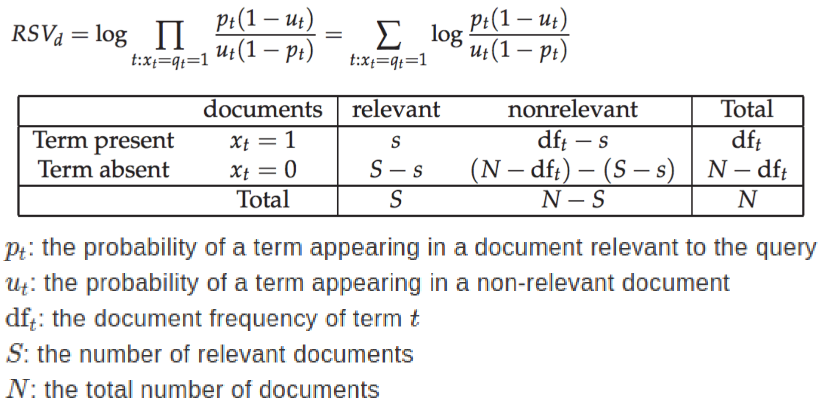
WSM Assignment 2: Probabilistic Information Retrieval and Language Model for Information Retrieval

資科碩一110753201 曹昱維

April 4, 2022

1 [20 points] Probabilistic Information Retrieval

Consider the RSJ retrieval model, the contingency table of counts of documents, and the statements 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	nonrelevant	Total
Term present	$x_t = 1$	s	$\mathrm{df}_t - s$	df_t
Term absent	$x_t = 0$	$S - s$	$(N - \mathrm{df}_t) - (S - s)$	$N - \mathrm{df}_t$
	Total	S	$N - S$	N

p_t: the probability of a term appearing in a document relevant to the query

u_t: the probability of a term appearing in a non-relevant document

df_t: the document frequency of term t

S: the number of relevant documents

N: the total number of documents

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

Ans:

1. The tf-idf weighting considers the term frequency but, **the RSJ retrieval model(with no relevance information) only consider the absence or presence of term**.
2. RSJ retrieval model(with no relevance information) is more like only idf weight without tf weight.

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

Ans:

1. Probabilistic retrieval models is most natural for relevance/pseudo feedback.
   - Because probabilistic retrieval models can use feedback to reallocate the probablity of the term A_i occurs in relevent/non-relevent documents
   - In other words, probabilistic retrieval models can regard feedback as relevence infomation
2. So, Probabilistic retrieval model **need to recalculate initial probability value of each term**.
3. In vector space , it uses the relevance feedback to calculate re-rank score, so it **doesn't need recalculate initial weight of each term.**

1.3 Please show that based on the model above, documents can be ranked via the following formula:

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

Ans:

$$p_t = \frac{s}{S} \tag{1}$$

$$u_t = \frac{df_t - s}{N - S} \tag{2}$$

$$\begin{aligned} RSV_d &= \sum log \frac{p_t(1 - u_t)}{u_t(1 - p_t)} \qquad (3) \\ &= \sum log \frac{p_t}{1 - p_t} + \sum log \frac{1 - u_t}{u_t} \\ &= \sum log \frac{s}{S - s} + \sum log \frac{N - S - df_t + s}{df_t - s} \\ &= \sum log \frac{s * ((N - df_t) - (S - s))}{(S - s)(df_t - s)} \end{aligned}$$

2 [10 points] Language Model

Consider making a language model from the following training text:

how much wood would a woodchuck chuck
if a woodchuck could chuck wood
he would chuck he would as much as he could
and chuck as much as a woodchuck would
if a woodchuck could chuck wood

2.1 Under a MLE-estimated unigram probability model, what are P(would) and P(chuck)?

Ans:

$$\hat{P}(t|M_d) = \frac{tf_{t,d}}{|d|}$$

$$\therefore P(would) = \frac{4}{37},\ P(chuck) = \frac{5}{37}$$

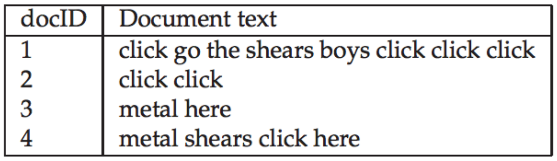
2.2 Under a MLE-estimated bigram model, what are P(wood — chuck) and P(chuck — would)?

Ans:

1. $P(wood|chuck) = \frac{count(wood,chuck)}{count(chuck)} = \frac{2}{5}$
2. $P(chuck|would) = \frac{count(chuck,would)}{count(would)} = \frac{1}{4}$

3 [15 points] Language Model

Suppose we have a 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	metal here
4	metal shears click here

Build a query likelihood language model for this document collection. Assume a mixture model between the documents and the 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.

3.1 [5/15 points] click

Ans:

query	$Model_{d_1}$	$Model_{d_2}$	$Model_{d_3}$	$Model_{d_4}$
click	$\frac{\frac{4}{8}+\frac{7}{16}}{2}$	$\frac{\frac{2}{2}+\frac{7}{16}}{2}$	$\frac{\frac{0}{2}+\frac{7}{16}}{2}$	$\frac{\frac{1}{4}+\frac{7}{16}}{2}$

$$\therefore Doc_2 > Doc_1 > Doc_4 > Doc_3$$

3.2 [5/15 points] shears

Ans:

query	$Model_{d_1}$	$Model_{d_2}$	$Model_{d_3}$	$Model_{d_4}$
shears	$\frac{\frac{1}{8}+\frac{2}{16}}{2}$	$\frac{\frac{0}{2}+\frac{2}{16}}{2}$	$\frac{\frac{0}{2}+\frac{2}{16}}{2}$	$\frac{\frac{1}{4}+\frac{2}{16}}{2}$

$$\therefore Doc_4 > Doc_1 > Doc_2 = Doc_3$$

3.3 [5/15 points] click shears

Ans:

query	$Model_{d_1}$	$Model_{d_2}$	$Model_{d_3}$	$Model_{d_4}$
click shears	$\frac{\frac{4}{8}+\frac{7}{16}}{2} * \frac{\frac{1}{8}+\frac{2}{16}}{2}$	$\frac{\frac{2}{2}+\frac{7}{16}}{2} * \frac{\frac{0}{2}+\frac{2}{16}}{2}$	$\frac{\frac{0}{2}+\frac{2}{16}}{2} * \frac{\frac{0}{2}+\frac{2}{16}}{2}$	$\frac{\frac{1}{4}+\frac{2}{16}}{2} * \frac{\frac{1}{4}+\frac{2}{16}}{2}$

$$\therefore Doc_4 > Doc_1 > Doc_2 > Doc_3$$

4 [5 points] Mixture model

Given the query "Taipei Taiwan", please compute the ranking of the three documents by MLE unigram models from the documents and collection, mixed with lambda = 1/2

- He moved from Taiwan, Taipei, to Taiwan, Nantou.
- He moved from Taiwan, Nantou, to Taiwan, Taipei.
- He moved from Nantou to Taiwan, Taipei.

4.1 [5/5 points] Taipei Taiwan

Ans:

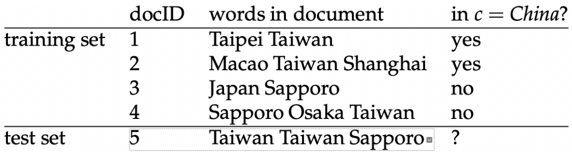
- He moved from Taiwan, Taipei, to Taiwan, Nantou. ⇒ Doc_1
- He moved from Taiwan, Nantou, to Taiwan, Taipei. ⇒ Doc_2
- He moved from Nantou to Taiwan, Taipei. ⇒ Doc_3

query	$Model_{d_1}$	$Model_{d_2}$	$Model_{d_3}$
Taipei Taiwan	$\frac{\frac{1}{8}+\frac{3}{23}}{2} * \frac{\frac{2}{8}+\frac{5}{23}}{2}$	$\frac{\frac{1}{8}+\frac{3}{23}}{2} * \frac{\frac{2}{8}+\frac{5}{23}}{2}$	$\frac{\frac{1}{7}+\frac{3}{23}}{2} * \frac{\frac{1}{7}+\frac{5}{23}}{2}$

$$\therefore Doc_1 = Doc_2 > Doc_3$$

5 [20 points] Text Classification

On the basis of the following data in the table:



	docID	words in document	in $c = China$?
training set	1	Taipei Taiwan	yes
	2	Macao Taiwan Shanghai	yes
	3	Japan Sapporo	no
	4	Sapporo Osaka Taiwan	no
test set	5	Taiwan Taiwan Sapporo	?

5.1 [5/20 points] Estimate a multinomial Naive Bayes (NB) classifier

Ans:

prior probability:
$\hat{P}(c) = \frac{1}{2}, \ \hat{P}(\bar{c}) = \frac{1}{2}$

`conditional probabilities with` *`Laplace smoothing`* :

$\hat{P}(t|c) = \dfrac{T_{ct}+1}{\sum_{t'\in V}+B}$

where T_{ct} is the number of occurrences of t in training documents from class c, including multiple occurrences of a term in a document,
and $B = |V|$ is the number of terms in the vocabulary

∴ conditional probabilities :

$$\hat{P}(Taiwan|c) = \frac{2+1}{7+7} = \frac{3}{14}, \ \hat{P}(Taiwan|\bar{c}) = \frac{1+1}{5+7} = \frac{2}{12}$$
$$\hat{P}(Sapporo|c) = \frac{0+1}{7+7} = \frac{1}{14}, \ \hat{P}(Sapporo|\bar{c}) = \frac{2+1}{5+7} = \frac{3}{12}$$

5.2 [5/20 points] Apply the classifier to the test document (docID = 5)

Ans:

$$\hat{P}(c|d_5) \propto \frac{1}{2} * \frac{3}{14} * \frac{3}{14} * \frac{1}{14} \simeq 0.00163994$$
$$\hat{P}(\bar{c}|d_5) \propto \frac{1}{2} * \frac{2}{12} * \frac{2}{12} * \frac{3}{12} \simeq 0.00347222$$

∴ the classifier assigns the test document to $\bar{c}$ = not-China

---

5.3 [5/20 points] Estimate a multivariate Bernoulli NB classifier

Ans:

prior probability:
$\hat{P}(c) = \frac{1}{2}, \ \hat{P}(\bar{c}) = \frac{1}{2}$

Bernoulli model estimates conditional probabilities：

$\hat{P}(t|c)$ as the fraction of documents of class c that contain term t

∴ conditional probabilities：

$$\begin{aligned}
&\hat{P}(Taiwan|c) = \frac{2+1}{2+2} = \frac{3}{4}, && \hat{P}(Taiwan|\bar{c}) = \frac{1+1}{2+2} = \frac{2}{4}\\
&\hat{P}(Sapporo|c) = \frac{0+1}{2+2} = \frac{1}{4}, && \hat{P}(Sapporo|\bar{c}) = \frac{2+1}{2+2} = \frac{3}{4}\\
&\hat{P}(Macao|c) = \frac{1+1}{2+2} = \frac{2}{4}, && \hat{P}(Macao|\bar{c}) = \frac{0+1}{2+2} = \frac{1}{4}\\
&\hat{P}(Shanghai|c) = \frac{1+1}{2+2} = \frac{2}{4}, && \hat{P}(Shanghai|\bar{c}) = \frac{0+1}{2+2} = \frac{1}{4}\\
&\hat{P}(Japan|c) = \frac{0+1}{2+2} = \frac{1}{4}, && \hat{P}(Japan|\bar{c}) = \frac{1+1}{2+2} = \frac{2}{4}\\
&\hat{P}(Osaka|c) = \frac{0+1}{2+2} = \frac{1}{4}, && \hat{P}(Osaka|\bar{c}) = \frac{1+1}{2+2} = \frac{2}{4}
\end{aligned}$$

5.4 [5/20 points] Apply the classifier to the test document (docID = 5)

Ans:

$$\begin{aligned}
\hat{P}(c|d_5) \quad &\propto \hat{P}(c) * \hat{P}(Taiwan|c) \qquad\qquad *\\
&* (1 - \hat{P}(Japan|c)) * (1 - \hat{P}(Osaka|c))\\
&= \frac{1}{2} * \frac{3}{4} * \frac{1}{4} * (1 - \frac{1}{2}) * (1 - \frac{1}{2}) * (1 - \frac{1}{4}) * (1 - \frac{1}{4})\\
&\simeq 0.01318359\\
\hat{P}(\bar{c}|d_5) \quad &\propto \hat{P}(\bar{c}) * \hat{P}(Taiwan|\bar{c}) * \hat{P}(Sapporo|\bar{c}) * (1 - \hat{P}(Macao|\bar{c})) * (1 - \hat{P}(Shanghai|\bar{c}))\\
&* (1 - \hat{P}(Japan|\bar{c})) * (1 - \hat{P}(Osaka|\bar{c}))\\
&= \frac{1}{2} * \frac{2}{4} * \frac{3}{4} * (1 - \frac{1}{4}) * (1 - \frac{1}{4}) * (1 - \frac{1}{2}) * (1 - \frac{1}{2})\\
&\simeq 0.0263671
\end{aligned}$$

∴ the classifier assigns the test document to $\bar{c}$ = not-China

6 [30 points] Classic Probabilistic Retrieval Model.

6.1 [10/30 points] In the derivation of the Robertson-Sparck-Jones (RSJ) model (see the slides and the survey paper by Norbert Fuhr for detail about this derivation), a multi-variate Bernoulli model was used to model term presence/absence in a relevant document and a non-relevant document. Suppose, we change the model to a multinomial model (see the slide that covers both models for computing query likelihood). Using a similar independence assumption as we used in deriving RSJ, show that ranking based on probability that a document is relevant to a 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?

Ans:

RSJ-model is a document generation model, and $c(\omega, D)$ is depending on $document_D$ itself, therefore $c(\omega, D)$ is not the parameter of model.

The parameters should based on conditions of retrieval model which are corpus(trained text) and query, **therefore the parameters are** $p(w|Q, R = 1)$ **and** $p(w|Q, R = 0)$ **for each word** ω_i**, so the total number of parameters is** $2|V|$**. But** $p(w|Q, R = 1)$ **subject to** $\sum_{\omega \in V} p(w|Q, R = 1) = 1$ **and** $p(w|Q, R = 0)$ **subject to** $\sum_{\omega \in V} p(w|Q, R = 0) = 1$**, so the retrieve model have** $2|V| - 2$ **free parameters.**

6.2 [5/30 points] The retrieval function above won't work unless we can estimate all the parameters. Suppose we use the entire collection $C = \{D_1, ..., D_n\}$ as an approximation of the examples of non-relevant documents. Propose an estimate of $p(w|Q, R = 0)$. (Hint: study the slide about how to do this for the RSJ model.)

Ans:

$$p(w|Q, R = 0) = \frac{\sum_{i=1}^{n} c(\omega, D_i)}{N}$$

where $c(\omega, D_i)$ is frequency of word ω occurs in document D_i,

$$\text{and } N = \sum_{i=1}^{n} |D_i|$$

6.3 [5/30 points] Suppose we use the query as the only example of a relevant document. Propose an estimate of $p(w|Q, R = 1)$.

Ans:

$$p(w|Q, R = 1) = \frac{c(\omega, Q)}{|1|}$$

where $c(\omega, Q)$ is # of word ω occurs in query Q,

and $|Q| = |1|$

6.4 [5/30 points] With the two estimates you proposed, we should now have a retrieval function that can be used to compute a score for any document D and any query Q. Write down your retrieval function by plugging in the two estimates. Can your retrieval function capture the three major retrieval heuristics (i.e., TF, IDF, and document length normalization)? How?

Ans:

$$\sum_{\omega\in V} c(\omega, D) log \frac{p(w|Q, R=1)}{p(w|Q, R=0)} = \sum_{\omega\in V} c(\omega, D) log \frac{\frac{c(\omega,Q)}{|1|}}{\frac{\sum_{i=1}^{n} c(\omega,D_i)}{N}}$$

where $c(\omega, D_i)$ is # of word ω occurs in document D_i,

and $N = \sum_{i=1}^{n} |D_i|$,

and $c(\omega, Q)$ is # of word ω occurs in query Q

- anology of TF term : $\frac{c(\omega,Q)}{|1|}$
- anology of IDF term : $\frac{\sum_{i=1}^{n} c(\omega,D_i)}{N}$
- anology of document length normalization : None

6.5 [5/30 points] Do you believe your formula would work well as compared with a state of the art formula such as BM25? Can you propose a way to further improve your formula? (While it's the best if you could improve your formula through using an improved estimate of $p(w|Q, R=1)$, it would also be fine to propose any reasonable heuristic modification of the formula.)

Ans:

No, it can't perform as well as BM25, improvement as following:

- Zero probability problem, which can solve by smoothing
- Doesn't consider the length normalization