Your Name Yan Zhao

Your Andrew ID yanzhao2

Homework 5

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 Yan Zhao

Your Andrew ID yanzhao2

Homework 5

1 Experiment: Baselines

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

	BM25	Indri BOW	Indri SDM
P@10	0.4080	0.3160	0.4320
P@20	0.4040	0.3420	0.4420
P@30	0.4013	0.3453	0.4413
MAP	0.2286	0.1949	0.2346

For BM25 model: k1 = 1.2, k2 = 0.75, k3 = 0

For Indri model: mu = 2500, lambda = 0.4

For SDM, I set weight of #AND = 0.3, weight of #NEAR = 0.4, weight of #WIN = 0.3, which is the parameter with best performance in my last assignment. Indri model parameter same as BOW.

2 Custom Features

Feature 17: Sequence Overlap Score

This feature is defined as number of query phrases that match "body" field of document. In specific, I search in "body" field of each document with a sliding window of size 8, to count the number of overlaps between terms within the window and query terms. Since I use a sliding window to scan each stems in the document, the computation complexity is O(n) where n is the length of stems for a document, and this complexity is the same as computing BM25 score for <q, d_{body}>. The intuition for this feature is to have a score similar with WINDOW operator. Based on previous experiments, WINDOW operator is quite useful on improving search results, and for most of the queries, the query terms are often combined together as phrases. So by searching a document with sliding window I can calculate the number of overlap phrases in the query that matches the document. Also, from computational point of view, I didn't use n-grams to search because it will cost much more time. Thus, I get this feature as a result of trade-off between accuracy and computation complexity.

Feature 18: Average DF Score

This feature is defined as average DF score of a document in "title" field. I use an online calculation method to go through each term in a document to avoid value overflow. The computation complexity is O(n) where n is the length of stems for a document. Since I search in "title" field, so the length will not be very large. The intuition for this feature is similar to spam property of a document, where a small value may reveal high valuable and meaningful for this document.

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	0.4480	0.4520	0.4760	0.4800
P@20	0.4240	0.4260	0.4660	0.4640
P@30	0.4160	0.4147	0.4547	0.4427
MAP	0.2459	0.2519	0.2590	0.2597

From the experiment, we can see with more features added, the MAP score keeps increasing. Thus, we can get to the conclusion that using content-based features is better than using IR features alone, and considering document properties like spam score is also useful in improving performance. Also, the custom features are useful on improving results. This experiment result behave as I expected, since IR score, term-overlap score, document properties and my custom features are all meaningful features which can be used to represent the match between document and query.

Compared with baseline, the results outperform all BM25, Indri BOW and Indri SDM. In specific, using all features, the MAP value improve 11% compared with Indri SDM, which is the best performance model in previous assignments. So the learning to rank model is very useful with these features.

Besides, I also get some other observations. Firstly, using BM25 can get a better performance than using Indri model. This is useful because we can use BM25 for initial retrieval in learning to rank instead of Indri. Also, it may show when calculating features, BM25 score might be more important than Indri or overlap scores. Secondly, when using learning to rank model, adding authority metrics for documents can improve MAP most (from 0.2459 to 0.2519), compared with this, other features added can only improve MAP a little bit.

Considering effective of my custom features, I run experiment separately to analyze each feature. When I add feature 17 to base features, the MAP is 0.2594; while when I add feature 18 to base features, the MAP is 0.2603. Although both of the performance is better than base features, we can see feature 18 can get more improvements (from 0.2590 to 0.2603), and the final all features' result is approximately average of these two MAP values. Thus, we can conclude that feature 18 is more meaningful on reflecting the query document matches. I think the reason is that feature 17 which calculates phrases overlap is quite correlated with previous features, while feature 18 which calculates average DF score is a separate property of documents which has not been calculated in previous features.

Besides, I also analyze P@10, P@20 and P@30 scores. It shows that using learning to rank, these metrics are all perform better than three baseline models. However, when compared within learning to rank model, we can see using Base features get highest P@20 and P@30 score, which is a kind of variation. I think the reason is that my custom features are not as stable as given features, although it may re-rank documents to get higher P@10, from more top documents point of view, like 20 or 30, it sometimes deceases the retrieval results.

4 Experiment: Features

Experiment with four different combinations of features.

	All (Baseline)	**1,2,3,4,5,6,7, 8,9,10,17,18**	**1,3,4,5,6, 8,9,18**	**1,3,5,8,18**	**1,3,5,8**	**1,5,8**
P@10	0.4800	0.4920	0.4880	0.4960	0.4960	0.5040
P@20	0.4640	0.4720	0.4820	0.4780	0.4780	0.4860
P@30	0.4427	0.4480	0.4520	0.4600	0.4520	0.4507
MAP	0.2597	0.2665	0.2670	0.2654	0.2686	0.2655

I generate combinations of features with the following steps.

Firstly, I think scores of query and documents in url and inlink field are less important, because these fields contain less contents of documents and for most of queries, these fields are not as comprehensive as body or title field. So I remove these features to keep only document authority metrics, IR and term overlap score in body and title field, and my custom features. Since url and inlink fields have less contents, the computational complexity is similar with all feature set (from 86945ms to 75196ms on local machine). The result shows an improvement on MAP from 0.2597 to 0.2665.

Secondly, based on the previous step, I think BM25 and Indri scores already contain property of overlap between query and documents, so I remove term overlap features. Also, the feature URL depth can be represented by spam score or page rank features, so I remove URL depth feature. Furthermore, based on analysis in experiment 3, my custom feature phrase overlap is less meaningful and can be represented by BM25 or Indri scores, so I remove feature 17. As a result, I use a combination of spam score, wiki score, pageRank score, BM25 and Indri score in body and title field and average DF score for learning to rank model, results show a little improvement on MAP compared to previous step.

Thirdly, based on second step, due to BM25 can get a higher performance in baseline compared with Indri, I continue to remove two Indri scores. I also remove pageRank score since it is highly correlated with spam score while spam score is calculated with Lucene library which can be more accurate. As a result, I use a combination of spam score, wiki score, BM25 score in body and title field and average DF score for learning to rank model. Compared with all feature set, I significantly reduce feature set size and only compute BM25 score in body and title field, so the computational complexity decreased (from 86945ms to 43125ms on local machine). However, results show a little decreasing on MAP but since we reduce the time complexity significantly, I decide to keep this step.

Forth, based on third step, I remove my custom feature because it is correlated with spam score. As a result, I use a combination of 4 features: spam score, wiki score, BM25 score in body and title field, for learning to rank model. Time complexity decreases to about one-third of all feature set, and result shows an improvement on MAP and it is even higher than the second step, from 0.2670 to 0.2686.

Lastly, based on forth step, I remove wiki scores to further reduce size of feature set. I do this not because this feature is not important but because I want to experiment on further reducing feature set. Result shows a relatively high decreasing on MAP compared with forth step, from 0.2686 to 0.2655, while the time complexity remains similar. Consider the trade-off between retrieval accuracy and feature set size, I think it's not worthy to have a feature set less than 4 due to this worse performance.

In conclusion, from above experiments, I'm able to get better effectiveness from a smaller set of features with less time complexity. I think the reason is that most of the features are actually correlated with each other, like BM25 and term overlap scores. Also, some of the features are less useful than the others, like IR scores in inlink or url field. Thus, with a better representation of feature set which remove duplicates and less meaningful features, we can make SVM model learned more concise and accurate.

5 Analysis

When analyzing SVM models, the feature with larger absolute weight value is more useful and the feature with less absolute weight value is less useful.

Thus, after analyzing feature weight, the top features are ranked as
Feature5: BM25 in body, Feature1: spam score, Feaure6: Indri in body, Feature10: term overlaps in title;

And the bottom features are ranked as
Feature17: sequence overlap score, Feature9: Indri in title, Feature14: BM25 in inlink, Feature4: pageRank score;

From the above results, for the top features, we can see that the top two features are selected as the final feature set in experiment 4, which is consistent with my analysis. BM25 score in body field have a good representations of the match between query and documents; spam score which calculated by Lucene is a good authority metric for document.

Besides, Indri score in body field is also a good metric for query and document pairs, but since it is similar and correlated with BM25 score, so I didn't use it in the final feature set.

Results also show term overlap score in title field can reflect the topic of a document which may match the query terms, I didn't use term overlap score in experiment 4 because I think it is not as good representative as BM25 scores, while facts show it is still meaningful, so I may experiment this feature in future work.

From the above results, for the bottom features, we can see none of them are used in my final feature set, which is consistent with experiment results. For custom feature17, as I have analyzed in experiment 3, it is less meaningful, so the weight is quite small.

For feature 9, as shown in my experiment steps in experiment 4, after I remove this feature together with Feature6, witch has a very high weight, the MAP value remains similar as previous steps. This seems that Feature6 on good representation may be approximately equals to Feature9 on bad representation. Thus, a better solution in experiment 4 may be keeping Indri score in body but removing Indri score in title instead.

For feature 14, as stated above, IR score in inlink field may be meaningful for a specific query, but in general, it may be less meaningful for retrieval. For feature 4 pageRank, it is correlated with other authority metrics but is read from a file, which actually lacks much data, so it is not as representative as spam scores or wiki scores.