Your Name: Shuang Xu

Your Andrew ID: sxu1

Homework 4

Collaboration and Originality

1. Did you receive help of any kind from anyone in developing your software for this assignment (Yes or No)? It is not necessary to describe discussions with the instructor or TAs.
   No
   If you answered Yes, provide the name(s) of anyone who provided help, and describe the type of help that you received.

2. Did you give help of any kind to anyone in developing their software for this assignment (Yes or No)?
   No
   If you answered Yes, provide the name(s) of anyone that you helped, and describe the type of help that you provided.

3. Are you the author of every line of source code submitted for this assignment (Yes or No)? It is not necessary to mention software provided by the instructor.
   Yes
   If you answered No:
   a. identify the software that you did not write,
   b. explain where it came from, and
   c. explain why you used it.

4. Are you the author of every word of your report (Yes or No)?
   Yes
   If you answered No:
   a. identify the text that you did not write,
   b. explain where it came from, and
   c. explain why you used it.

Your Name: Shuang Xu

Your Andrew ID: sxu1

Homework 4

1 Experiment: Baselines

Provide information about the effectiveness of your system in three baseline configurations.

	BM25	Indri BOW	Indri SDM
P@10	0.216	0.204	0.236
P@20	0.248	0.268	0.274
P@30	0.2573	0.2653	0.2867
MAP	0.1334	0.1462	0.1441

Document the parameter settings that were used to obtain these results.

BM25:k_1=1.2
BM25:b=0.75
BM25:k_3=0
Indri:mu=2500
Indri:lambda=0.4
Indri SDM:0.5 (bow query) 0.25 (#near/1 query) 0.25 (#window/8 query)

2 Custom Features

Describe each of your custom features, including what information it uses and its computational complexity. Explain the intuitions behind your choices. This does not need to be a lengthy discussion, but you need to convince us that your features are reasonable hypotheses about what improves search accuracy, and not too computationally expensive to be practical.

Feature 17: tfidf for <q, d_{body}>. Time complexity is O(k) where k is the number of query terms (after stop and stem). I believe those query terms that are frequent in document and can also distinguish the document from other documents would be more helpful.

Feature 18: total tf for <q, d_{body}>, which is total term frequency of query terms normalized by the total fields length of the document. Time complexity is O(k) where k is the number of query terms (after stop and stem). Term frequency itself is a good indicator as it partly shows how similar the document is with the query, but tf is not included in the previous 16 features.

3 Experiment: Learning to Rank

Use your learning-to-rank software to train four models that use different groups of features.

	IR Fusion	Content-Based	Base	All
P@10	0.272	0.284	0.272	0.264
P@20	0.28	0.284	0.28	0.276
P@30	0.2707	0.276	0.2747	0.2707
MAP	0.1231	0.1197	0.1199	0.1191

Discuss the trends that you observe; whether the learned retrieval models behaved as you expected; how the learned retrieval models compare to the baseline methods; and any other observations that you may have.
Also, discuss the effectiveness of your custom features. This should be a separate discussion, and it should be more insightful than "They improved P@10 by 5%". Discuss the effect on your retrieval experiments, and if there is variation in the metrics that are affected (e.g., P@k, MAP), how those variations compared to your expectations.

Learned retrieval models produce higher precision but lower MAP. The best model with content-based features improved P@10 by 31.5% compared with BM25, 39.2% compared with Indri bow, 20.3% with Indri SDM. But the cost is MAP. The content-based features lowered down MAP by 10% compared with BM25, 18% compared with Indri bow, 17% compared with Indri SDM. I suppose the main reason precision improved is that SVM combined more features (query-independent features, indri and bm25 score for different fields) into one model so that the model is more comprehensive. Features like query-independent features like spam score and wiki score represent the quality of document itself and may produce more precise results. See the following 3 query-independent support features, we can prove that, larger spam score, smaller URL depth and Wikipedia url correlates with high-relevance documents.

1:0.405121 2:-0.27732182 3:0.55482072

I suppose base and all settings would get best performance but apparently they don't. So there must be some bad features. For learning retrieval models, MAP is quite stable while P@10 varies a lot. I think bad features like combination of feature 1-4, those feature vectors contribute little to the model. So it would slightly change the ranking of the very first document but may not affect other document or the whole set of 100 documents very much.

It seems combination of feature 1-4 and feature 17-18 are harmful. Content-based features wins.

4 Experiment: Features

Experiment with four different combinations of features.

	All (Baseline)	**$Comb_1$**	**$Comb_2$**	**$Comb_3$**	**$Comb_4$**
P@10	0.264	0.288	0.272	0.28	0.264
P@20	0.276	0.28	0.28	0.284	0.278
P@30	0.2707	0.2773	0.2707	0.2733	0.268
MAP	0.1191	0.1202	0.1213	0.1209	0.119

Describe each of your feature combinations, including its computational complexity. Explain the intuitions behind your choices. This does not need to be a lengthy discussion, but you need to convince us that your combinations are investigating interesting hypotheses about what delivers good search accuracy. Were you able to get good effectiveness from a smaller set of features, or is the best result obtained by using all of the features? Why?

I found according to the previous experiment, content-based features got best precision and recall results, so based on content based features, I removed feature vectors that are smaller than 0.1($Comb_{1:}$) and 0.3($Comb_2$). I think these features are less related with the classification model. Thus, I chose the following two combinations.

$Comb_{1:}$ with 10 feature. letor:featureDisable=1,2,3,4,17,18,14,16
$Comb_{2:}$ with 8 features. letor:featureDisable=1,2,3,4,17,18,14,16,10,7

Observe the SVM model of all features:
1:0.405121 2:-0.27732182 3:0.55482072 4:0.087131381 5:-0.68345273 6:-0.94934827 7:0.26710075 8:1.2629287 9:-0.79209358 10:0.26227102 11:0.60811841 12:0.4484852 13:-0.49270934 14:0.08810395 15:-0.49375722 16:0.0013776645 17:0.26710075 18:0.26710075

Then order the absolute value of each feature vector in ascending order as follows:
[('16', 0.0013776645), ('4', 0.087131381), ('14', 0.08810395), ('10', 0.26227102), ('17', 0.26710075), ('18', 0.26710075), ('7', 0.26710075), ('2', 0.27732182), ('1', 0.405121), ('12', 0.4484852), ('13', 0.49270934), ('15', 0.49375722), ('3', 0.55482072), ('11', 0.60811841), ('5', 0.68345273), ('9', 0.79209358), ('6', 0.94934827), ('8', 1.2629287)]

Combination 3 and 4 are based on above observation. I tried to remove some features that are less important.

$Comb_3$: with 10 features. Disable first 8 features (letor:featureDisable=16,4,14,10,17,18,7,2)
$Comb_4$: with 15 features. Disable first 3 features (letor:featureDisable=16,4,14)

The best results are obtained by a smaller set of features ($comb_2$ with 10 features). Because some features are quite confusing and not convincing. For example, I found feature 4 in some model is negative and in others are positive, which indicates it a bad feature as the relation with the feature and final classification is not sure. By eliminating those features and also the features that are less important, would improve the performance. But over-eliminating, would produce worse performance.

The computational complexity for the above combinations is O(100nk) where n is the number of queries and k is the average length of query.

5 Analysis

Examine the model files produced by SVM^{rank}. Discuss which features appear to be more useful and which features appear to be less useful. Support your observations with evidence from your experiments. Keep in mind that some of the features are highly correlated, which may affect the weights that were learned for those features.

Some of this discussion may overlap with your discussion of your experiments. However, in this section we are primarily interested in what information, if anything, you can get from the SVM^{rank} model files.

SVM model of all features:

1:0.405121 2:-0.27732182 3:0.55482072 4:0.087131381 5:-0.68345273 6:-0.94934827 7:0.26710075 8:1.2629287 9:-0.79209358 10:0.26227102 11:0.60811841 12:0.4484852 13:-0.49270934 14:0.08810395 15:-0.49375722 16:0.0013776645 17:0.26710075 18:0.26710075

Order the absolute value of each weight vector in ascending order as follows:

[('16', 0.0013776645), ('4', 0.087131381), ('14', 0.08810395), ('10', 0.26227102), ('17', 0.26710075), ('18', 0.26710075), ('7', 0.26710075), ('2', 0.27732182), ('1', 0.405121), ('12', 0.4484852), ('13', 0.49270934), ('15', 0.49375722), ('3', 0.55482072), ('11', 0.60811841), ('5', 0.68345273), ('9', 0.79209358), ('6', 0.94934827), ('8', 1.2629287)]

5 least important features:

Feature 16 (term overlap score for $<q, d_{inlink}>$),

Feature 4 (PageRank score for d),

Feature 14 (BM25 score for $<q, d_{inlink}>$)

Feature 10 (term overlap score for $<q, d_{title}>$),

Feature 17 (tfidf score for $<q, d_{body}>$),

5 most important features

Feature 8 (BM25 score for $<q, d_{title}>$),

Feature 6(Indri score for $<q, d_{body}>$),

Feature 9(Indri score for $<q, d_{title}>$) ,

Feature 5(BM25 score for $<q, d_{url}>$) ,

Feature 11(BM25 score for $<q, d_{url}>$)

We can find that for such experiments, inlink field is least useful filed and title and url field are most important fields. This makes sense because inlink field contains little information, many documents don't contain inlink or the inlink is not related to document at all (eg. inlink for people to jump to the main page). Title and url fields are best abstraction for document content and there's no surprise that they are important. Term overlap score in most cases is not useful. And PageRank score is not as useful as expected.

The 5 most important features are all content-based features, this corresponds with Experiment 3.

Feature 5 and feature 6 are negatively related with document relevance which means smaller BM25 score for $<q, d_{url}>$, smaller Indri score for $<q, d_{body}>$ correlates with high-relevance documents. This is amazing and I cannot figure out why.