Your Name: Chih-Wei Chang

Your Andrew ID: cchang3

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.

   Yes. Yu-Heng Lei reminded me some pitfalls we might encounter when calculating the features, such as avoiding incorrect normalization. Also, I discussed with Zhong Teng on how to retrieve some statistics (DF / CTF) from the TermVector properly.

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

   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.

   Yes.

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

   Yes.

Your Name: Chih-Wei Chang

Your Andrew ID: cchang3

Homework 4

1 Experiment: Baselines

	BM25	Indri BOW	Indri SDM
P@10	0.2160	0.2040	0.2040
P@20	0.2480	0.2680	0.2380
P@30	0.2573	0.2653	0.2480
MAP	0.1334	0.1462	0.1338

For BM25, we set k1=1.2, b=0.75, and k3=0. For Indri BOW, we used default #AND operator with mu=2500 and lambda=0.4. As for Indri SDM, we set the weight for term features to 0.8, the weight for ordered features to 0.15, and the weight for unordered features to 0.05. This assignment was derived from previous homework's experiment.

2 Custom Features

Our first customized feature is the number of dots in the raw URL. This idea was inspired by "Learning to detect phishing emails", Ian Fette 2007. In this paper, the authors suggested that the number of dots in URL might be correlated to the chance that a document or a web page is a phishing one. Normally, we would like to reduce the score of a spam website. Thus, we chose our first feature to be the number of dots in that it is somehow an indicator of spam and it can also easily be computed.

Our second customized feature is the simplified TF-IDF score. TF-IDF is a powerful indicator that can be used to identify whether or not a term is important. Therefore, inspired by BM25, we sum over all the TF-IDF scores that occurred in both query and the document. TF-IDF score can be easily calculated, and we also ignore other tedious details by just using $-\log(df+1) + \log(tf+1)$ as our formula.

3 Experiment: Learning to Rank

	IR Fusion	Content-Based	Base	All
P@10	0.2240	0.1960	0.4040	0.4200
P@20	0.2480	0.2500	0.4020	0.4100
P@30	0.2613	0.2707	0.3733	0.3760
MAP	0.1225	0.1246	0.1868	0.1916

Interestingly, neither IR Fusion features nor Content-Based features resulted in a better retrieval performance after re-ranking. This might indicate that if the features were not informative enough, then the re-ranking might hurt the performance instead of boosting it. Such decreasing could be thought as the noise in the learning process. Once the noises are too strong, then the model would be affected and be biased easily. Thus, if we were using noisy or non-informative features to train a ranking model, then it is very likely that it would generate some nonsense ranking.

On the other hand, we can notice that the Base features, which includes the first five document-based features, improve the retrieval performance a lot. The reason behind the difference between Base and Content-Based might be that the first five features contain some information that cannot be captured by the base retrieval model (i.e., BM25 BOW). Therefore, once adding these five features in, the chance is high that the learning model might be able to catch the information provided by the additional features and eventually resulted in a better ranking.

After adding our customized features, the major gain occurred at the P@10. This might because that the additional features focus on subtle characteristics (e.g., the number of dots) and thus can help the classifier differentiate the nuance among the top documents. The P@20 and P@30 are also improved but only with a relatively small scale. Furthermore, we noticed that the MAP score also have a significant improvement. This might confirm the improvement of P@10 comes from the additional features instead of just some unstable fluctuations.

4 Experiment: Features

Experiment with four different combinations of features.

	All (Baseline)	$Comb_1$	$Comb_2$	$Comb_3$	$Comb_4$
P@10	0.4200	0.4320	0.4320	0.4240	0.4520
P@20	0.4100	0.3940	0.4040	0.3980	0.3960
P@30	0.3760	0.3747	0.3920	0.3827	0.3987
MAP	0.1916	0.1935	0.1938	0.1910	0.1970

Our first combination ignored features from 5 to 16. That is, we used only the first four document-based features and the two customized features. From previous experimental results, we conjectured that these document-based features might be the much more informative one and required less computational cost. Thus, we tried to examine the effect of these features when being used separately. The result confirmed our hypothesis and gave us a smaller but still efeective feature set.

Our second combination is based on the first combination, but we include BM25 features (5, 8, 11, 14) to examine whether BM25 scores can give improvement. The experimental results suggested that adding BM25 might have minor improvement. We therefore wondered whether adding Indri can boost or hurt the performance. Therefore, our third combination further includes the Indri scores. It turned out that adding Indri might actually hurt the retrieval performance, which agreed with our previous experimental results.

Finally, the fourth combination ignored features 11 to 16, which means we ignored all features related to URL and Inlink. We examined this feature combination because although URL and Inlink might serve as a useful information source when doing base ranking, they might contain a lot of noise (as the average length of URL and Inlink is short) and hurt the learning to rank. Therefore, we tried to ignore these features and check whether it can improve the performance. The result suggested that removing these noise can in fact boost the performance a lot. This also showed the importance of excluding noisy features when applying learning to rank.

5 Analysis

We examined the model produced by using all 18 features.

1:0.42685887 2:-0.063437857 3:0.73654836 4:-0.023823293 5:0.065725237 6:-0.02480972 7:0.03397084 8:0.23384495 9:0.06628938 10:0.20659555 11:0.12676576 12:0.10683324 13:0.0916304 14:-0.0084203 15:0.007356754 16:-0.03220184 17:-0.01245919 18:0.19816676

Since we are using a linear model, the weights in the decision vector can somewhat revealed the importance of each feature. We notice that the most powerful feature is the third feature, i.e., the Wikipedia score, which is not very surprise as Wikipedia can be considered a quite reliable source. Next useful feature is the first feature, the spam score, which also aligned with our intuition in that spam is truly a useful indicator of whether or not a document should be ranked top.

Other than these two powerful features, we can see that feature 8, 10, and 18 are also informative, which corresponded to our conclusion in previous experimental results. Also, we noticed that features 14 to 16 are pretty weak. This also corresponded to our hypothesis that Inlink features might not be that useful, and they can even work as noise when being added to the training data.