Your Name: liping xiong

Your Andrew ID: lipingx

Homework 5

Statement of Assurance

I certify that all of the material that I submit is original work that was done only by myself.

1 Experiment: Baselines

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

	BM25	Indri BOW	Indri SDM
P@10	0.4600	0.3280	0.4440
P@20	0.4160	0.3420	0.4260
P@30	0.4160	0.3573	0.4213
MAP	0.2417	0.1964	0.2205

Document the parameter settings that were used to obtain these results.
BM25:k_1=1.2
BM25:b= 0.75
BM25:k_3= 0
Indri:mu=2500
Indri:lambda=0.4
The weights for SDM is: 0.2, 0.9, 0.9
An example:
1: #WAND(0.2 #AND(obama family tree) 0.9 #AND(#NEAR/1(obama family) #NEAR/1(family tree)) 0.9 #AND(#WINDOW/8(obama family) #WINDOW/8(family tree)))

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.

I tried several features and following is the reason why I choose them:

Length of body: From model analysis, I found that the features related to the body of documents are given high weights, which indicates that body field is import. Since the body field related features among base features don't consider the length of body as a separated feature. When we get BM25 score and Indri score, we use the field length. But it's just used as a normalization of term frequency. Thus, it's worth trying. In addition, it's simple to compute.

And intuitively, the longer the document, the higher the probility that the document will cover the information need expressed by the query. Also long document usually means high quality.

Vector Space Similarity of query and title of document:

Vector Space Similarity of query and body of document:

These two features are both query-dependent. Intuitively, more similar the query and document, the more relevant the document will be. So it's very natural to choose such features. And I tried "title" and "body" field.

When calculating cosine similarity, I use this:

$$\frac{\sum d_i \cdot q_i}{\sqrt{\sum d_i^2}\cdot\sqrt{\sum q_i^2}} = \frac{\overset{\text{Query operator}}{\searrow}\sum_{t\in d\cap q}\left(\log(tf_{t,d})+1\right)\cdot\left(\left(\log(qtf_t)+1\right)\,\log\frac{N}{df_t}\right)}{\underbrace{\sqrt{\sum_{t\in d}\left(\log(tf_{t,d})+1\right)^2}}_{\text{Document vector length}}\cdot\underbrace{\sqrt{\sum_{t\in q}\left(\left(\log(qtf_t)+1\right)\,\log\frac{N}{df_t}\right)^2}}_{\text{Query vector length}}}$$

We can see from the equation that most costly part is the document vector length. However, this part is not related to query. Thus it can be precomputed before query comes and also it's not too computationally expensive.

TFIDF sum of document terms:

TFIDF is the classical method to measure the importance of terms. Intuitively, the more important terms a document has, the more important the document. Thus I think the TFIDF sum can be a way to measure the quality of the document. Also it's query-independent, so it can be precomputed before query comes.

Length of title: From model analysis, the f8: BM25 score for <q, dtitle> and f10: Term overlap score for <q, dtitle> both get high weights, which suggest the importance of the title. But these two features are all measure how similar the query and the title. So I want to try the length of the title. I don't expect it'll improve the accuracy. Actually I don't think it'll affect the accuracy, because it's a query-independent feature, and we cannot judge the importance of a document with the length of its title. But I saw some people using this feature, so I just want to try.

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 (TFIDF sum + bodylen)
P@10	0.4520	0.4680	0.4880	0.4960
P@20	0.4260	0.4420	0.4620	0.4680
P@30	0.4293	0.4173	0.4573	0.4507
MAP	0.2608	0.2628	0.2678	0.2700

Discuss the trends that you observe; whether the learned retrieval models behaved as you expected; how the learned retrieval models compare to the baseline methods; and any other observations that you may have.

All learned retrieval models tried in this experiment outperformed the baseline. And MAP and p@10,p@20,p@30 are improved at the same time, which shows that Letor can improve the overall MAP without harming p@10,p@20,p@30.

Though model analysis, I found that "Termoverlap" has high weights. "IR Fusion" doesn't contain "Termovelap" compared to "content base". I think that's why its accuracy is lower than "content base".

"content base" discards query-independent features (features that just reflect documents' characteristics but not the query) compared to "base". And thus lose some accuracy. From this comparison, we know that query-independent features also important.

In the "All" experiment I use **baseline + TFIDF sum+ bodylen:**

This is the best I get among the custom features combination I tried (See following table).

I also tried TFIDF average, but it's not as good as TFIDF sum. And the Vector Space Similarity between query and field also doesn't outperform TFIDF sum+ bodylen.

Also, discuss the effectiveness of your custom features. This should be a separate discussion, and it should be more insightful than "They improved P@10 by 5%". 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.

I tried bunch of features, the results are listed below:

	Base +TFIDF sum +bodylen	**Base +TFIDF avg +bodylen**	**Base +bodylen**	**Base +SVS(title _query) +bodylen**	**Base +SVS(body_query) +bodylen**
P@10	0.4960	0.4960	0.4960	0.4920	0.4920
P@20	0.4680	0.4600	0.4640	0.4580	0.4580
P@30	0.4507	0.4573	0.4533	0.4493	0.4480
MAP	0.2700	0.2695	0.2699	0.2706	0.2679

SVS(title _query): Vector Space Similarity of query and title of document.

SVS(body_query): Vector Space Similarity of query and body of document.

From experiments, some of them are useful, while others are not. Following is the details:

Length of body: It turns out that body length is a good feature. It improves the search accuracy a lot. Both MAP and p@10,p@20,p@30. This is what I expected when I choosing this feature.

SVS(title _query):

SVS(body_query):

The experiments show that the Vector Space Similarity between query and field doesn't improve accuracy significantly. However. SVS(title _query) do improve MAP in a reasonable amount. This is not what I expected. I think Vector Space Similarity of query and field is a good indicator of relevance and thus it'll improve accuracy a lot. I guess the reason why it failed may be the equation I used to compute similarity need to adjusted. As the instructor told us in the lecture, it's heuristic. There is no guidance about how to set term weights and no guidance about how to determine similarity.

In addition, I found that SVS(title _query) works almost as well as SVS(body _query). And it's less expensive to compute. So it's a good candidate feature.

Sum of the tfidf of all the terms in document body field: This feature doesn't contribute much to the accuracy. It gets the weight of 0.06905 in "**Base+tfidf+bodylen"** experiment, while bodylen gets 0.1282 weight. This surprised me at first. Just like the PageRankScore gets weight 0.01472. I thought Tfidf sum of the terms in the document is a good indicator of document quality, just like PageRankScore.

But, think it carefully, it does make sense. Good quality page cannot guarantee relevance to the query.

Length of title: It turns out that the length of title didn't contribute much to the search accuracy. As I mentioned before I don't think the title length alone can measure the importance of the document. This experiment verified my intuition.

4 Experiment: Features

Experiment with four different combinations of features.

	All (Baseline)	Comb_1	Comb_2	Comb_3	Comb_4
P@10	0.4880	0.4800	0.4880	0.4920	0.4760
P@20	0.4620	0.4560	0.4560	0.4540	0.4540
P@30	0.4573	0.4533	0.4573	0.4560	04507
MAP	0.2678	0.2681	0.2684	0.2683	0.2675

Describe each of your feature combinations, including 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 combinations are investigating interesting hypotheses about what delivers good search accuracy. 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?

Comb1: discard the scores of inlink field, i.e f14,f15,f16

From model analysis, I found that the scores for inlink field is close to zero. So I think these scores can be discarded with losing accuracy. And my experiment verified my guess. The accuracy of Comb1 and Baseline is almost the same. So the scores of inlink field are useless and can be removed.

Comb2: based on Comb1, discard f3: PageRankScore

Model analysis shows that PageRankScore gets weight 0.01472. So I guess removing it would not change accuracy very much. My experiment verified my guess. p@10,p@20 and p@30 is almost the same as Comb1. The p@10, MAP even increased a little bit.

Comb3: based on Comb2, discard f9: Indri score for <q, dtitle>.

Model analysis shows that f9 gets weight -0.015149449. And from the results of Comb1 and Comb2, I guess discarding it would not affect accuracy. The result of experiment shows that the accuracy is very similar to Comb2. p@10 increased a little bit. Others dropped a little bit.

Comb4: based on Comb3, discard f12: Indri score for <q, durl>

Model analysis shows that f12 gets weight 0.080724552. After removing this feature from Comb2, both MAP and p@k dropped. This make sense because the weight is 0.0807 which is larger than the weight of f14,f15,f16 ,f9 and f12. So removing f12 will have more affect on the accuracy.

After 4 experiments, I tried another experiment which removed f2 whose weight is -0.1270 which is the least one among the remaining features. The experiment shows that discarding f2 decreased the MAP and p@k in a meaningful amount. So I guess removing other higher weight features will also harm the accuracy. Thus, I think it's the best to use the features of comb3 which discarded f14,f15,f16 and f9.

5 Analysis

Examine the model files produced by SVM^{rank}. Discuss which features appear to be more useful and which features appear to be less useful. Support your observations with evidence from your experiments. Keep in mind that some of the features are highly correlated, which may affect the weights that were learned for those features.

Some of this discussion may overlap with your discussion of your experiments. However, in this section we are primarily interested in what information, if anything, you can get from the SVM^{rank} model files.

Features	**weight**
f_1: Spam score for d (read from index).	1:**0.49295467**
f_2: Url depth for d(number of '/' in the rawUrl field).	2:-0.1270732
f_3: FromWikipedia score for d (1 if the rawUrl contains "wikipedia.org", otherwise 0).	3:0.26722953
f_4: PageRank score for d (read from file).	4:0.014724474
f_5: BM25 score for <q, d_{body}>.	5:**0.54837853**
f_6: Indri score for <q, d_{body}>.	6:0.35297787
f_7: Term overlap score for <q, d_{body}>.	7:0.29415902
f_8: BM25 score for <q, d_{title}>.	8:0.23988146
f_9: Indri score for <q, d_{title}>.	9:-0.015149449
f_{10}: Term overlap score for <q, d_{title}>.	10:0.31611425
f_{11}: BM25 score for <q, d_{url}>.	11:0.12998751
f_{12}: Indri score for <q, d_{url}>.	12:0.080724552
f_{13}: Term overlap score for <q, d_{url}>.	13:0.24929196
f_{14}: BM25 score for <q, d_{inlink}>.	14:-0.0091250082
f_{15}: Indri score for <q, d_{inlink}>.	15:-0.049972497
f_{16}: Term overlap score for <q, d_{inlink}>.	16:0.06469097

From above table, we can see that spam score and BM25 score for <q, dbody> are the most important features. The second important features are FromWikipedia score, Indri score for <q, dbody>, Term overlap score for all fields expect for "inlink". BM25 score for <q, durl> and urlDepth are followed.

PageRankScore and the scores for "inlink" field are the least useful features.

Firstly, Let's discuss query-independent features, i.e f1,f2,f3,f4. Query-independent features measures only the characteristic of documents. Model analysis shows that three of these features are important except PageRank score.

Spam score gets almost the same weight of BM25 score. This shows that spam score is an important measure of document quality. Higher score indicates less spammy, which means good quality of document.

UrlDepth gets negative weight. It makes sense. One reason is that longer url increases the probility of spam. So longer the url, less important the document will be.

PageRankScore gets very low weight, which surprised me at first. The PageRank value indicates an importance of a particular page. But PageRank prefers old pages. Maybe that's one reason why it gets low weight.

Overall, the quality of documents is important, but good quality doesn't guarantee high relevance to the query.

Secondly, let's talk about query-dependent features, which measures some relationship between query and documents.

The scores for the "body" field are very useful. This can be seen from baseline model analysis and also from the custom features I choose. When choosing custom features, I tried body length and Vector Space Similarity between body and query. Both custom features improved the accuracy when added to the base features. This makes sense because the "body" field is definitely the most important part of most documents when computing the relevance of the document.

The scores for the "inlink" field is useless. And previous experiments also verified this.

Termoverlap scores are importantd, almost in every field, except "inlink", which is consistent with intuition. The overlap terms are a strong and direct indicator of relevance. The more terms are matched between field and query, the more relevant the field is to this query.

BM25 scores are higher than Indri scores, especially in "title" and "url" fields. The indri scores of "title" and "url" fields are almost zero. From experiment 1(the baseline), we can see that the Indri model has lower accuracy than BM25 when they both using default parameters. Since, we also used default parameters when calculate Indri scores, it makes sense to put lower weight to these Indri scores.