A Short Report For Selective Search

1 abstract

Selective search is a archtecture in the distrubuted information retrieval. The traditional search solution divides the collection into shards and process query against each shards to get top-K relevant documetns results(exhaustive search). The serach computational cost for this traditional solution is often prohibitively high. In the selective search, we examined the topic-based sharding solution that making each shard contains similarly documents. Then, it is possible to only search for a few number of shards to retrieve to the desirable documents .

There are two sharding approach we used in the research: k-means and size-bounded k-means. Besides of examing how selective serach can improve search efficiency, we also proposed two kinds of expansion terms on each orignal docuement from the whole clueweb09-b collection to see if they can improve the sharding quality. The first expansion is generated from deep learning model DocT5, and the second expansion is the arfifical terms generated from real query routing.

2 topic based sharding

2.1 K-mean

K-mean is a well studied and well exmained unsupervised clustering method. In each iteration round in K-mean, the documents will be clustered to the most similarly centroids, and the feature vectors of those clustered documents will be used to generate centroids for next clustering iteration. And this will eventually results in a very unbalanced shards distributions since a centorid generated from a larger number of documents will be more attractive to the to-be-clustered-documents in next iteration and will recrusively growing larger. Thus, it is reasonably to propose a balance sharding approach.

2.2 Size-bounded K-mean

Distributed search system perfer the equal size shards for a better load-balacning and low-variance in each query routing time. The sb-kmean can parition collection into nearly equal size shards. It has two more steps than k-mean: divide the 'large' shards into 'small' shards, and after parition for the whole collection, merge 'small' shards into target size shard.

3 ExpansionTerms

3.1 DocT5 expansion

The DocT5 expansion is generated from a deep learning transformer model called DocT5. For each docuements in the collection, the model will raise question towards the 'key word' in the documents. For example, for a docuement talk about cars, the attention-mechanism will decide which word is the most important keyword in the document and then raise question like 'what is car' or 'how to use a car' to this keyword. For each docuement, we collect three questinos and append them to the end of the docuement.

3.2 Artifical Terms expansion

Since selective search based on the idel that top-k retrieved docuements for a query should be stay in same shards. we can use the actual query routing results to improve the clustering quality. Firstly, we do exhaustive seach for each query to get top-1000 retrieved docuements. Then we find those top-1000 documents and append query-id as a artifical term to the docuements.

4 Evaluation Method

For both evaluation method, we use retrieved docuements for each query in the exhaustive search as the label data, we will check the overlap percentage of those label-data in the top-k shards we select.

4.1 upper bound evaluation

Assuming we have a perfect shard selection algorithm, we can just use label-data to select the top-k most matched shards, and then to check overlap percentage. For each query, we have top-1000 ranked label documents result from exhaustive search. Mapping those document ids back to the shards they contained, we can generate a score for each shards of the targeted query. Then we select shards according to the descending rank of $\frac{ScoreOfShard_i}{LengthOfShard_i}$

4.2 sample based evaluation

For the sample based evaluation, we used the ReDDe algothrim to exam shards quality in real routing case. Firstly, we gererate a central sampled index file (randomly generate 1 percetage sample file of the whole collection). And run the query against the CSI file to retrieve top-k result docuements. Similarly to the first evaluation approach, we can generate score for each shards by mapping those retrieved doc-id back to the shards contained them. Then we select shards according to the descending rank of $\frac{ScoreOfShard_i}{LengthOfShard_i}$

5 result

5.1

This is result table for overlap percentage of top10 label document

Query-length info	P@10 original	P@10 terms+trans	P@10 queryOnly	random
All (num = 1886)	0.695	0.695	0.695	0.05
lenght = 1(num =36)	0.674	0.667	0.68	0.05
lenght = 2(num =485)	0.665	0.667	0.667	0.05
lenght = 3(num =586)	0.737	0.734	0.738	0.05
lenght = 5(num =200)	0.687	0.687	0.683	0.05

Table 1: K-mean P@10 on Top-5 shards

5.2

This is result table for overlap percentage of top100 label document

Query-length info	P@100 original	P@100 terms+trans	P@100 queryOnly	random
All (num = 1886)	0.356	0.352	0.356	0.05
lenght = 1(num =36)	0.297	0.302	0.291	0.05
lenght = 2(num =485)	0.375	0.370	0.375	0.05
lenght = 3(num =586)	0.409	0.403	0.408	0.05
lenght = 5(num =200)	0.347	0.337	0.347	0.05

Table 2: K-mean P@100 on Top-5 shards

5.3 real-routing result

This is the real-routing result for the original dataset a vs. terms expansion + transformer expansion dataset

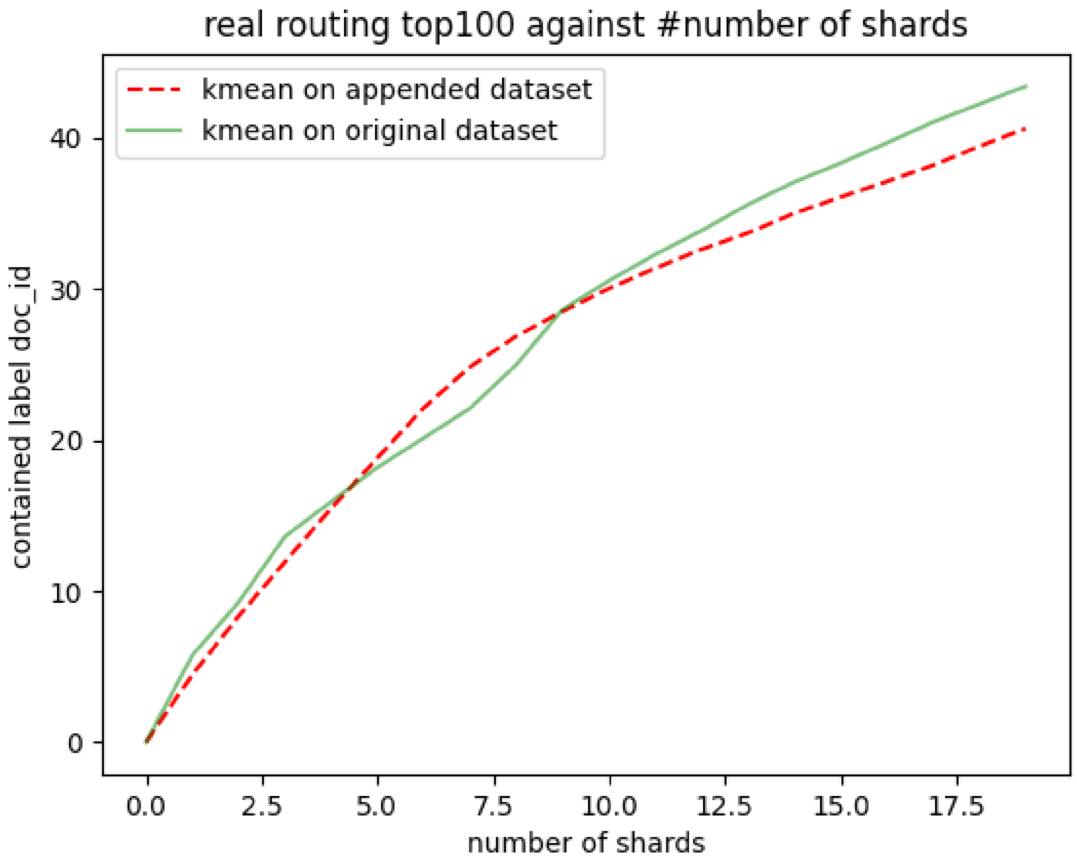

Figure 1: real-routing result

6 conclusion

For now, whole work is not done yet, the result for size-bound k-mean is still running and will be finished soon. For k-mean only, we can see top-5 shards(out of 100 shards) can help us retrieve almost 7 label docs out of of total 10, and 35.6 docs out of total 100. For artifical-term expansion approach, we did not see a significant improvement on the sharding quaility. I think we may need to change how we do NLP on the documents: One document usually are composed by hundreds of words, and aftifical terms only composed by less than 5 words. If we just append artifical terms on the original doc, it may only have trivail impact. Thus, we may need to find a way to do wieghting, making artifical terms weights more in the NLP phase. For the Transformer expansion, in the upper bound evaluation part, it make sharding quality a little bit worse since question-like terms will not help in clustering, instead of, it just appends more 'no senese' terms to the original documents that will make the feature extracting on the document more complex. And we still need to test whether its benefits in the query real-routing docuement retrieval can overcome its drawback for causing bad sharding quaility.