1. **[10 points] Evaluation**

   (a) **[3 points]** Suppose we have a topic (i.e., a query) with a total of 10 relevant documents in a collection with 100 documents. A system has retrieved 6 documents whose relevance status is 
   
   \[
   [+,-,-,+,-;+]
   \]

   in the order of ranking. A .+ (or -. ) indicates that the corresponding document is relevant (or nonrelevant). For example, the _rst document is relevant, while the second is non-relevant, etc. Compute the precision, recall, and the (non-interpolated) mean average precision for this result.

   **Answer:**
   
   - Precision = 3/6 = 0.5
   - Recall = 3/10 = 0.3
   - MAP = \( \frac{1 + 2/4 + 3/6}{6} \approx 0.333 \)

   (b) **[4 points]** Now suppose our collection has a total of 10,000 documents, instead of 100 documents, but the topic has the same number of relevant documents (i.e., 10) in the collection, and the system has retrieved exactly the same results as in the previous question for this topic. Intuitively, the system's performance is better in this case than in the previous case since now we have more distracting non-relevant documents and thus the retrieval task is much harder. Explain why precision and recall cannot reflect this difference, and propose a reasonable new measure that would reflect this difference. How important is it to capture this difference in our evaluation of retrieval methods?

   **Answer:**

   1) Neither the precision or recall has relation with the size of the corpus. So they can not reflect the difference.

   2) 增加一个能反映文档集大小及获取到相关文档数关系的因素，例如

   \[ \alpha = \frac{\text{relevant} \cap \text{retrieved}}{\text{total docs}}, F = \frac{PR}{P + R} \]

   新测量： \( N = \frac{F}{\alpha} \)

   或

<table>
<thead>
<tr>
<th></th>
<th>Retrieved</th>
<th>Non-retrieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>Irrelevant</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

   其中 total docs = a + b + c + d
3) 从用户的角度，这种评估并不重要，因为用户更关心返回结果中多少相关文档。从评估检索系统的角度，不同方法面临的处理难度相同，因此文档集大小对评估结果影响并不重要。

(c) [3 points] When reusing a retrieval test collection created using the pooling strategy to evaluate new retrieval functions that have not contributed to the pool, why is there a potential bias against the new functions?
Answer:
因为 pooling 结果由构造该 pool 的 run 贡献并经人工判断找其中正确答案，新的 run 如果从原数据集中找到了一些未在 pool 中的正确结果，是不被判断未正确的。所以对未参与 pool 的结果有偏差（不公正）。

2. [20 points] Global Corpus Statistics
Consider a document collection $C_0 = \{D_1, \ldots, D_N\}$, where $D_i$ is a document. We construct two other collections $C_1$ and $C_2$ in the following way:
(1) $C_1$ is formed by deleting $D_1$ from $C_0$, thus it has $N - 1$ documents.
(2) $C_2$ is formed by replacing $D_1$ in $C_0$ with a new document $D_1'$, which contains all the distinct terms in $D_1$ but all the terms only occurs once. $C_2$ thus has the same number of documents as $C_0$.

Let $t$ be a term occurring more than once in $D_1$ (i.e. $c(t, D_1) > 1$) and IDF$(t, C_i)$ be the IDF value of $t$ in collection $C_i$. Let $\theta(C_i)$ be a collection language model estimated based on collection $C_i$.

(a) [5 points] What can we say about IDF$(t;C_1)$ and IDF$(t;C_0)$? Are they the same? If not, which is higher? Why?
Answer:
$\because \quad IDF(t, C_1) = \frac{N - 1}{m - 1}, \quad IDF(t, C_0) = \frac{N}{m}, \quad m 是 C_0 中包含 t 的文档数 (m \leq N)$

$\frac{N - 1}{m - 1} = \frac{N m - m}{Nm - N} \geq 1, \quad \therefore \quad IDF(t; C_1) \geq IDF(t; C_0)$

(b) [5 points] What can we say about IDF$(t, C_2)$ and IDF$(t, C_0)$? Are they the same? If not, which is higher? Why?
Answer:
因为 $D_1'$ 包含了所有 $D_1$ 中的单一 term。So they are the same.

(c) [5 points] Is $p(t \mid \emptyset (C_1))$ higher than, lower than, or equal to $p(t \mid \emptyset (C_0))$? Why?
Answer:
\[
p(t \mid \theta(C_0)) = \frac{c(t, C_0)}{\sum_{t \in C} c(t', t_0)}, \quad p(t \mid \theta(C_1)) = \frac{c(t, C_0) - c(t, D_1)}{\sum_{t \in C} (c(t', t_0) - c(t', D_1))}
\]

如果词t在文档集C0语言模型分配到的概率小于t在文档D1中分配到的概率，去掉这个文档势必会减小减小词t在新的文档集C1中的概率，反之亦然。所以:

If \( p(t \mid \theta(C_0)) > p(t \mid \theta(D_1)) \), then \( p(t \mid \theta(C_1)) > p(t \mid \theta(C_0)) \)

Else If \( p(t \mid \theta(C_0)) = p(t \mid \theta(D_1)) \), then \( p(t \mid \theta(C_1)) = p(t \mid \theta(C_0)) \)

If \( p(t \mid \theta(C_0)) < p(t \mid \theta(D_1)) \), then \( p(t \mid \theta(C_1)) < p(t \mid \theta(C_0)) \)

(d) [5 points] Is \( p(t \mid \theta(C_2)) \) higher than, lower than, or equal to \( p(t \mid \theta(C_0)) \)? Why?

\[
p(t \mid \theta(C_2)) = \frac{c(t, C_0) - c(t, D_1) + 1}{\sum_{t \in V'} (c(t', t_0) - \sum_{t \in V'} (c(t', D_1) - 1))}, \quad \text{where } V' = \{ t \mid c(t, D_1) > 1 \}, c(t,D_1) > 1
\]

If \( p(t \mid \theta(C_2)) < \frac{c(t, D_1) - 1}{\sum_{t \in V'} (c(t', D_1) - 1)} \), then \( p(t \mid \theta(C_2)) < p(t \mid \theta(C_0)) \)

Else if \( p(t \mid \theta(C_2)) = \frac{c(t, D_1) - 1}{\sum_{t \in V'} (c(t, D_1) - 1)} \), then \( p(t \mid \theta(C_2)) = p(t \mid \theta(C_0)) \)

else \( p(t \mid \theta(C_2)) > \frac{c(t, D_1) - 1}{\sum_{t \in V'} (c(t', D_1) - 1)} \), then \( p(t \mid \theta(C_2)) > p(t \mid \theta(C_0)) \)

3. [15 points] Entropy and Bayes Rule

Consider the following structurally ambiguous sentence

*John saw a student with a telescope.*

The prepositional phrase (PP) “with a telescope” can be attached to either the verb “saw” or the noun “student”. One approach to resolving such an ambiguity (called PP-attachment.) is to treat it as a statistical classification task. The instance to classify is the PP “with a telescope”, and the class labels are “attached to the verb” and “attach to the noun” respectively. We will represent the PP with two features that may help predict the class label: (1) the preposition (i.e., with in this case); (2) the head noun (i.e., telescope in this case). For example, we know that if the preposition is of then the PP should almost always be attached to the noun. Similarly, if the word telescope were replaced by book., then the PP with a book. is also unlikely attached to the verb. Suppose we consider only two prepositions “with” and “of”, and only two nouns “telescope” and “book”. We thus deal with three binary random variables:

- Class label: \( C \in \{0, 1\} \) with \( C = 1 \) (or \( C = 0 \)) indicates that the PP should be attached to the verb.
(or the noun).

- Preposition feature: $F_p \in \{\text{with, of}\}$ indicates which preposition is used in the PP under consideration.
- Head noun feature: $F_n \in \{\text{telescope, book}\}$ indicates which noun is the head noun in the PP under consideration.

Suppose we have 10 example sentences for which we know the correct PP attachment. Their class labels and feature values are show in the following table:

<table>
<thead>
<tr>
<th>PPAttachedTo (C)</th>
<th>PrepositionFeature ($F_p$)</th>
<th>Head noun feature ($F_n$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb</td>
<td>with</td>
<td>telescope</td>
</tr>
<tr>
<td>Verb</td>
<td>with</td>
<td>telescope</td>
</tr>
<tr>
<td>Verb</td>
<td>with</td>
<td>book</td>
</tr>
<tr>
<td>Verb</td>
<td>of</td>
<td>telescope</td>
</tr>
<tr>
<td>Verb</td>
<td>with</td>
<td>telescope</td>
</tr>
<tr>
<td>Noun</td>
<td>of</td>
<td>book</td>
</tr>
<tr>
<td>Noun</td>
<td>of</td>
<td>book</td>
</tr>
<tr>
<td>Noun</td>
<td>of</td>
<td>book</td>
</tr>
<tr>
<td>Noun</td>
<td>of</td>
<td>telescope</td>
</tr>
</tbody>
</table>

(a) (4 points) Use the maximum likelihood estimator to compute the following probabilities:

- $p(C = 1) = 0.5$
- $p(F_p = \text{with}) = 0.4$
- $p(F_p = \text{with} | C = 1) = 0.8$
- $p(F_p = \text{with}, F_n = \text{telescope} | C = 1) = 0.6$

(b) (3 points) Consider the entropies of the two binary random variables $F_p$ and $F_n$ (i.e., $H(F_p); H(F_n)$). Without calculating the entropies, can you tell which is larger? Why?

Answer:
$F_n > F_p$, 因为$F_n$的两个取值中每个值出现概率均为0.5，而$F_p$的两个取值出现概率更确定些（0.4, 0.6），所以从取值确定性上说$H(F_n) > H(F_p)$。

(c) (3 points) Now consider the three conditional distributions $p(F_p | C = 1)$, $p(F_n | C = 1)$, and $p(F_p | C = 0)$. We may also compute their entropies (i.e., $H(F_p | C = 1); H(F_n | C = 1); H(F_p | C = 0)$). Which of the three entropies is the smallest? Why?

Answer:

- $p(F_p | C = 1) = \{0.8, 0.2\}$
- $p(F_n | C = 1) = \{0.8, 0.2\}$
- $p(F_p | C = 0) = \{1\}$

$H(F_p | C = 0) = 0$ 最小，因为最确定

(d) (5 points) Given a new sentence $Q = \text{John saw the price of a book.}$ and assuming that the observation of both $F_p$ and $F_n$ is independent of each other given a particular PP attachment, can you use Bayes Rule to infer whether the preposition .of a book. should be attached to the verb .saw. or the noun .price.? Show your calculation.

Answer:
即计算 $P(C=1 | F_p=\text{of}, F_n=\text{book})$是否大于$P(C=0 | F_p=\text{of}, F_n=\text{book})$
设 $F_p = pf, F_n = book$ 为 $Q$

$$
\frac{P(C = 1 | Q)}{P(C = 0 | Q)} = \frac{P(Q | C = 1)P(C = 1)}{P(Q | C = 0)P(C = 0)} = \frac{P(F_p = of, F_n = book | C = 1)P(C = 1)}{P(F_p = of, F_n = book | C = 0)P(C = 0)}
$$

$$
= \frac{0.2 \times 0.2 \times 0.5}{1 \times 0.8 \times 0.5} < 1
$$

有 $P(C=1 | Q) < P(C=0 | Q)$

所以应 attach 到名词 price 后面。

4. [25 points] Dirichlet prior smoothing and retrieval

(a) [5 points] Let $q = q_1...q_m$ be a query and $d$ be a document and $p(q_i | d)$ be the probability of query word $q_i$ according to a smoothed document language model estimated based on $d$. Does the formula

$$
p(q | d) = \sum_{i=1}^{m} p(q_i | d)
$$

correctly describe the query likelihood retrieval model? If not, write down the correct formula.

Answer: 错

$$
p(q | d) = \prod_{i=1}^{m} p(q_i | d)
$$

Suppose we have a document collection with an extremely small vocabulary of only 8 words $w_1, ..., w_8$. The following table shows the estimated reference language model $p(w \mid REF)$ using the whole collection (2nd column) and the word counts $c(w; d)$ in document $d$ (3rd column). The 4th and 5th columns are two unigram language models for document $d$ estimated using the unsmoothed maximum likelihood estimator and Dirichlet prior smoothing (with parameter $\mu$), respectively.

<table>
<thead>
<tr>
<th>Word</th>
<th>$p(w \mid REF)$</th>
<th>$c(w; d)$</th>
<th>$p_{\text{ml}}(w \mid d)$</th>
<th>$p_\mu(w \mid d)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>0.3</td>
<td>2</td>
<td>0.2</td>
<td>&gt;</td>
</tr>
<tr>
<td>$w_2$</td>
<td>0.15</td>
<td>1</td>
<td>0.1</td>
<td>&gt;</td>
</tr>
<tr>
<td>$w_3$</td>
<td>0.1</td>
<td>2</td>
<td>0.2</td>
<td>0.125</td>
</tr>
<tr>
<td>$w_4$</td>
<td>0.1</td>
<td>4</td>
<td>0.4</td>
<td>&lt;</td>
</tr>
<tr>
<td>$w_5$</td>
<td>0.05</td>
<td>1</td>
<td>0.1</td>
<td>&lt;</td>
</tr>
<tr>
<td>$w_6$</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>&gt;</td>
</tr>
<tr>
<td>$w_7$</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>&gt;</td>
</tr>
<tr>
<td>$w_8$</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>&gt;</td>
</tr>
</tbody>
</table>

(b) [5 points] Fill in the values for unsmoothed maximum likelihood probabilities $p_{\text{ml}}(w \mid d)$ for all the eight words (column 4).
(c) [5 points] Column 5 is the probability of a word after applying Dirichlet prior smoothing with prior sample size parameter \( \mu \). In the table, only the smoothed probability for word \( w_3 \) is shown, which is 0.125. What is the value of \( \mu \)?

Answer:

\[
P_\mu(w \mid d) = \frac{c(w, d) + P(w \mid \text{REF}) \times \mu}{|d| + \mu} = \frac{2 + 0.1\mu}{10 + \mu} = 0.125, \text{ then } \mu = 30
\]

(d) [5 points] For the rest seven words on column 5, without actually computing the smoothed probability values, decide whether the smoothed probability is larger than, equal to, or smaller than the original unsmoothed maximum likelihood estimate on column 4. Use one of \{>, =, <\} to finish the empty cells in column 5.

(e) [5 points] What condition should \( c(w; d) \) satisfy so that the smoothed probability of word \( w \) would always be the same as the unsmoothed maximum likelihood estimate regardless what value \( \mu \) is?

Answer:

要使

\[
\frac{c(w, d) + P(w \mid \text{REF}) \times \mu}{|d| + \mu} = \frac{c(w, d)}{|d|}
\]

当

\[P(w \mid \text{REF}) = \frac{c(w, d)}{|d|}\]时，满足上式。

即词汇\( w \)在\( d \)中分布和p(\( w \mid \text{REF} \))相同时满足题意。

5. [10 points] Feedback

Web search engines can record which pages in the search results are viewed and which pages are skipped by a user. Such information is often called .clickthroughs.. Over time, we can thus collect a set of tuples of the following form:

(query1, URL1, skipped)
(query1, URL2, viewed)
(query1, URL3, viewed)
...
(query2, URL4, skipped)
(query2, URL2, viewed)
...
(query3, URL5, skipped)
(query3, URL3, skipped)
...
(query1, URL1, skipped)
(query1, URL2, skipped)
(query1, URL3, viewed)
...

Suppose a current user has justed entered a query which is identical to .query1.. A straightforward way
to use the past clickthrough data to improve ranking of pages for this query is to retrieve all the clickthroughs for .query1. from the query log as shown above and then simply rank the URLs that are viewed frequently. However this strategy does not take advantage of the clickthroughs of related queries. For example, if .query1. and .query2. are similar, the clickthroughs for .query2. may also be potentially interesting to the user who entered .query1.. Can you propose an algorithm to further improve performance by exploiting the clickthroughs of such related queries? Briefly sketch the algorithm, preferably with some formulas.

Answer:
We can score the url by related queries.
1) find k nearest neighbor queries, similarity between different queries can be calculated by the clickthrough similarity.
\[ V(q_i) = <r_{i1}, ..., r_{iN}> \], \( r_i \) means whether URLi is viewed or skipped
Then \( \text{sim}(q_i, q_j) = \cos (v(q_i), v(q_j)) \)
2) score url by the KNN queries

\[
\text{score}(q_j, URL_k) = \frac{1}{Z} \sum_{q_j \in \text{KNN}(q_i)} \text{sim}(q_i, q_j) \times \text{score}(q_j, URL_k)
\]

0 skipped
1 viewed
average unknown

\[ Z \] is is normalized

6. [10 points] Mixture models
Suppose a mixture model for documents has two component unigram language models \( \theta_1 \) and \( \theta_2 \) and the mixing coefficients are \( \lambda_1 \) and \( \lambda_2 \) for \( \theta_1 \) and \( \theta_2 \), respectively. Of course \( \lambda_1 + \lambda_2 = 1 \). Consider the following two different ways to “generate” a document \( D = d_1d_2...d_n \).

- Model 1
\[
p(D \mid \theta_1, \theta_2, \lambda_1, \lambda_2) = \lambda_1 \prod_{i=1}^{n} p(d_i \mid \theta_1) + \lambda_2 \prod_{i=1}^{n} p(d_i \mid \theta_2)
\]

- Model 2
\[
p(D \mid \theta_1, \theta_2, \lambda_1, \lambda_2) = \prod_{i=1}^{n} (\lambda_1 p(d_i \mid \theta_1) + \lambda_2 p(d_i \mid \theta_2))
\]

Given a collection of m documents \( C = \{D_1,...,D_m\} \) generated using such a mixture model, we often use the EM algorithm to estimate all the parameters of our mixture model, in which we augment the observed data with hidden variables. Each hidden variable is binary and indicates whether \( \theta_1 \) or \( \theta_2 \) is chosen. In which case, Model 1 or Model 2, would we need to introduce more distinct hidden variables? How many distinct hidden variables do we introduce exactly for estimating Model 1?

Answer:
Model1:
There are m documents, each of which has a topic, so there are m topics (m hidden variables)

Model2:
For each word in a document, there is a topic, so there are m*n topics (hidden variables)

(Model1: 先 random 出一个 topic，然后根据这个 topic random 出每个 term。Model2: 对于每个 term，先 random 一个 topic，再根据这个 topic random 出一个 term。显然，第二个 model 的表达能力更强，需要的隐变量也更多。对于第一个 model，隐变量就是一个 document 的一个 topic。对于第二个 model，隐变量是每个文档每个 term 对应的 topic。)

7. [10 points] Open question (proximity)
When two documents match the same number of query terms, intuitively we should prefer a document in which all the matches are close to each other to a document in which the matches are far away (e.g., one at the very beginning of the document and one at the end) because in the latter case, the matches may be due to chances as opposed to a coherent discussion of the topic. Can you propose a way to extend the query likelihood retrieval function to incorporate such a proximity heuristic?

Answer:
\[ P(q \mid d, R = 1) \approx (\prod_i p(q_i \mid d, R = 1)) \times p_{\text{prox}}(q \mid d, R = 1) \]

Ignore R=1,
\[ \log p(q \mid d) = \sum \log p(q_i \mid d) + \log p_{\text{prox}}(q \mid d) \]

We can assume \( p_{\text{prox}}(q \mid d) \) is independent on the context of document d, but only dependent on the distances between query terms. So some heuristic formula can be used, such as
\[ p_{\text{prox}}(q \mid d) \propto p(q \mid \min_{i,j} d(q_i, q_j)) \quad \text{or} \quad p_{\text{prox}}(q \mid d) \propto p(q \mid \text{mean}_{i,j} d(q_i, q_j)) \]

The probability can be estimated on training data set by logistic regression.