IR-IITBHU at TREC 2016 Open Search Track: Retrieving documents using Divergence From Randomness model in Terrier

Mitodru Niyogi[1] and Sukomal Pal[2]

[1]Department of Information Technology, Government College of Engineering & Ceramic Technology, Kolkata
[2]Department of Computer Science & Engineering, Indian Institute of Technology(BHU), Varanasi

Abstract

In our participation at TREC 2016 Open Search Track which focuses on ad-hoc scientific literature search, we used Terrier, a modular and a scalable Information Retrieval framework as a tool to rank documents. The organizers provided live data as documents, queries and user interactions from real search engine that were available through Living Lab API framework. The data was then converted into TREC format to be used in Terrier. We used Divergence from Randomness (DFR) model, specifically, the Inverse expected document frequency model for randomness, the ratio of two Bernoulli's processes for first normalisation, and normalisation 2 for term frequency normalization with natural logarithm, *i.e.*, In_expC2 model to rank the available documents in response to a set of queries. Altogether we submit 391 runs for sites CiteSeerX and SSOAR to the Open Search Track via the Living Lab API framework. We received an 'outcome' of 0.72 for test queries and 0.62 for train queries of site CiteSeerX at the end of Round 3 Test Period where, the "outcome" is computed as: #wins / (#wins + #losses). A 'win' occurs when the participant achieves more clicks on his documents than those of the site and 'loss' otherwise. Complete relevance judgments is awaited at the moment. We look forward to getting the users' feedback and work further with them.

1 Introduction

Open Search is a new evaluation paradigm for IR. The experimentation platform is an existing search engine. The task which focuses on academic search at the moment provides participants an opportunity to use a live search system through displaying search results developed by the participants and getting feedback on them by real users of the search system. During our participation

Table 1: Indexing

Collection Size	980MB
#indexed docs	22309
#size of vocabulary	23093
#tokens	191928
#pointers	185557

at TREC 2016, we submitted official runs for CiteSeerX and SSOAR to TREC Open Search Track. We used the Inverse expected document frequency model for randomness as ranking model in Terrier as an evaluation system to rank documents related to queries.

The remainder of the paper is organized as follows. Section 2 contains a description of our methodology: indexing and our ranking model. Section 3, presents our experiment for the Open Search Track. In Section 4, we present the results obtained by our approach after the end of Round 1, 2 and 3 respectively. Section 5 discusses about future scope and we close with some concluding remarks in Section 6.

2 Methodology

2.1 Indexing

In order to index the .JSON test collection, we employ a local inverted file approach [7]. We split the collection in a number of disjoint sets of documents and index them separately. While indexing, we remove standard stopwords and apply the first step of Porter's stemming algorithm [4]. For each disjoint set of documents, we create the following data structures.

- direct file that contains all the terms of each document. The direct file is used for the query expansion models.
- an inverted file that contains all the document identifiers, in which a term appears.
- a lexicon that contains the vocabulary of the indexed documents.
- a document index that contains information about the indexed documents.

The average number of tokens per document is given in Table 1.

2.2 Ranking Model

DFR models[1] are implemented by instantiating three components of the framework: selecting a basic randomness model, applying the first normalisation and normalising the term frequencies. The DFR models are based on this simple idea: "The more the divergence of the within-document term-frequency from its

frequency within the collection, the more the information carried by the word t in the document d". In other words, the term-weight is inversely related to the probability of term-frequency within the document d obtained by a model M of randomness:

$$weight(t \mid d) \quad \propto \quad -\log Prob_M(t \varepsilon d \mid Collection) \tag{1}$$

where the subscript M stands for the type of model of randomness employed to compute the probability.

Terrier offers a number of DFR-based models for document weighting. The relevance score of a document d for a particular query Q is given by:

$$score(d, Q) = \sum_{t \varepsilon Q} (qtfn \cdot w(t, d)) \tag{2}$$

where $w(t, d)$ is the weight of the document d for a query term t and $qtfn$ is the normalised frequency of term t in the query. It is given by $qtf/qtfmax$, where qtf is the original frequency of term t in the query, and $qtfmax$ is the maximum qtf of all the composing terms of the query.

We have used the model In_expc2 (DFR): Inverse expected document frequency model for randomness, the ratio of two Bernoulli's processes for first normalisation, and Normalisation 2[6] for term frequency normalisation with natural logarithm [1]. Model $I(n_e)C2$

$$w(t, d) = \frac{F + 1}{N_t \cdot (tfn_e + 1)} (tfn_e \cdot \log_2 \frac{N + 1}{n_e + 0.5}) \tag{3}$$

The

- tf is the within-document frequency of term t in document d
- F is the term frequency of term t in the whole collection
- N is the number of documents in the collection
- N_t is the document frequency of term t
- n_e is given by $N \cdot (1 - (1 - \frac{N_t}{N})^F)$
- λ is given by $\frac{F}{N}$ and F $\ll$ N.

The relation f is given by the Stirling formula:

$$f(n, m) = (m + 0.5) \cdot \log_2 \frac{n}{m} + (n - m) \cdot \log_2 n \tag{4}$$

tfn is the normalised term frequency. It is given by the normalisation 2[6]:

$$tfn = tf \cdot \log_2 (1 + c \cdot \frac{avg\_l}{l}) \tag{5}$$

where c is a parameter, l is the document length, which corresponds to the number of tokens in a document, and avg_l is the average document length in the collection.
tfne is the normalised term frequency, which is given by the modified version of the normalisation 2:

$$tfn_e = tf \cdot \log_e(1 + c \cdot \frac{avg\_l}{l}) \tag{6}$$

The only free parameter of the DFR framework is the term frequency normalization c from Equations (5) and (6). The tuning of such a parameter is a crucial issue in IR, because it has an important impact on the retrieval performance [8, 2]. A classical tuning method is the pivoted normalization [10], which fits the document length distribution to the length distribution of relevant documents.However, since the document length distribution is collection-dependent, the pivoted normalization suffers from the collection-dependency problem. Indeed, the optimal parameter settings of diverse document collections are different [8]. In our experiments with Terrier, the parameter c is automatically tuned, according to a method proposed by He and Ounis [9]. This method assumes a constant optimal normalization effect with respect to the document length distribution of the collection, and it assigns the parameter value such that it gives this constant. Thus, it is a collection-independent approach.

3 Experiments

The downloaded document data which is in JSON format was split into multiple files containing multiple documents using bash command split; keeping in mind the split files are in correct formatting of JSON. After splitting, each file was converted into XML format which IR framework Terrier supports i.e., TREC Web Collection by running a python script which was developed using dicttoxml module in Python. The converted format documents file was indexed in Terrier 4.1 system by configuring terrier's terrier.properties file. After indexing, ranking models In_expC2 was chosen for retrieving. Meanwhile the queries, i.e. topics in TREC format had already been downloaded from API endpoint. The result file (.res) file gets generated which is now split according to query id using a bash script and now it's been converted back into JSON format using bash script where Ubuntu's awk tool is used for formatting JSON. Finally, the rankings of each query is uploaded at api.trec-open-search.org/api/participant/run/key/qid using a bash script for upload where curl command is used.

3.1 Data Conversion

Conversion of downloaded data of TREC Open Search (.JSON format) into TREC Style format,i.e,XML for evaluating it in Terrier using python script and terrier has been setup for indexing and retrieval.

Table 2: Test Data Results for CiteSeerX

	OUT	WIN	LOSS	TIES	IMPRESSIONS
Round 1	0.0	0	0	1	1
Round 2	0.54	7	6	2	15
Round 3	0.7142	5	2	2	9

Table 3: Train Data Results for CiteSeerX

OUT	WIN	LOSS	TIES	IMPRESSIONS
0.62	18	11	4	33

Sample: Converted format of Citeseerx documents for Terrier accepted TREC format

```
<?xmlversion="1.0"encoding="UTF-8"?> <root> <docs> <content>
<text> </text> </content> <creation_time> </creation_time>
<docid> </docid> <site_id> </site_id> <title> </title>
</docs> <docs> </root>
```

Sample: Converted format of SSOAR documents for Terrier accepted TREC format

```
<?xml version="1.0" encoding="UTF-8" ?> <root> <docs> <content> <abstract> </abstract>
<author> </author> <available> </available> <description> </description> <identifier>
</identifier> <issued> </issued> <language> </language> <publisher> </publisher>
<subject> </subject> <type> </type> </content> <creation_time> </creation_time>
<docid> </docid> <site_id> </site_id> <title > </title> </docs> </root>
```

4 Results

The results for sites CiteSeerX and SSOAR at the end of Round 2 are given in Table 2 for test queries and Table 3, 4 for train queries respectively. The "outcome" is computed as: #wins / (#wins + #losses). Where a win is defined as the participant having more clicks on documents assigned to it by Team Draft Interleaving than clicks on documents assigned to the site.

5 Discussion

There are some more models implemented within Terrier such as Bose Einstein model for randomness, language models etc. We could not experiment with

Table 4: Train Data Results for SSOAR

Round 1	OUT	WIN	LOSS	TIES	IMPRESSIONS
	0.0	0	2	1824	1826

these models.

6 Conclusion

We chose the ranking model In_expC2 for submitting runs to the task. We received an impression count (the number of times the results have been shown to users) of 1969 as of January 30, 2017 positioning us at a top spot in terms of outcomes among sixteen teams with an outcome of 0.72 at the end of Round 3. The outcome reflects that our ranking of documents got 72 % more clicks by users after being interleaved with its production system. As the relevance judgments for all queries are not available to us as of now, we have not been to evaluate our runs which we have generated for each individual queries. But the work so far has been exciting and we look forward to continue with the work.

References

[1] Amati, G.,Van Rijsbergen, C.J.: *Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness*, ACM - Transactions on Information Systems, 20, 357-389, (2002).

[2] G. Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness.* PhD thesis, Department of Computing Science, University of Glasgow, 2003.

[3] *Manning, C.D., Prabhakar R., Schtze, H.: Introduction to Information Retrieval*, Cambridge University Press, New York, USA (2009).

[4] *Porter, M.F., (1980), An Algorithm for Suffix Stripping*, 14(3) :130-137.

[5] *Vassilis Plachouras, Ben He, and Iadh Ounis. University of Glasgow at TREC2004: Experiments in Web, Robust and Terabyte tracks with Terrier* In TREC, Vol. Special Publication 500-261 (2004).

[6] Terrier 4.1 Documentation,
`http://terrier.org/docs/v4.1/` [last accessed on 22-July-2016].

[7] *B. A. Ribeiro-Neto and R. A. Barbosa. Query performance for tightly coupled distributed digital libraries.* In Proceedings of the third ACM conference on Digital libraries, pages 182.190. ACM Press, 1998.

[8] *A. Chowdhury, M. C. McCabe, D. Grossman, and O. Frieder. Document normalization revisited.* IIn Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 381.382. ACM Press, 2002.

[9] *B. He and I. Ounis. A study of parameter tuning for term frequency normalization.* In Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM), pages 10.16. ACM Press, 2003.

[10] *A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization.* In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 21.29. ACM Press, 1996.