Xiaoyu He

xiaoyuh1

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, I asked Mouwu Lin, how to handle the case when TermVector is empty and when overlap score is zero.

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

Xiaoyu He

xiaoyuh1

Homework 4

Instruction

1 Experiment: Baselines

	BM25	Indri BOW	Indri SDM
P@10	0.3200	0.2000	0.2840
P@20	0.2400	0.1940	0.2600
P@30	0.2173	0.1907	0.2320
MAP	0.1246	0.0865	0.1323

BM25:k_1=1.2, BM25:b=0.75, BM25:k_3=0, Indri:mu=2500, Indri:lambda=0.4

For sequential dependency model, I gave #and, #near, #window weight 0.3, 0.35, 0.3. An example query looks like:

102:#wand(0.3 #and(fickle creek farm) 0.35 #and(#near/1(creek farm) #near/1(fickle creek)) 0.35 #and(#window/8(creek farm) #window/8(fickle creek)))

2 Custom Features

Feature 17 is Ranked Boolean score for body field. It can be calculated as the total sum of query term frequency in a document. The computation effort of computing Ranked Boolean is the cheap as compute overlap rate.

Though this model looks simple, I think it as can work as a complement of BM25 and Indri model, because this feature is only term frequency, without any normalization on the document length. I choose body field because I body field is relatively more important than other field (based on observation on the SVM weight). If I can add more custom feature, I would also try other fields.

Feature 18 is a simple version of VSM Score for body field. The idea of Vector space model is compute the similarity between query and documents. VSM score can be calculated in 2 steps. First, compute the dot product of query BOW vector and document BOW vector. To do this calculation, we will need TermVector of the document, dot product can be computed as the total frequency. Second, divide the dot product by the L2 norm of query vector and document vector. Those are not computationally very expansive.

My intuition behind this is, high cosine similarity means higher scores. I choose body field because I

body field is relatively more important than other field (based on observation on the SVM weight). If I can add more custom feature, I would also try other fields.

	IR Fusion	Content-Based	Base	All
P@10	0.2920	0.3120	0.3080	0.3120
P@20	0.2280	0.2420	0.2420	0.2460
P@30	0.2107	0.2200	0.2267	0.2280
MAP	0.1050	0.1150	0.1149	0.1154

The general trend is, with more feature throw into the SVM, the performance gets better. Content based model performs nearly as good as baseline and much better than IR fusion. This shows that, the term overlap score is can improve the IR Fusion system that only use BM25 and Indri model. Content-based system has comparably higher precision and worse recall than Base system.

My custom featured improved the system accuracy in both precision and recall. It improved the P@10 by 1.3%, P@20 by 1.6%, P@30 by 0.5%, MAP by 0.4%. Generally, my system improved the precision. I think both custom features contributed to this increased in precision.

3 Experiment: Features

Experiment with four different combinations of features.

	All (Baseline)	$Comb_1$	$Comb_2$	$Comb_3$	$Comb_4$
P@10	0.3120	0.3120	0.3160	0.2920	0.2960
P@20	0.2460	0.2480	0.2420	0.2560	0.2560
P@30	0.2280	0.2227	0.2227	0.2280	0.2320
MAP	0.1154	0.1158	0.1161	0.1256	0.1260

Since this SVM only use a linear model. Features that have higher weights are more important than features that have lower weights. I examined the weights file and picked the combination based on features that have highest weight.

Comb1: use body field 5,6,7, title field 8,9,10, spam 1, wiki 3. Ignore 2,4,11,12,13,14,15,16,17,18

The Comb1 only selected those features that have highest weights. It performs nearly the same as the baseline of using all features. This means that we can actually shrink down our feature sets.

Comb2: use body field 5,6,7, title field 8,9,10, spam 1, wiki 3, custom feature 17,18 Ignore 2,4,11,12,13,14,15,16

The difference between comb2 and comb1 is that, comb2 added custom features. This slightly improved the performance in P@10, MAP.

Comb3: use body field 5,6,7, spam 1, wiki 3, Ignore 2,4,8,9,10,11,12,13,14,15,16, 17, 18

Compare to comb1, comb3 only used body field, deleted title field from Comb1. Here, I try to test how will the search result behaves if I only use the body field. The result is surprising, the overall performance of only using the body performs much better than the base line. Comb3 got lower P@10 , but better P@20, P@30, MAP. This shows that, only use body field decreased the precision improves the recall .

Comb4: use body field 5,6,7, 1,2,3,4, 17,18 Ignore 8,9,10,11,12,13,14,15,16, 17, 18

Test the effectiveness of feature2 and feature3. The result show that they doesn't make mush difference.

4 Analysis

As I have discussed above. Features that have higher weights are more important than features that have lower weights. First, I observed the weights of using all features, the sorted results are as follows:

```
No. 1,  7:0.557488      No. 2,  5:0.511318          No. 3, 10:0.399755
No. 4,  8:0.307353      No. 5,  6:0.248693          No. 6,  9:0.163924
No. 7,  1:0.154086      No. 8, 16:0.086584          No. 9, 13:0.064682
No.10, 14:0.045234      No.11,  2:0.030869          No.12, 11:0.018610
No.13, 17:-0.007577     No.14, 12:-0.011623         No.15, 15:-0.031923
No.16, 18:-0.041240     No.17,  4:-0.135837         No.18,  3:-0.147612
```

5,6,7,8,9,10 are top 6 most important features. They are BM25, Indri, overlap score of body field and title field. Feature 1 Spam score ranks 7th, has weight similar to Feature 3 Wiki score and Feature 4 is PageRank. Theses three have comparably high absolute weight compare to the rest of features excluding the top 6.

I can support my assumption by the Comb test in the previous section.

Comb1 only use the features that have high weights and get slightly better performance than baseline. The SVM weights of Comb1 is:

```
No. 1,  7:0.566592          No. 2,  5:0.517489      No. 3, 10:0.403720
No. 4,  8:0.308930          No. 5,  6:0.250702      No. 6,  9:0.159634
No. 7,  1:0.154441          No. 8,  3:-0.114801
```

I think the weights of title and body are highly correlated, so I did another experiment comb3 to further explore it.

Comb3 only uses 5 feature 1,3,5,6,7 and performs much better than the baseline. Comb3 got lower P@10, but better P@20, P@30, MAP. It decreased the precision but improves the recall. The SVM weights is:

```
No. 1,  5:0.657600          No. 2,  7:0.616864
No. 3,  6:0.315697          No. 4,  1:0.147069      No. 5, 3:-0.246440
```

I think title field and body field are high correlated, so delete one of them doesn't harm the system performance.