Your Name

Your Andrew ID

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.

   If you answered Yes, provide the name(s) of anyone who provided help, and describe the type of help that you received.
   No

2. Did you give help of any kind to anyone in developing their software for this assignment (Yes or No)?

   If you answered Yes, provide the name(s) of anyone that you helped, and describe the type of help that you provided.
   No
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.

   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.

   Yes
4. Are you the author of every word of your report (Yes or No)?

   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.

Yes

Your Name Raghav Sharma

Your Andrew ID rvsharma

Homework 4

Instruction

1 Experiment: Baselines

	BM25	Indri BOW	Indri SDM
P@10	0.2160	0.2120	0.2160
P@20	0.2400	0.2740	0.2740
P@30	0.2493	0.2680	0.2853
MAP	0.1239	0.1327	0.1331

Settings for BM25
trecEvalOutputLength=100
retrievalAlgorithm=BM25
BM25:k_1=1.2
BM25:b=0.75
BM25:k_3=0

Settings for Indri-BOW
retrievalAlgorithm=Indri
Indri:mu=2500
Indri:lambda=0.4

Settings for Indri – SDM
retrievalAlgorithm=Indri
Indri:mu=2500
Indri:lambda=0.4
Weight for AND = 0.7
Weight for NEAR = 0.1
Weight for WINDOW = 0.1

2 Custom Features

Custom Feature 1: Query Term Density:

This measures the query stems' document frequency in each document. This is to determine how important is the document to the query and may be useful to increase the documents score when we consider the query document pair. It will rank the documents which contain the query tokens higher in the final results.

To calculate we use the term vector of the document to determine the index of the query stem. If the query stem exists in the document, we find out its document frequency by using termVector.stemDf(index) and sum that over all the query tokens. Averaging it over length of the query tokens will give us the score of that document for that query. A higher ranked document implies that the document contains more terms matching with the query stem than lower ranked documents.

In terms of computational complexity, this feature is not time taking as it is calculated in O(N) time, N being the number of query tokens in the query.

Custom Feature 2: Content Length

Next feature I chose is independent of the query terms. It calculates the length of the title field in each document. If the index doesn't contain the spam score (feature 2) for this document, then this is particularly helpful in detecting whether the document is spam or not. Spammy documents contain big title fields to spoof the search engine to get included in the search results.

We can calculate this feature using the Idx.getFieldLength("title", InternalDocId) and this require constant access in time. Thus adding this feature will not take much time in the query calculation.

3 Experiment: Learning to Rank

	IR Fusion	Content-Based	Base	All
P@10	0.2520	0.2680	0.4400	0.4440
P@20	0.2540	0.2700	0.3920	0.3920
P@30	0.2747	0.2987	0.3747	0.3720
MAP	0.1215	0.1261	0.1728	0.1728

3.1 Parameters

included files named HW4-Exp2-x.qry and HW4-Exp2-x.param in the QryEval directory, where x indicates a table column (fusion, content, base, all).

3.2 Discussion

IR fusion – This model takes features of BM25 and Indri features and combines them into one. Intuitively since this is a fusion of both models, we expect the IR fusion to be better than BM25 and Indri BOW. Though it performs better than BM25 and Indri Bow for P@n precisions, it fails in MAP which is a more robust accuracy evaluating parameter. This is not expected.

This may be due to the machine learning training. As we use query document pair with relevance judgement ranging from 0 to 4 (with 0 being not relevant and 4 being most relevant), the training algorithm is not capturing the differences between documents that are having the same relevance. The difference between the relevance level is 1 and the documents being judged is small and the scores are arbitrary. This makes predicting the score difficult. This causes learning algorithm to learn different models and leads to reduction in MAP.

Content based – In this feature set, the term overlap is also considered. This metric hugely improves the quality of results as evidenced by uniform increase in the P@n and MAP. Overlap indicates the percentage of query terms that match the document field. If a document contains more, number of query terms then it is more relevant to the query than with other documents that do not match the query terms. This leads to higher quality results as it puts these documents at the top of the ranking.

Base and All Feature models – These models include query independent features like PageRank, Spam score, Wikipedia pages, URL depth into the feature mix. These features are independent of the query and are highly correlated to the quality of the documents themselves. Spam scores eliminate the spammy content and documents from the results while a document having a higher page rank score indicates that people trust the documents from these websites/pages. A shorter URL indicates that URL is either a home page and closer to it, indicating it is frequently visited and trusted. Similarly, Wikipedia score is a good indicator of the reliability of the result. All these query independent features being included in the training and learning. All these features lead to an improvement of the P@n and MAP in the model.

For All, inclusion of customized feature such as ‘query term density’ and ‘Title field’ length further improves the P@n for the model. Query term density calculates the presence of stem words in the documents. This is much better indicator of the relevance of the document and is successful in. improving the precision of the P@10.

Thus in conclusion, Fusion of Indri and BM25 content based feature was expected to perform better but due to the noisy relevance data it got negatively effected. Query independent features however improved the MAP and P@n for the model, and are a good indicator of the relevance of the data set.

4 Experiment: Features

Experiment with four different combinations of features.

	All (Baseline)	$Comb_1$ = BM25 only	$Comb_2$ = Spam+Wiki+ BM25 <Body,Title>+ Overlap <Body,Title>	$Comb_3$ = Spam+Wiki +BM25+ Overlap <Body>	$Comb_4$ = qry Independent + Overlap
P@10	0.4440	0.2640	0.4200	0.3600	0.4360
P@20	0.3920	0.2700	0.3800	0.3360	0.4060
P@30	0.3720	0.2773	0.3640	0.3293	0.3893
MAP	0.1728	0.1220	0.1693	0.1638	0.1752

4.1 Parameters

included files named HW4-Exp4-x.qry and HW4-Exp4-x.param in the QryEval directory, where x indicates a table column (all, comb1, comb2, comb3, comb4).

4.2 Discussion

Combination 1: BM25 only,

Included Features: 5,8,11,14

Using the model file we observed that BM25 is given a higher weightage when compared to Indri, so to judege the effect of BM25 query-document features, I just enabled these features for body, title, url and inlink and left the other features disabled. I expected the P@n and MAP to not be comparable to baseline LeToR. Since LeToR contains query independent features and overlap percentages which is a major parameter to improve the search accuracy.

This feature traverses each query in all the documents thus, In terms of computational complexity the time complexity of this is O(MN) where m is the number of query terms and N is the number of document.

Combination 2: Spam + Wiki + BM25, Overlap <Body, Title>

Included features: 1, 3, 5, 7, 8, 10

Since in experiment 2 we noticed that Independent query features improved the MAP and P@n significantly. So, using the model file the most effective independent feature are Wikipedia pages and Spam score for the documents. Additionally according to earlier experiments Term overlap also improves MAP and P@n scores. When these features are added to BM25 we see 30% improvement in MAP and P@n improves by 45%. All these at the cost of 2 additional features, in the feature pool.

In terms of computational complexity BM25 and Overlap calculation takes O(MN) time where M is the number of query and N is the number of documents. The lookups for Spam is O(1) as it takes the spam score from index. Thus the overall complexity is O(MN) which is comparable to BM25 only feature model above.

This model though doesn't beat the baseline model of all features but is very close to that in terms of MAP and P@n. It is 2% less MAP than baseline and 5% less P@10 to the baseline model. This is a better achievement for the model since it only includes 6 features as opposed to 18 features included in All. So there must be some features that are negatively affecting the learning of the algorithm.

Combination 3: Spam + Wiki + BM25 + Overlap <Body>

Included feature: 1, 3, 5, 7

In an effort to reduce the number of features required and still maintain the P@n and MAP, I removed the content for "Title" field – BM25 and term overlap scores. This model only considers Wikipedia, spam score and body content features. However, the MAP and P@10 falls below the combination 2 and the biggest loss is to the P@10 score. It decreases by 16%. This might be due to removal of title and query document pair. Since the matching of title places the document at the top of the result since it becomes highly relevant document. By removal of this feature, the title field is no longer considered and thus P@10 falls. However, reduction in P@30 and MAP is not that considerable. Thus these feature models are also a good choice in case there is a limitation on number of features that are permitted in LeToR.

The time complexity is O(MN) where M = number of query terms and N = number of documents.

Combination 4: All Query independent features + Overlap<Body, Title, URL, Inlink>

Included Features: 1,2, 3, 4, 7, 10, 13, 16

By the above experiments it is evident that Overlap field will provide maximum gain to the learning algorithm. In the search of a better model than baseline, I included all the query independent features and Term overlap features taking an intersection of features that improved the MAP and P@n, in earlier experiments. The MAP and P@n for this model beat the baseline model by a narrow margin (P@10 by 5%, MAP by 1%).

This is due to the inclusion of PageRank score and length of the URL in addition to SpamScore and Wikipedia indicator. Both are the good indicator of the quality of a document, and improve the MAP since they are independent of the query.

Inclusion of Page Rank and length of URL doesn't change the time complexity of the model which is still O(MN), since the lookup for these scores is from index and is in constant time. Thus the computation complexity is similar to other LeToR feature models.

Customized features:

Inclusion of customized features such as 'query term density' and 'Title field length' further improves the P@n for the model. Query term density calculates the presence of stem words in the documents. This is

much better indicator of the relevance of the document and is successful in. improving the precision of the P@10.

The Query Term density is similar to overlap score for each fields but it calculates document frequency for each query token. This slightly improves the precision for top. Queries as seen by the improvement in P@10 when compared to base feature model.

On the other hand, title field length is similar to Spam score since it indicates the probability of the document being spam by measuring the title field. A higher spammy document has a longer title field length to improves its rank in search results. By adding this score we control the spam in the results if the spam score is not available for any document. This also improves the P@10 since a document that matches title field is placed at a higher rank than some document that only matches terms in a body field.

5 Analysis

	Feature	IR Fusion	Content Based	Base	All	1	2	3	4	5	6	7	8	9
1	Wikipedia	0.0000	0.0000	0.6477	0.7160	0.0000	0.0000	0.0000	0.7381	0.7076	0.9054	0.7657	0.6716	0.6839
2	Spam	0.0000	0.0000	0.3817	0.4484	0.0000	0.0000	0.0000	0.4112	0.3949	0.0000	0.4108	0.3953	0.3938
3	Overlap<q,dTitle>	0.0000	0.2976	0.2670	0.3712	0.0000	0.0000	0.4735	0.3584	0.3660	0.0000	0.0000	0.4005	0.3937
4	BM25<q,dBody>	0.4141	0.2920	0.2558	0.3613	0.4556	0.0000	0.0000	0.2843	0.2856	0.3831	0.3952	0.0000	0.0000
5	Overlap<q,dURL>	0.0000	0.2960	0.2240	0.2820	0.0000	0.0000	0.4584	0.0000	0.0000	0.0000	0.0000	0.3096	0.3165
6	BM25<q,dTitle>	0.4014	0.2158	0.2100	0.2831	0.4778	0.0000	0.0000	0.2757	0.2826	0.0000	0.0000	0.0000	0.0000
7	Overlap<q,dBody>	0.0000	0.2281	0.1792	0.2177	0.0000	0.0000	0.3626	0.1980	0.1951	0.2790	0.2873	0.2602	0.2584
8	BM25<q,dURL>	0.4137	0.2338	0.1658	0.2020	0.4993	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9	Indri<q,dURL>	0.2932	0.1576	0.0987	0.1314	0.0000	0.5182	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	Indri<q,dBody>	0.1700	0.1436	0.0756	0.0451	0.0000	0.2349	0.0000	0.0736	0.0000	0.0439	0.0000	0.0000	0.0000
11	Indri<q,dTitle>	0.1465	0.0653	0.0632	0.0051	0.0000	0.2729	0.0000	0.0803	0.0000	0.0000	0.0000	0.0000	0.0000
12	BM25<q,dInlink>	0.1308	0.0121	0.0535	0.1195	0.1945	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13	Overlap<q,dInlink>	0.0000	0.0154	0.0522	0.1335	0.0000	0.0000	0.0488	0.0000	0.0000	0.0000	0.0000	0.1338	0.1198
14	Term density	0.0000	0.0000	0.0000	0.1633	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	Doc frequency	0.0000	0.0000	0.0000	-0.1777815	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
16	Indri<q,dInlink>	-0.0125	-0.0235	-0.0035	-0.0149	0.0000	0.0097	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17	PageRank	0.0000	0.0000	-0.0339	-0.0234	0.0000	0.0000	0.0000	-0.0374	0.0000	0.0000	0.0000	0.0000	-0.0329
18	URL	0.0000	0.0000	-0.2055	-0.2145	0.0000	0.0000	0.0000	-0.2072	0.0000	0.0000	0.0000	0.0000	-0.2120

Using the perl file, I calculated all the weights for the features for IR fusion, content based, Base, ALL, Combination 1 – 4, and added them to the table given above. Then I sorted the weights in descending values of weights for All, base, combination4, combination 2, combination 1 and combination 3. The results of the sort appear above.

The noteworthy results are impact of query independent features: Wikipedia and Spam score for documents. These two query independent documents are better performing than the URL and PageRank query independent features, as seen by comparing their absolute values for All (Wikipedia = 0.71, Spam = 0.44, URL = 0.21, PageRank = 0.023). This is as expected and proven from prior experiments, that including the Wikipedia and Spam score increase the MAP and P@10 of the results. These provide a better improvement over anything content based.

When considering only content based features, most useful features are OverLap<q, dTitle> and BM25<q, dBody>. This is expected because Title and Body are two most useful field which are a good indicator of relevance of the document as evidenced in the prior experiments.

Additionally, BM25 and Overlap give the highest jump in the MAP and P@n scores. So the combination of BM25, overlap with Title and Body field are the most useful features in the bunch.

For the least useful content based features we have BM25<q,dInlink> and Overlap<q, dInlink>, this is due to Inlink containing least information and lowest indicator of the relevance of the document. Also inlink can be absent in most documents or be related to completely different content from the query.