ICTNET at TREC 2017 Dynamic Domain Track

Weimin Zhang[1,2,3] Yaokang Hu[1,2,3] Rongqian Jia[1,2,3]
{zhangweimin, huyaokang, jiarongqian}@software.ict.ac.cn

Xianfa Wang[1,2,3] Le Zhang[1,2,3] Yue Feng[1,2,3] Sihao Yu[1,2,3]
{wangxianfa, zhangle, fengyue, yusihao}@software.ict.ac.cn

Yuanhai Xue[1,2] Xiaoming Yu[1,2] Yue Liu[1,2] Xueqi Cheng[1,2]
{xueyuanhai, yuxiaoming, liuyue, chengxueqing}@ict.ac.cn

[1] Institute of Computing Technology, CAS
[2] Key Laboratory of Web Data Science and Technology, CAS
[3] University of Chinese Academy of Sciences

1 Introduction

This track focuses on interactive search algorithms that adapt to the dynamic information needs of professional users as they explore in complex domains. [1] Unlike the common retrieval, there are two different aspects in TREC Dynamic Domain task, that we need to interact with users and retrieve in a specific data set. The JIG program acts as users' feedback, then we use the feedback to expand the query, the different subtopics of the user feedback information are also used for the result diversification algorithm. In addition, we also tried to integrate a variety of ranking algorithms and deal with some detail issues. Five solutions using the above algorithm were submitted. In those submissions, one combined Suggested Queries and JIG feedback [2], one didn't use stop strategy, the others all based on the same algorithm, but with different argumentstending to improve CT or ACT.

2 Data Preprocess and Rank Algorithms

2.1 Data Preprocess

For Ebola dataset, we extracted title and content using Boiler Pipe*. The New York Times dataset is structured, we analysed and read the main content and didn't take much effort on it.

2.2 Ranking Algorithms

We use Apache Solr† to do basic ranking. With the default arguments, the Dirichlet smooth language model works best among all the algorithms. Further, we have tried to integrate a variety of ranking algorithm which performed well in the Ebola data set. Because of the limitation of time, we didn't spend much time tuning parameters of this method on New York Times dataset. The preliminary results shows this method may not work well on New York times. One of the reason is Dirichlet smoothed Language Model already works well on New York Times Dataset while it works not so good on Ebola Dataset. The combination of ranking algorithms may make the results worse on the cases where the basic ranking algorithms works already very well. We only use this method on Ebola Dataset because we believe using different algorithms on different domain is necessary if it enhances performance greatly. The performance of the integrating algorithm on Ebola Dataset is shown in Table 1. And whether this method works well on other datasets needs further experiments.

*https://github.com/kohlschutter/boilerpipe

†http://lucene.apache.org/solr/

3 Methods

3.1 xQuAD

Automatically solution in TREC Dynamic Domain track should not read the ground data. What's more, we think interactive with JIG too many times, which makes it possible to know the full ground data, is also not good. Therefore, supervised learning is not suitable in this task.

xQuAD [3] Framework is an unsupervised algorithm which works well in previous TREC Dynamic Domain task. We also thought about HxQuAD [4] and HPM2 [4]. However, we found it hard to build hierarchical queries when the queries is long. The queries of the New York times dataset are too long to get suggested queries from search engines, not to mention building hierarchical queries for HxQuAD. For Ebola dataset, we use such methods to build subqueries or subtopics: use google to get suggested queries, use jig feedback to build subtopics, combine both of suggested queries and jig feedback [2]. As to the New York Times dataset, we only use jig feedback to build subtopics.

xQuAD Framework is described is follows:

$$d^* = \arg\max_{d \in R \setminus D} (1-\lambda) rel(q,d) + \lambda div(q,d,D) \text{ [2]} \tag{1}$$

where

$$div(q,d,D) = \sum_{s \in S_i} P(s|q) P(d|q,s) \prod_{d_j \in D} 1 - P(d_j|q,s) \text{ [2]} \tag{2}$$

where $rel(q,d)$ is the relevance score of d^*, $div(q,d,D)$ is the diversity score of d^*, $P(s|q)$ denotes the relative importance of aspect s given q, $P(d|q,s)$ denotes the coverage of document d with respect to this aspect, and the rightmost product denotes the novelty of any document covering this aspect, based upon how badly this aspect is covered by documents in D. [2] The document d^* the score of which is highest is selected.

The aspect s can be the suggested subqueries of q or JIG feedback. We also combine different kind of diversity score using different kind of aspect. This is shown as follows:

$$div_M(q,d,D) = \sum_{k \in K} \theta_k div_k(q,d,D) \text{ [2]} \tag{3}$$

where $div_k(q,d,D)$ is the diversity score build by suggested queries or jig feedback. The details about how to use jig feedback as subtopic/aspect is described in section 3.3.

3.2 PM2

In this experiment we used the query expansion methods to improve the relevance of the query, but only improving the relevance often makes the results too single around the minority several subtopics, but not well covered more subtopics. This requires us to improve the diversity of experimental results. We use the PM2 method [5] to improve the diversity of our results. We use several articles obtained by search engines as corpus. We use LDA model to obtain a certain number of subtopics. Each time we combine the feedback results to select the subtopic that should be compensated as the main subtopic. For each document, calculate the relevance of the main subtopic and the relevance of other subtopic, obtain the weight of each document, and then select the document with the largest weight as the submission. In the selection of main subtopic, we use the proportional representation to select the main subtopic, every time when we need to allocate a seat (select a subtopic), We assign the seats to the Party which currently has the lowest proportion of seats (select the subtopic which currently has the lowest proportion). This allows the distribution results to be as close as possible to the overall rate. We define quotient qt_i as the compensation value for each subtopic. Each time the subtopic with the largest compensation value is chosen as the main subtopic, we calculate the compensation value of each subtopic by the following formula:

$$qt_i = \frac{w_i}{2s_i + 1} \text{ [5]} \tag{4}$$

Where s_i is the current proportion of $topic_i$ in the selected document, and w_i is the proportion of $topic_i$ in the whole corpus in possession. Then we need to calculate the weight of each document. We select the document with maximum weight as the final document. Let the selected document be as relevant as possible to the main subtopics. we calculate the weights of each document by the following formula:

$$d^* = \arg\max_{d \in R \setminus D} \Phi(d^*, D, t) \text{ [5]} \tag{5}$$

where

$$\Phi(d^*, D, t) = \lambda \cdot qt_{i*} \cdot P(d|t_{i*}) + (1 - \lambda) \cdot \sum qt_i \cdot P(d|t_i) \text{ [5]} \tag{6}$$

3.3 Result Diversification and Query Expansion

The Rocchio algorithm [6] is used in Dynamic Domain task [7]. We tried to use Rocchio algorithm and xQuAD at the same time. However, we found it hard to define irrelevance documents. If the 'on_topic' of a document is 0, the document is still possible relevant to the topic [8]. What's more, we may only get documents that are scored 1 and 2. If we mark the documents, the score of which is lower than 2, as irrelevant documents, we may lose the relevance information returned by jig. Since we cannot find a good way to define irrelevant documents which is needed by the Rocchio algorithm, we only use relevant documents to reformulate query.

In the first iteration, we use initial query to retrieve documents. And when jig marks some documents as 'on topic', we expand queries using the feedback of jig. We do this process as follows:

- If all the documents we gave to jig are not on topic, we use initial query until we get the documents, the on topic value of which is 1.
- Concat all the passage text returned by jig if the passage text is of same score and on the same subtopic. This paragraph is called an aspect or a subtopic.
- If the number of aspects bigger than the limit(we set this limit as 2), use tf-idf to get top t words from each aspects. The weight value of words are the value of their tf-idf. Expand all these words to initial query.

Notice the process above is dynamic, which means in the next iteration, we will expand query based on the initial query not the expanded query in the last iteration.

We reformulate the xQuAD Framework as follows:

$$d^* = \arg\max_{d \in R \setminus D} (1 - \lambda) rel(q_m, d) + \lambda \sum_{s \in S_i} P(s_m|q) P(d|q, s_m) \prod_{d_j \in D} (1 - P(d_j|q, s_m)) \tag{7}$$

where query q and aspect s are changing every iteration. In the mth iteration, we uses this equation to rerank documents.

3.4 Detailed Strategies

We observed the dynamic process and make some rules to achieve better performance.

- We tried several ways to calculate $P(d|q)$ which is necessary in xQuAD. If using language model(which is suggested in the paper where xQuAD is proposed [3]), we found the relevance part and the diversification part may have a big gap. For example, on some occasion the relevance part in xQuAD may be 10^{-6} and the diversification part may be 10^{-12}, while on some other occasion the gap of the two part may be very small, which makes it hard to tune the parameters. Also, it is impossible for parameters to take effect. What's worse, the value is too small to ensure precision. If use LDA [9], it is easy to find the subtopic found by LDA is not precise and it takes much time to train LDA. We finally use cosine similarity between query and document and use tf-idf value to represent the vector. Moreover, everytime we only use the top 30 documents that retrieved by solr which have not be returned to jig as the input of the xQuAD algorithm .This strategy makes xQuAD works better. We have tried using top 20,30,40,50 documents and 30 is the best parameter.
- There are many duplicate documents, especially in Ebola Dataset. So we record all the documents we gave to jig. If the jig mark these documents not on topic, we will filter documents that are similar to such documents. This supplement our algorithms.

Table 1: Comparison between LM Dirichlet and Ensemble Algorithms

	ct@1	ct@2	ct@10	act@1	act@2	act@10
LM Dirichlet	0.2923569	0.1942791	0.0517162	0.2207451	0.1988821	0.1070681
Ensemble Algorithms	**0.3467666**	**0.2043171**	**0.0524392**	**0.2501640**	**0.2209722**	**0.1138523**

- The xQuAD algorithm use documents set D to record the documents we gave to jig. On the first iteration, if less than 3 document are marked as on topic by jig, we will not add these on-topic documents into D. In fact, relevance and diversification are not totally unrelated. Since we use query expansion algorithms, it is necessary to get more documents that are related to the topic, which will increase diversification in turn. Therefore, we use this strategy to get more relevance documents so that we can achieve more diversification.

3.5 Stop Strategy

After consulting some methods [2], we tried some stop strategy. After comparing the experimental results, we selected the second methods in this paper: when the number of our method submit unrelated documents reach 10, we will stop the search algorithm.

4 Experiments

4.1 Ensemble Ranking Algorithms

We only tried this method on Ebola Dataset. We notice the jig of 2016 is different from 2017. This Experiment is on Ebola Dataset using jig 2017. LM Dirichlet means Dirichlet smoothed language model, while ensemble algorithms means the combination of LM Dirichlet, BM25, Classic Similarity, IB Similarity, LM Jelinek Mercer Similarity, the weight of which is 3,1,1,1,1. The results is shown as Table 1.

4.2 PM2 with Query Expansion

We improve the diversity of the results by the PM2 method, but the relevance of result has been reduced, which requires us to have a better search results. In this experiment, we use the Rocchio algorithm [6] to improve the query quality. Everytime we will modify the query according to the results. After the submission, if the sum of scores of subtopic is not greater than 8, we will select 10 keywords from the document as the words added to the query, selected 3 key words from passage_text added to the query. In the selection of keywords, we firstly filter out stopwords, and then use the TF-IDF value to judge whether a word is a keyword. The result is shown in Table 2.

4.3 Run Results of our Submissions

We submit 5 runs on this task. The results are shown in the Table 2. We started use jig feedback when we get more than 2 subtopics from jig. And once we get even only one subtopic from jig, we expand our query.

- Baseline1 The result of solr. All the data are sorted by Dirichlet smoothed language model.
- Baseline2 The result of solr. For the New York Times dataset, we use Dirichlet smoothed language model. For ebola dataset, we use Ensemble Ranking Algorithms proposed in previous section.
- ictnet_div_qe Based on baseline2, we use xQuAD and query expansion algorithms from the second iteration. The value of λ is 0.8. The number of words expanded from each subtopic is 5. We use only jig feedback as subtopics.
- ictnet_emulti Based on baseline2, we use xQuAD and query expansion algorithms from the second iteration. For ebola dataset, we use both suggested queries and jig feedbacks as subtopics. The score of these two parts are added to sort the documents. The weight of both parts are 0.1. The value of λ is 0.8 for jig feedback based xQuAD. The value of λ is 0.6 for suggested queries based xQuAD. The number of words expanded from each subtopic is 5.

Table 2: Results of our runs

	Submit	act@1	act@2	act@10	ct@1	ct@2	ct@10
Baseline1	N	0.3242499	0.2785729	0.1386702	0.4092635	0.2469839	0.0625547
Baseline2	N	0.3328052	0.2846921	0.1404897	0.4244252	0.2496322	0.0626838
PM2 QE	N	0.3239961	0.2807215	0.1869917	0.4105926	0.2536386	0.1129254
ictnet_div_qe	Y	0.3328052	0.2934678	0.2091378	0.4244252	0.2690583	0.1280947
ictnet_emulti	Y	0.3328052	0.2936302	0.2107725	0.4244252	**0.2697782**	**0.1290707**
ictnet_fom_itr1	Y	**0.3339564**	0.2936881	**0.2120427**	**0.4419974**	0.2654532	0.1277942
ictnet_params1_s	Y	0.3328052	**0.2937297**	0.2077913	0.4244252	0.2687960	0.1271756
ictnet_params2_ns	Y	0.3328052	**0.2937297**	0.1452637	0.4244252	0.2687960	0.0651831

- ictnet_fom_itr1 Based on ictnet_div_qe, we start to use our algorithm from the first iteration. The parameters are just as described in ictnet_div_qe.
- ictnet_params1_s Based on ictnet_div_qe, we only change the number of words expanded to 6.
- ictnet_params2_ns Based on ictnet_params1_s, we didn't use stop strategy.

5 Conclusion and Acknowledgements

TREC Dynamic Domain task addresses both relevance and diversification. It is a dynamic process. It is necessary to change algorithms according to different domains, queries and even iterations. We have tried some strategy in the task this year, such as ensemble ranking algorithms, using both query expansion and result diversification algorithms and so on. However, we still have a long way to go.

We would like to thank all organizers and assessors of TREC and NIST. This work is sponsored by the National Basic Research Program of China (the 973 program) under grant numbers 2014CB340401 and 2014CB340406.

6 References

[1] http://trec.nist.gov/pubs/call2017.html

[2] Moraes F, Santos R L T, Ziviani N. UFMG at the TREC 2016 Dynamic Domain track[C]//TREC. 2016.

[3] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In Proc. of WWW, pages 881890, 2010.

[4] S. Hu, Z. Dou, X. Wang, T. Sakai, and J.-R. Wen. Search result diversification based on hierarchical intents. In Proc. of CIKM, pages 6372, 2015.

[5] V. Dang and W. B. Croft. Diversity by proportionality: An election-based approach to search result diversification. In SIGIR, pages 6574, 2012.

[6] J. Rocchio. Relevance Feedback in Information Retrieval. Prentice Hall, Englewood, Cliffs, New Jersey, 1971.

[7] Albahem A, Spina D, Cavedon L, et al. RMIT @ TREC 2016 Dynamic Domain: Exploiting Passage Representation for Retrieval and Relevance Feedback[C]// TREC. 2016.

[8] http://trec-dd.org/guideline.html

[9] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3:993-1022.