right) = -0{,}8309$$

- Document 2, position 3: “corn onion soup”

$$Z_3 = 1k(3{,}1) + 1k(3{,}2) + 1k(3{,}3)$$

$$Z_3 = \exp[\frac{-(3-1)^2}{2 \times 0.5^2}] + \exp\left[\frac{-(3-2)^2}{2 \times 0.5^2}\right] + \exp[0] = 1{,}1357$$

$$P_{Dir}(\text{onion}|M_{D2,3}) = \frac{1k(3{,}2) + 100 \times \frac{3}{8}}{1{,}1357 + 100} = \frac{\exp\left[\frac{-(3-2)^2}{2 \times 0.5^2}\right] + 100 \times \frac{3}{8}}{1{,}1357 + 100} = 0{,}3721$$

$$P_{Dir}(\text{soup}|M_{D2,3}) = \frac{1k(3{,}3) + 100 \times \frac{1}{8}}{1{,}1357 + 100} = \frac{\exp[0] + 100 \times \frac{1}{8}}{1{,}1357 + 100} = 0{,}1335$$

$$S(\text{"onion soup"}, M_{D2,3}) = -\frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}3721}\right) - \frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}1335}\right) = -0{,}8080$$

According to the best position strategy, the maximal score for document 2 is therefore on position 3.

- Document 1, position 1: “onion vegetable vegetable”

$$Z_1 = c(\text{onion}, 1) + c(\text{vegetable}, 1) = 1k(1{,}1) + 1k(1{,}2) + 1k(1{,}3)$$

$$Z_1 = \exp[0] + \exp\left[\frac{-(1-2)^2}{2 \times 0.5^2}\right] + \exp\left[\frac{-(1-3)^2}{2 \times 0.5^2}\right] = 1{,}1356707$$

$$P_{Dir}(\text{onion}|M_{D1,1}) = \frac{1k(1{,}1) + 100 \times \frac{3}{8}}{1{,}1356707 + 100} = \frac{\exp[0] + 100 \times \frac{3}{8}}{1{,}1356707 + 100} = 0{,}3806767$$

$$P_{Dir}(\text{soup}|M_{D1,1}) = \frac{0 + 100 \times \frac{1}{8}}{1{,}1356707 + 100} = 0{,}1235963$$

$$S(\text{"onion soup"}, M_{D1,1}) = -p(\text{onion}|q)\ln\left(\frac{p(\text{onion}|q)}{p(\text{onion}|M_{D2,1})}\right) - p(\text{soup}|q)\ln\left(\frac{p(\text{soup}|q)}{p(\text{soup}|M_{D2,1})}\right)$$

$$= -\frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}3806767}\right) - \frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}1235963}\right) = -0{,}835122$$

Neither "onion" nor "soup" appear in positions 2 and 3 so we can deduce that the highest score for document 1 will be for position 1.

- Document 3, position 2: "potato onion"

$$\mathrm{Z}_2 = \mathrm{c}(\text{potato}, 2) + \mathrm{c}(\text{onion}, 2) = 1k(2{,}1) + 1k(2{,}2)$$

$$\mathrm{Z}_2 = \exp\left[\frac{\text{-}(2\text{-}1)^2}{2 \times 0.5^2}\right] + \exp[0] = 1{,}135335$$

$$\mathrm{P}_{\mathrm{Dir}}(\text{onion}|\mathrm{M}_{\mathrm{D}3,2}) = \frac{1\mathrm{k}(2{,}2) + 100 \times \frac{3}{8}}{1{,}135335 + 100} = \frac{\exp[0] + 100 \times \frac{3}{8}}{1{,}135335 + 100} = 0{,}380678$$

$$\mathrm{P}_{\mathrm{Dir}}(\text{soup}|\mathrm{M}_{\mathrm{D}3,2}) = \frac{0 + 100 \times \frac{1}{8}}{1{,}135335 + 100} = 0{,}123597$$

$$\mathrm{S}(\text{"onion soup"}, \mathrm{M}_{\mathrm{D}3,2}) = -\frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}380678}\right) - \frac{1}{2}\ln\left(\frac{\frac{1}{2}}{0{,}123597}\right) = -0{,}835118$$

Hence, we obtain the following ranking:

Rank	Document	Smax(q,d)
1	2	-0,8080
2	3	-0,835118
3	1	-0,835122

Wet Part

Here are the results when running trec_eval method on Dinit before and after the query expansion:

Table1	Dinit			Best Expansion		
Query	MAP	P@5	P@10	MAP	P@5	P@10
301	0.053	0.6	0.6	0.1174	0.6	0.7
302	0.5636	0.6	0.7	0.5441	1	0.9
303	0.2090	0.2	0.1	0.2803	0.4	0.3
304	0.116	0.2	0.3	0.1299	0.4	0.3
305	0.0149	0.2	0.1	0.0580	0.4	0.3
306	0.1222	1	0.8	0.1222	1	0.8
307	0.2169	0.6	0.5	0.2251	0.8	0.7
308	0.4320	0.4	0.2	0.7145	0.6	0.3
309	0.0005	0.0	0.0	0.069	0.0	0.1
310	0.1484	0.4	0.3	0.2275	0.6	0.3
Average	0.1876	0.42	0.36	0.2488	0.58	0.47

The following are the expanded queries:

Query	Original query	Expansion words
301	international organized crime	drug trafficking
302	poliomyelitis post polio	vaccination cuba
303	hubble telescope achievements	quasar helium
304	endangered species mammals	beast
305	dangerous vehicles	cars fatal
306	african civilian deaths	-
307	new hydroelectric projects	china thailand
308	implant dentistry	tooth clinical
309	rap crime	suicide
310	radio waves brain cancer	radiation tumors

For the query expansion, we reviewed the most frequent terms in the relevant documents and considered the context of the document and of the query. According to this, we selected words and ran the evaluation for the expanded queries again, were we obtained the presented results. For some cases, like the query 302 “poliomyelitis post polio”, we read online that Cuba was amongst the first countries to eliminate polio and considered this information for our query expansion. We also consider synonyms of query terms, like for the case of query 305 “dangerous vehicles” we added the synonyms “fatal” for “dangerous” and “cars” for “vehicles”. However, for the query 306 “African civilian deaths” we decided not to expand it, since in this case the query expansion caused a decrease in the indicators. Indeed, there are some cases in which expanding a query will generate more generic results than before the expansion, so for those cases query expansion is not recommended.

Language Models:

(1) $q =$ "information retrieval"

Doc1 = "information retrieval course is fun"
Doc2 = "information information information information information information computer"
Doc3 = "information retrieval retrieval methods"

$\mu = 1000$

TFs:

	information	retrieval	course	is	fun	computer	methods
q	1	1	0	0	0	0	0
Doc1	1	1	1	1	1	0	0
Doc2	6	0	0	0	0	1	0
Doc3	1	2	0	0	0	0	1
collection	8	3	1	1	1	1	1

$|Doc1| = 5 \quad |Doc2| = 7 \quad |Doc3| = 4$

$|collection| = 16$

Collection language model w.o. smoothing:

$P(information \mid C) = 8/16 = 0.5$
$P(retrieval \mid C) = 3/16 = 0.1875$

Query likelihood, using Dirichlet smoothing:

$$p(Q|D) = \prod_{t \in Q} \frac{C(t|D) + \mu \cdot p(t|C)}{|D| + \mu}$$

Doc1:

$$p(\text{information} | Doc1) = \frac{1 + 100 \cdot 0.5}{5 + 100} = 0.485$$

$$p(\text{retrieval} | Doc1) = \frac{1 + 100 \cdot 0.1875}{5 + 100} = 0.188$$

$$p(q | Doc1) = 0.485 \cdot 0.188 = 0.0911$$

Doc2:

$$p(\text{information} | Doc2) = \frac{6 + 100 \cdot 0.5}{7 + 100} = 0.523$$

$$p(\text{retrieval} | Doc2) = \frac{0 + 100 \cdot 0.1875}{7 + 100} = 0.175$$

$$p(q | Doc2) = 0.523 \cdot 0.175 = 0.0915$$

Doc3:

$$p(\text{information} | Doc3) = \frac{1 + 100 \cdot 0.5}{4 + 100} = 0.49$$

$$p(\text{retrieval} | Doc3) = \frac{2 + 100 \cdot 0.1875}{4 + 100} = 0.199$$

$$p(q | Doc3) = 0.097$$

Ranking:

1. Doc3
2. Doc2
3. Doc1

(2) Original: $\hat{P}(t|D) = \lambda \cdot P(t|D) + (1-\lambda) \cdot P(t|C)$

One possible modification is to make λ be inversly proportional to the query length, since we want to give less weight to the document model, and more to the collection model as the query becomes longer.

$$\lambda' = \frac{1}{1+|Q|}$$

$$\hat{p}'(t|D) = \lambda' \cdot P(t|D) + (1-\lambda') \cdot P(t|C)$$

(3) Let's assume $Q = \{q_1, q_2, \ldots q_n\}$ is a set of terms representing the query.
Also, lets assume the query likelihood is approximated using the relative

frequency of each term in the document (the standart model).

Lastly, lets assume the cross entropy is calculated between two distributions:

P - the distribution of the query terms in the document (the query likelihood).

Q - which we will assume to be a uniform distribution over the query terms.

The Cross Entropy between these is:

$$H(P,Q) = -\sum P(q_i|D) \log Q(q_i)$$

but, since Q is uniform accoring to our assumption, we can substitude $Q(q_i)$ for $\frac{1}{n}$, giving us:

$$H(P,Q) = -\sum P(q_i|D) \log\left(\frac{1}{n}\right)$$

now, the query likelihood is

given by:

$$P(Q|D) = \prod P(q_i|D)$$

taking the log of this gives us:

$$\log P(Q|D) = \sum \log P(q_i|D)$$

Since log is monotonically increasing, the order remains the same.

next:

$$-H(P,Q) = \sum [P(q_i|D)] + \underbrace{c \cdot \log(\tfrac{1}{n})}_{\text{constant}}$$

we can remove the constant, since it only changes the value without changing the order, and we will get $\sum P(q_i|D)$ which is the same as what we achieved for $\log P(Q|D)$, showing that the order induced by $-H(P,Q)$ is the same as the one induced by the query likelihood model. $\square$

4. The Jensen-Shannon divergence uses the KL divergenc to achieve a symmetric, non-negative divergence,

It does so by averaging the two directions of the KL divergence, once from P to Q, and once from Q to P.

$$JS(P,Q) = 0.5\ KL(P\|M) + 0.5\ KL(Q\|M)$$

where $M = 0.5 \cdot (P+Q)$