CS54701-Homework2

Peng Li

February 24, 2016

1 Boolean Retrieval can exact match with the query and use many operators such as logical operators, proximity operator, field operator. For example, when we want to search some sport news but we really don't like soccer, our query is (sport news) AND NOT (soccer) for a boolean retrieval. And I think it is difficult to do this using a vector space model. And if we want to search the document which title has a certain word, we can use boolean retrieval ti query in this way: title(retrieval model). If we query "retrieval model" with a vector space model, we will get more documents we don't want.

2.A TF-IDF formula

l = $ln(tf_{t,d}) + 1$
t = $ln(\frac{N}{df_t})$
c = $\frac{1}{\sqrt[2]{w_1^2+w_2^2+w_3^2}}$
$tf_{t,d}$ = the term occurrence in document / length of document
N = number of all paper
df_t = count of paper has this word
TF-IDF = TF * IDF

2.B

	DocA
CS54701	0.7783
Final	0.6279
May	0

3.A

	DocC
CS54701	0.179
Final	0.462
May	0.359

I think μ is average length of document 6

3.B First, we calculate all words probability of whole collection use unigram language model.

$$p_c = \frac{\sum_i tf_i(w_k)}{\sum_i |\vec{d_i}|}$$

Next, we smooth the unigram model of each document with Dirichlet smoothing.

$$p_i(w_k) = \frac{tf_i(w_k) + \mu p_c(w_k)}{|\vec{d_i} + \mu|}$$

Finally, we compute the vector q in vocabulary space.

$$\vec{q} = \{tf_q(w1), ..., tf_q(w_k)\}$$

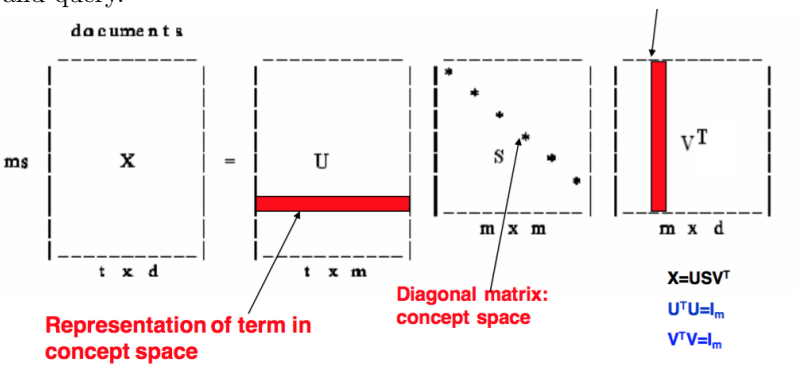
Computing each document likelihood,

$$p(q|\vec{(}d_i)) = \prod [p(w_i|\vec{d_i})]^{tf_q(w_k)} = \prod_{k=1}^{K} \frac{tf_i(w_k) + \mu p_c(w_k)}{\vec{d_i} + \mu}$$

So, the higher result we get, means the more relevant the document is to the query.

3.C This is false. Because when calculate a probability of a document to decide if this document is what user need, we do not think more about the relation between query words, just compare it with query words. In this way, what user get maybe have less relevant with the user's need. So, what we get may not meet the users need.

4 When we have a term-document matrix, we do a singular value decomposition to it and we will get three matrix, they are U, S, Vt. We do this because we want to reducing the number of dimensions. We group up this documents on similar topics so even if a query not exact match the documents.
The rows in U are vectors represent term in concept space and the columns in Vt are vectors represent documents in concept. We will also use the function:$\vec{q} = q_t U_k Inv(S)$ to get the vector which represent the query in concept space. After we get that, we can use it and compute the cosine similarity between documents and query.



5.A

$$\begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 \end{pmatrix} \tag{1}$$

5.B $M = stochastic\ matrix$
$\vec{x_0} = (1, 0, 0, 0)$
$\vec{x_1} = (0, 1/3, 1/3, 1/3)$
$\vec{x_2} = (1/3, 0, 1/2, 1/6)$
$\vec{x_3} = (1/3, 1/9, 7/36, 13/36)$

5.C

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

D has no out-link and there isn't a random walk. To fix it, when move to D, just go to one node randomly and this probability is 1/4