Your Name Mengyu Yang

Your Andrew ID mengyuy

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

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 Mengyu Yang

Your Andrew ID mengyuy

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	0.3840
P@20	0.4040	0.3420	0.3920
P@30	0.4013	0.3453	0.3907
MAP	0.2286	0.1949	0.2156

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

BM25: k_1 = 1.2, b = 0.75, k_3 = 0.0;

Indri: mu = 2500, lambda = 0.4;

SDM: weight=[0.7, 0.2, 0.1]

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.

Custom Feature 1:

The first custom feature I use is Tf*Idf. Tf is the term frequency in a document and idf is the inverse document frequency. Idf is calculated by Math.log((docNum + 1) /df).

The time complexity is O(n) where n is the length of the query. Since n is relatively small, it is not too computationally expensive to be practical.

The intuitions behind Tf*Idf is that if a document has a higher term frequency (that is, contains more query term) it should be more relevant. At the same time, considering some common frequent word such as “I” “you” “they” etc, Idf is added to penalize those common but little meaning terms. Note that if a word occurs in many document, df will be large, and Idf will be small, so that the score of Tf*Idf will be reduced. This makes sense because the term that occurs in many document is very likely to be the common words that have less meaning.

Custom Feature 2:

The second custom feature I use is document length normalized score: (Σ(log(tf)+1))/ (Σ(log(doclen)+1))

Here Tf is the term frequency in a document. Doclen is the length of the docuemtent. In implementation log(tf) = Math.min(log(tf), 0), log(doclen) = Math.min(log(doclen), 0) in order to avoid negative infinity generated by log(0).The time complexity is O(n) where n is the length of the query. Since n is relatively small, it is not too computationally expensive to be practical.

The intuition behind is that if a document has higher term frequency, it is more likely to be relevant. At the same time, considering that a longer document has more words, so it is more likely to have higher tf, document length is added to compensate for higher tf in long documents. Note that for a shorter document, log(doclen) will be small, so that the score of this feature will be increased. This makes sense because a short document that contains the term is more likely to be relevant compared with a very long document that contains the term.

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	0.4400	0.4800	0.5080
P@20	0.4260	0.4280	0.4640	0.4660
P@30	0.4160	0.4213	0.4467	0.04467
MAP	0.2493	0.2530	0.2582	0.2590

1. Discuss the trends that you observe;
   Answer:

The more features enabled, the better the search engine performance. More specifically, IR Fusion contains only BM25 and Intri feature (f_5, f_6, f_8, f_9, f_{11}, f_{12}, f_{14}, f_{15}), Content-Based contains more feature (f5-f16), Base includes all the features except the custom features (f1 – f16), and All contains all the 18 features. From the chart we can see, the more features included, the higher the MAP value, which indicates the better the search engine performance.

2. Whether the learned retrieval models behaved as you expected;
   Answer:
   Yes. The retrieval models behaved as I expected. Before doing the experiment, I guess that more features for training will generate better model so that the final result will be more precise. The experiment result proves my guess. Also, the increase of MAP from Base to All proves that my custom features work well.

3. How the learned retrieval models compare to the baseline methods;
   Answer:
   All the learned retrieval models give better performance compared with the baseline methods. From the two charts we can see that MAP of learned model is around 0.24-0.26, while MAP of baseline is around 0.19-0.23. Therefore, all the four learned retrieval models (with different feature set) give better performance compared with the baseline methods.

4. Any other observations that you may have.
   Answer:
   Adding features in the learn retrieval model gives more improvement in precision than the improvement in recall. From the chart, P@10 is increased from 0.4480 to 0.5080 (more than10%) while MAP is only increased from 0.2493 to 0.2590 (less than 5%). While both precision and recall is improved by adding more features, precision improves more compared with recall.

5. Also, discuss the effectiveness of your custom features. 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.
   Answer:
   The two custom features are effective. From the experiment result, P@10, P@20 and MAP are all improved after adding the two custom features. P@10 improves from 0.48 to 0.508, P@20 improves from 0.464 to 0.466 and MAP improves from 0.2582 to 0.259. The improvement is not much though, after all there are only two features added. This result matches my expectations. Before experiment, I guess that the performance will improve a little bit if my custom features make sense. But the performance should not improve a lot since there are only two custom feature. The experiment proves my guess and proves that my custom features do work. Besides that, the experiment result indicates that my custom features are more effective in improving precision than in improving recall, as the P@n value increase a larger amount compare with the MAP value.

4 Experiment: Features

Experiment with four different combinations of features.

	All (Baseline)	**$Comb_1$**	**$Comb_2$**	**$Comb_3$**	**$Comb_4$**
P@10	0.5080	0.4920	0.5240	0.4760	0.4920
P@20	0.4660	0.4520	0.4880	0.4700	0.4760
P@30	0.04467	0.4387	0.4440	0.4493	0.4507
MAP	0.2590	0.2461	0.2645	0.2588	0.2599

1. Describe each of your feature combinations including its computational complexity.
   Answer:
   All: all the 18 features. Time: 87s

Comb1: f1, f5 (feature with highest coefficient in learned model). Time: 52s
Comb2: f1, f5, f10, f18 (features with coefficient > 0.3 in learned model). Time: 53s
Comb3: f1, f3, f5, f6, f7, f8, f10, f11, f13, f18 (features with coefficient > 0.1 in learned model). Time: 56s
Comb4: f1, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, f16, f17, f18 (features with coefficient > 0 in learned model). Time: 59s

2. Explain the intuitions behind your choices.
   Answer:
   I check the SVM model files, if the feature has positive value, it has positive effect on the result, otherwise it has negative effect on the result. The larger the value, the more useful the feature. The intuition behind my choices is to use features with different level of usefulness. It is a tradeoff between the quality of features and the number of features. In the first comb I only use the two most useful features. In the following combination, I lower the threshold of the coefficient so that the number of feature selected is increased. In the Comb4, I include all the features that have positive effect. The all four combination exclude feature that has negative effect on performance, that is, only the useful features remain, so the performance should be improved.

3. 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?
   Answer:
   There is a tradeoff between the quality of features and the quantity of features. In my experiment, comb2 and comb4 are better than baseline while comb1 and comb3 do not beat the baseline. The best result is obtained by using the Top 4 features, following that the result with all the POSITIVE features also beat the baseline. This is easily understood because we exclude those negative features, so features left all have positive contribution to the final result. Comb1 and Comb3 are not better than baseline. The reason may be that the positive effect by excluding low value features is not greater than the negative effect by reducing features. This is a tradeoff between the quality of the features and the number of the features. Some combinations get a better balancing point while others not.

5 Analysis

1. Examine the model files produced by SVM^{rank}. Discuss which features appear to be more useful and which features appear to be less useful.

   Answer:

   The model files produced by SMV is as below:

1:0.5297963 2:-0.13010536 3:0.27541566 4:0.020721598 5:0.49502027 6:0.27545696 7:0.29237711 8:0.23503914 9:0.0045359083 10:0.34898251 11:0.11942149 12:0.081011459 13:0.23298842 14:0.0095685329 15:-0.043293521 16:0.089209042 17:0.085191846 18:0.36816534

   The feature with positive value has positive effect on the result while the feature with negative value has negative effect. The larger the value, the more useful the feature. Feature1,5,10,18 has the highest value so they are most useful. Other features that ha value greater than 0 are less useful.

2. Support your observations with evidence from your experiments.

   Answer:

   From my experiment, the feature set [1,5,10,18] gives the best performance which proves my statement that the larger the feature value, the more useful the features. The feature set [1,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18] also beats the baseline because it excludes the negative features.