WSM Assignment 2:

Probabilistic Information Retrieval & Language Model for Information Retrieval

1. Consider RSJ retrieval model, the contingency table of counts of documents, and the statement below:

$$RSV_d = \log \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t:x_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)}$$

documents	relevant	non-relevant	Total
Term present $x_t = 1$	s	$df_t - s$	df_t
Term absent $x_t = 0$	$S - s$	$(N - df_t) - (S - s)$	$N - df_t$
Total	S	$N - S$	N

a) What are the differences between standard VSM with tf-idf weightings & RSJ probabilistic retrieval model (in the case where no relevance information is provided)?

tf-idf weighting compare vocab frequency between one particular document and other documents; but the RSJ probabilistic retrieval model consider only the absence or presence of term in the document, which it treats all terms are the same. And we say the vsm with tfidf will function better than RSJ probabilistic retrieval model when there are no relevant information is provided, since RSJ counts words contribution by the fraction of such other documents in the collection. VSM based on similarity in geometric / linear algebra notion, RSJ based on probability theory.

b) Describe the differences of relevance feedback used in VSM and probabilistic retrieval models.

Since p_i in probabilistic retrieval models is constant, an initial guess is to be done for the relevant document set under the assumption that all query terms give same values of document relevance; whereas in VSM will need no any require initial assumption as the relevant document set can be computed over the collection & query, and can be done in fewer iterations with better performance.

c) Please show that based on the model above, documents can ranked via the following formula: $\log \frac{S \times [(N-df_t)-(S-s)]}{(S-s)(df_t-s)}$

we know $p_t = \frac{s}{S}$, $u_t = \frac{df_t - s}{N-S}$

then
$$\log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{\frac{s}{S}\left(1-\frac{df_t-s}{N-S}\right)}{\frac{df_t-s}{N-S}\left(1-\frac{s}{S}\right)}$$

$$= \log \frac{\frac{s}{S}\left(\frac{N-S-df_t+s}{N-S}\right)}{\frac{df_t-s}{N-S}\left(\frac{S-s}{S}\right)} \times \frac{N-S}{N-S}$$

$$= \log \frac{\frac{s}{S}(N-S-df_t+s)}{(df_t-s)\left(\frac{S-s}{S}\right)} \times \frac{S}{S}$$

$$= \log \frac{s(N-S-df_t+s)}{(df_t-s)(S-s)}$$

$$= \log \frac{s[(N-df_t)-(S-s)]}{(S-s)(df_t-s)}$$

2. training text : she sells seashells by the seashore 6
the shells she sells are surely seashells 7
so if she sells shells on the seashore 8
I am sure she sells seashore shells 7

a) Under a MLE-estimated unigram probability model, what are P(the) & P(seashore)?

(total of words) $T_w = 28$

$P(\text{the}) = \frac{3}{28}$, $P(\text{seashore}) = \frac{3}{28}$

b) Under a MLE-estimated bigram model, what are P(seashells | sells) and P(sells | the)?

bigram : $P(w_i | w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})}$

$P(\text{seashells} | \text{sells}) = \frac{1}{4}$, $P(\text{sells} | \text{the}) = \frac{0}{3}$

3. collection that consists of the 4 documents given in the below table.

docID	Document text
1	click go the shears boys click click click
2	click click
3	mental here
4	mental shears click here

Build a query likelihood language model for this document collection. Assume a mixture model between the documents & collection, with both weighted at 0.5. Maximum Likelihood Estimation (MLE) is used to estimate both as unigram models. Work out the model probabilities of the queries click, shears, and hence click shears for each document, and use those probabilities to rank the documents returned by following query.

$$P(q|d) \propto \prod_{1 \le k \le |q|} \left[\lambda P(t_k|M_d) + (1-\lambda) P(t_k|M_c)\right]$$

$$P(t|d) = \lambda P(t|M_d) + (1-\lambda) P(t|M_c)$$

λ will usually be $\frac{1}{2}$.

a) click

$$P(\text{click}|d_1) = \frac{1}{2}\left(\frac{4}{8}\right) + \frac{1}{2}\left(\frac{7}{16}\right) = \frac{15}{32}$$

$$P(\text{click}|d_2) = \frac{1}{2}\left(\frac{2}{2}\right) + \frac{1}{2}\left(\frac{7}{16}\right) = \frac{23}{32}$$

$$P(\text{click}|d_3) = \frac{1}{2}\left(\frac{0}{2}\right) + \frac{1}{2}\left(\frac{7}{16}\right) = \frac{7}{32}$$

$$P(\text{click}|d_4) = \frac{1}{2}\left(\frac{1}{4}\right) + \frac{1}{2}\left(\frac{7}{16}\right) = \frac{11}{32}$$

$\therefore$ for query "click", the ranking will be $d_2 > d_1 > d_4 > d_3$.

b) shears

$$P(\text{shears}|d_1) = \frac{1}{2}\left(\frac{1}{8}\right) + \frac{1}{2}\left(\frac{2}{16}\right) = \frac{1}{8}$$

$$P(\text{shears}|d_2) = \frac{1}{2}\left(\frac{0}{2}\right) + \frac{1}{2}\left(\frac{2}{16}\right) = \frac{1}{16}$$

$$P(\text{shears}|d_3) = \frac{1}{2}\left(\frac{0}{2}\right) + \frac{1}{2}\left(\frac{2}{16}\right) = \frac{1}{16}$$

$$P(\text{shears}|d_4) = \frac{1}{2}\left(\frac{1}{4}\right) + \frac{1}{2}\left(\frac{2}{16}\right) = \frac{3}{16}$$

$\therefore$ for query "shears", the ranking will be $d_4 > d_1 > d_2 = d_3$.

c) click shears

$P(\text{click shears} \mid d_1) = P(\text{click} \mid d_1)\, P(\text{shears} \mid d_1) = \frac{15}{32}\left(\frac{1}{8}\right) = \frac{15}{256}$

$P(\text{click shears} \mid d_2) = P(\text{click} \mid d_2)\, P(\text{shears} \mid d_2) = \frac{23}{32}\left(\frac{1}{16}\right) = \frac{23}{512}$

$P(\text{click shears} \mid d_3) = P(\text{click} \mid d_3)\, P(\text{shears} \mid d_3) = \frac{7}{32}\left(\frac{1}{16}\right) = \frac{7}{512}$

$P(\text{click shears} \mid d_4) = P(\text{click} \mid d_4)\, P(\text{shears} \mid d_4) = \frac{11}{32}\left(\frac{3}{16}\right) = \frac{33}{512}$

∴ for the query "click shears", the ranking will be $d_4 > d_1 > d_2 > d_3$

4. Given query "Taiwan Taoyuan", compute the ranking of the three documents by MLE unigram models from the documents & collection, mixed with lambda = $\frac{1}{2}$.

d_1 = He moved from Taiwan, Taipei, to Taiwan, Taoyuan. 8

d_2 = He moved from Taiwan, Taoyuan, to Taiwan, Taipei. 8

d_3 = He moved from Taoyuan to Taiwan, Taipei. 7

$$P(\text{Taiwan Taoyuan} \mid d_1) = P(\text{Taiwan} \mid d_1)\, P(\text{Taoyuan} \mid d_1)$$
$$= \frac{1}{2}\left(\frac{2}{8} + \frac{5}{23}\right) * \frac{1}{2}\left(\frac{1}{8} + \frac{3}{23}\right)$$
$$= 0.02985$$

$$P(\text{Taiwan Taoyuan} \mid d_2) = P(\text{Taiwan} \mid d_2)\, P(\text{Taoyuan} \mid d_2)$$
$$= \frac{1}{2}\left(\frac{2}{8} + \frac{5}{23}\right) * \frac{1}{2}\left(\frac{1}{8} + \frac{3}{23}\right)$$
$$= 0.02985$$

$$P(\text{Taiwan Taoyuan} \mid d_3) = P(\text{Taiwan} \mid d_3)\, P(\text{Taoyuan} \mid d_3)$$
$$= \frac{1}{2}\left(\frac{1}{7} + \frac{5}{23}\right) * \frac{1}{2}\left(\frac{1}{7} + \frac{3}{23}\right)$$
$$= 0.02461$$

5. a) In the derivation of the RSJ model, a multi-variate Bernoulli model was used to model term presence/absence in a relevant document & a non-relevant document. Suppose, we change the model to a multinomial model. Using a similar independence assumption as we used in deriving RSJ, show that ranking based on probability that a document is relevant to query Q, i.e. $P(R=1|D,Q)$, is equivalent to ranking based on the following formula:

$$\text{Score}(Q,D) = \sum_{w \text{ in } V} c(w,D) \log \frac{P(w|Q,R=1)}{P(w|Q,R=0)}$$

where the sum is taken over all the word in our vocabulary (denoted by V). How many parameters are there in such a retrieval model?

ranking by $P(R=1|D,Q)$ is equivalent to ranking by $O(R|D,Q)$,

and $$O(R|D,Q) \propto \frac{P(D|Q,R=1)}{P(D|Q,R=1)}$$

$$\Rightarrow \log \frac{P(D|Q,R=1)}{P(D|Q,R=1)} = \log \prod_{w \in V} \frac{P(D|Q,R=1)}{P(D|Q,R=1)}$$

∵ we are gonna be need for all $w \in V$ $P(D|Q,R=1)$ & $P(D|Q,R=0)$,

∴ there are $2|V|$ parameters.

b) The retrieval function above won't work unless we can estimate all the parameters. Suppose we use the entire collection $C = \{D_1, \ldots, D_n\}$ as an approximation of the examples of non-relevant documents. Propose an estimate of $P(w|Q,R=0)$

$$\log O(R=1|Q,D) \approx \sum_{i=1, d_i=q_i=1}^{k} \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

N: # documents in collection

n_i: # documents in which term A_i occurs

$$P(D|Q,R=0) = \frac{\#\ W \text{ in } C}{\#\ \text{words in } C} = \frac{\sum_{D \in C} C(W, D_i)}{\sum_{D \in C} |D_i|}$$

c) Suppose we use the query as the only example of a relevant document. Propose an estimate of $p(w|Q,R=1)$.

$$P(D|Q,R=1) = \frac{\#\ W \text{ in } Q}{\#\ \text{words in } Q} = \frac{C(W,Q)}{|Q|}$$

d) improving RSJ with TF:

basic doc generation model $\frac{P(R=1|Q,D)}{P(R=0,Q,D)} \propto \frac{P(D|Q,R=1)}{P(D|Q,R=0)}$

$$\propto \prod_{i=1}^{k} \frac{P(A_i = d_i | Q, R=1)}{P(A_j = d_i | Q, R=0)}$$

we can also incorporating doc length, since 2-poisson model assumes equal document length. and incorporating query TF with a similar TF transformation.

e) I dont think my formula would work better than BM25, thus I cannot come up with any better idea to further improve my formula.