HW 3

Doris Jung-Lin Lee

1)a) In the RSJ derivation, we showed that the order depends on the odds-ratio of relevance and irrelevance:

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

Instead of using binary presence $P(A_i|Q,R)$ for our document generation model P(D|Q,R), we will now have the multinomial model:

$$P(D|Q,R) = \prod_{i=1}^{|V|} P(w_i|d)^{c(w_i,q)}$$

$$score \propto \prod_i \frac{P(w_i|Q,R=1)^{c(w_i,D)}}{P(w_i|Q,R=0)^{c(w_i,D)}}$$

$$= log\left(\prod_i \frac{P(w_i|Q,R=1)^{c(w_i,D)}}{P(w_i|Q,R=0)^{c(w_i,D)}} \right)$$

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

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

Since we know about the $\sum_i p(w_i|d) = 1$ constraint, so only V-1 parameters are required, where V is the number of vocabularies in the document.

b) Let c be counts of the word occurence in the document, $\mathbf{C}$ refers to the collection $D_1 \ldots D_n$. Using MLE, we can write the multinomial distribution likelihood as :

$$P(C = D_1 \ldots D_n|\theta) = \prod_i P(w_i|\theta)^{c(w_i,\mathbf{C})}$$

$$logP(C = D_1 \ldots D_n|\theta) = \sum_i c(w_i,\mathbf{C}) logP(w_i|\theta)$$

Using Langrange multiplier, the cost becomes :

$$\mathscr{L} = \sum c(w, \mathbf{C}) log P(w|\theta) + \lambda \left[1 - \sum P(w|\theta)\right]$$

Taking the derivative with respect to $P(w|\theta)$ yields two equations, solving those yields the ML estimate:

$$\therefore P(w|\hat{\theta}) = \frac{c(w, \mathbf{C})}{|\mathbf{C}|} = \frac{c(w, \mathbf{C})}{n}$$

.

c) If there is only one example,

$$\mathscr{L}(\theta, Q = x_1 \dots x_n) = \prod_n f(x_1 \dots x_n|\theta)$$

where $x_1 \dots x_n$ is the words in the query Q.

$$P(w|Q, R = 1) = P(w|\hat{\theta}) = \frac{\sum_{x \in Q} c(w, x)}{|Q|}$$

d) We apply linear interpolation to our MLE estimate P(w|Q,R=1) using Jelinek-Mercer:

$$P_\lambda(w|\hat{\theta}) = (1 - \lambda)\frac{c(w, D)}{|D|} + \lambda p(w|C)$$

with p(w|C) as the collection language model.

e)

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

$$= \sum_{w \in V} c(w, D) log \frac{(1 - \lambda)\frac{c(w,D)}{|D|} + \lambda p(w|C)}{\frac{c(w,\mathbf{C})}{|\mathbf{C}|}}$$

$$= \sum_{w \in V} c(w, D) log \left[\frac{|C|}{c(w, \mathbf{C})} \left[(1 - \lambda)\frac{c(w, D)}{|D|} + \lambda p(w|C) \right] \right]$$

TF : term frequency of the document is captured by c(w,D), it acts as a weight inside the summation infront of every log term. Intuitively, high TF documents should be ranked higher.

IDF : The inverse term frequency of the document is captured by $log(\frac{|C|}{c(w,\mathbf{C})})$ which could be extracted out from the log term. If the word is common across the collection, then the IDF would be small, thus IDF weighting term discounts the rank based on seeing occurence of word w in document D.

Document length normalization : Longer documents (e.g. papers, essays) should be penalized since it has more content (thus on average a higher word frequency) than compared to shorted documents such as abstracts. This is accounted for by the the term $score\frac{c(w,D)}{|D|}$, since we are normalizing based on the document length, each word in the docuement contribute equally to the document. In addition, the document length normalization is implicitly encoded inside the smoothing coefficient λ because longer document require less smoothing (because they will have less zero-word-count occurences) and shorter documents will require more smoothing.

2) a) Using query likelihood model:

$$score(Q,D) = logp(q|d) = \sum c(w,Q)logp(w_i|d)$$

For JM Smoothing:

$$p(w|d) = (1-\lambda)\frac{c(w,d)}{|d|} + \lambda p(w|REF)$$

$$p(w|d) = \lambda p(w|REF)\left[1 + \frac{c(w,d)}{|d|\lambda p(w|REF)}\right]$$

For ranking purposes we can ignore $\lambda p(w|REF)$ because they are independent of Q,D.

$$score(Q,D) = \sum_{w \in Q \bigcap D} c(w,Q)log\left[1 + \frac{c(w,d)}{|d|\lambda p(w|REF)}\right]$$

b) In the vector space model, both the query and docuemnt vectors would have bag-of-words representations. The document and query vector would be $d = [w_1 \ldots w_n]$,$q = [w_1 \ldots w_n]$ where w_i indicates term frequency. Cosine simmilarity between the query and the document vectors should be computed. The TF-IDF weighting is captured by the term weight c(w,D) and IDF is captured by the reference language model p(w|REF) (since intuitively, rare/common words should be reflected in the reference language model) . The document length normalization is captured by the smoothing parameter λ.

c) Check if score(Q,D) = score (Q,D') where D' = kD for JM Smoothing:

$$score(Q,D') = \sum_{w \in Q \bigcap D'} c(w,Q)log\left[1 + \frac{c(w,D')}{|D'|\lambda p(w|REF)}\right]$$

We know that λ, c(w,Q), p(w|REF) are same for both scoring function. For the summation:

$$\sum_{w \in Q \bigcap D'} = \sum_{w \in Q \bigcap D}$$

since D' doesn't provide additional information compared to D, the subset that intersect with Q is the same.

We also know that |D'| = k|D| and c(w,D') = kc(w,D), so

$$score(Q, D') = \sum_{w \in Q \bigcap D'} c(w, Q) log \left[1 + \frac{kc(w, D)}{k|D|\lambda p(w|REF)} \right]$$

The factors of k cancel out, ∴ $\boxed{\text{score (Q,D')} = \text{score(Q,D)}}$ for JM smoothing.

For Dirichlet smoothing:

$$P(w_i|D) = \frac{c(w_i) + \mu P(w_i|REF)}{|D| + \mu}$$

The only thing that is diffference between score (Q,D') and score(Q,D) is that |D'| =k|D| in the denominator. So $\boxed{\text{score(Q,D')<score(Q,D)}}$ by a factor, ∴ Dirichlet smoother overpenalizes long documents.