LTRo: Learning to Route Queries in Clustered P2P IR

Alkhawaldeh, R. S., Padmanabhan, D., Jose, J. M., & Yuan, F. (2017). LTRo: Learning to Route Queries in Clustered P2P IR. In LTRo: Learning to Route Queries in Clustered P2P IR (Lecture Notes in Computer Science). Springer. DOI: 10.1007/978-3-319-56608-5_42

Published in:
LTRo: Learning to Route Queries in Clustered P2P IR

Document Version:
Peer reviewed version

LTRo: Learning to Route Queries in Clustered P2P IR

Rami S. Alkhawaldeh[1] Deepak P[2], Joemon M. Jose[3], and Fajie Yuan[3]

[1] Department of Computer Information Systems, University of Jordan-Aqaba, 77110, Jordan,
r.alkhawaldeh85@gmail.com
[2] School of Computer Science, Queen's University, University Road, Belfast BT7 1NN, UK,
deepakp7@gmail.com
[3] School of Computing Science, University of Glasgow University Avenue, G12 8QQ, UK,
Joemon.Jose@glasgow.ac.uk
f.yuan.1@research.gla.ac.uk

Abstract. Query Routing is a critical step in P2P Information Retrieval. In this paper, we consider learning to rank approaches for query routing in the clustered P2P IR architecture. Our formulation, LTRo, scores resources based on the number of relevant documents for each training query, and uses that information to build a model that would then rank promising peers for a new query. Our empirical analysis over a variety of P2P IR testbeds illustrate the superiority of our method against the state-of-the-art methods for query routing.

1 Introduction

Query routing (*aka* resource selection) refers to the task of selecting a subset of resources to send each query to, in de-centralized search systems such as P2P IR and federated search systems. The considerations for P2P IR systems are typically different from those in federated search systems due to the asymmetry of document distribution across peers; for example, there could be peers with an order of magnitude more documents than others. Thus, methods which perform very well in federated search systems (e.g. CORI[1], logistic regression [2]) do not necessarily work that well for P2P IR. However, supervised approaches that make use of training data (i.e., past queries and information about peers deemed relevant for them) have not been explored much for the P2P IR query routing task.

In this paper, we consider the task of supervised query routing within the semi-structured cluster-based P2P IR architecture [3]. This architecture has been subject of recent interest [4,5], largely due to the presence of intra-peer content coherence at the query routing layer. For the first time, we consider learning-to-rank methods for supervised query routing within clustered P2P IR. Learning to Rank (LtR) techniques are supervised learning methods that can exploit training data in the form of a ranked list of objects [6]. Additionally, LtR approaches can also work with peer-specific [7], and peer-pairwise [8] relevance information. As an example, for our task of query routing, LtR approaches can be trained on a list of peers ordered according to their relevance to each query in the training set. In particular, we consider the following questions:

- Are LtR approaches applicable for the query routing problem in clustered P2P IR?
- How do LtR approaches compare against state-of-the-art models for query routing in clustered P2P IR?

2 Related Work

We now briefly survey related work on supervised resource selection. Among the first approaches for supervised resource selection was the method due to Arguello et al. [2] targeted towards the task of federated search; they propose usage of logistic regression to rank resources against queries. For every query-resource pair, the training feature vector is a concatenation of:

- *Query-dependent Corpus features:* A set of documents are sampled from each resource, and their relevance to the query is estimated using methods such as CORI[1] and ReDDE.top[9].
- *Query features:* These features encode query information such as the category of the query, and web documents that are deemed to be relevant to the query.

The relevance judgement is generated by firing training queries against the full dataset, i.e., the dataset across all resources. A resource is considered relevant if has more than a threshold (τ) number of documents among the top T documents from the full result. Hong et al. [10] extend this work for cases where a full dataset search is infeasible. Instead of the full dataset result, they build the 'full result' using just the top-T documents from each resource. In order to offset for inaccuracy in such approximation, they model and exploit similarities between resources in the query routing task. Thus, a resource which is not highly ranked against the queries using features may still be chosen by virtue of high resource-level similarity to other resources that are relevant to the query.

Cetintas et al. [11] propose a query routing approach that assesses resource relevance using the following formulation:

$$Rel(r_j|q) \propto \sum_{q' \in training} Rel(r_j|q') \times Sim(q', q)$$

Here, the relevance judgements for training queries are determined using the information as to whether the resource was selected for the query (using any resource selection method), whereas the similarity between queries are estimated using the correlation of their respective result sets.

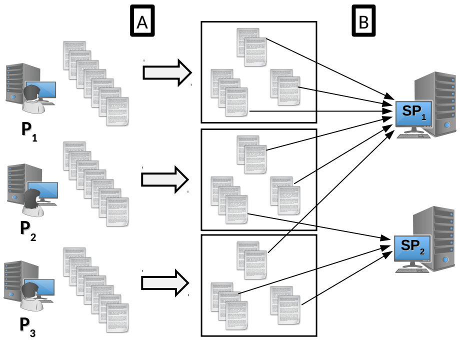

Fig. 1: Clustered P2P IR Architecture

3 Clustered P2P IR Architecture

Figure 1 illustrates the construction of the clustered P2P IR architecture [3,4,5], our target architecture in this paper. Each of the peers maintain a subset of documents, as shown by the different P_is in the left side of the figure. The subset of documents within each peer are clustered independently (into k clusters, $k = 3$ in the figure), represented as Step **A**; we will call this as *intra-peer clustering*. Phase **B** clusters these intra-peer clusters, across peers, into a specified number (two, in the figure) of clusters. Each such cluster is managed by a super-peer (SP_i). Due to the clustering, not every super-peer necessarily would have representation from each peer; in our example, SP_2 does not have representation from P_1. Every query to the P2P IR system is sent to *each* of the super-peers, which would then employ the query routing approach to route the query to a subset of peers judged to be relevant to the query.

4 LTRo: Learning to Route

We now describe our LtR-based query routing approach, codenamed LTRo. General classification-based approaches such as those from [2] and [10] work with training data in the form of $[V_{q,r}, L_{q,r}]$ pairs. $V_{q,r}$ is a vector for the combination of query q and resource r, whereas $L_{q,r} \in \{-1, +1\}$ denotes whether the resource r is relevant for the query q or not. This is used to learn a mathematical model that can predict whether a resource is relevant to a query, thus enabling query routing:

$$\mathcal{F} : V_{q,r} \rightarrow \{+1, -1\}$$

LtR Training Data Formats: In addition to training data with binary relevance judgements as above, learning to rank approaches can exploit pair-wise relevance judgements in the form of triplets like $[V_{q,r_1}, V_{q,r_2}, L_{q,r_1,r_2}]$ where L_{q,r_1,r_2} indicates whether resource r_1 is more relevant to q than r_2. Yet another format is list-wise training data which is typically of the form $[V_{q,r_1}, V_{q,r_2}, \ldots, V_{q,r_m}, L_{q,r_1,\ldots,r_m}]$ with $L_{q,r_1,\ldots,r_m}$ denoting whether the chosen ordering of resources (i.e., starting with r_1) corresponds to the ordering in the non-increasing order of relevance to the query q. Once we have training data that has numeric values quantifying relevance information for each query-resource pair, it is straightforward to use the scores to generate data in any of the three forms above. We now describe the construction of the feature vector $V_{q,r}$ and that of the associated numeric score in LTRo.

Feature Vector Construction: Our feature vector, i.e., $V_{q,r}$, is constructed using a variety of features that indicate the relatedness between the training query and the corpus within each resource. As in earlier methods for supervised query routing, we sample documents from each resource, and use that to estimate the relatedness of the resource to each training query. The features we use are the concatenation of features from the following sources:

- Classical resource selection methods such as CORI[1] and CVV[12].
- Document Retrieval methods from various families, viz., (i) vector space models (TF and TF-IDF[13]), (ii) query relevance models (Language Modeling[14]), and (iii) divergence from randomness models (DFI0[15], BB2[16]). The usage of document retrieval methods is inspired by recent work [4] indicating their effectiveness for resource selection in the clustered P2P IR architecture.

Labelling: The labels associated with training data are critical to supervised learning. We now outline our method to associate numeric scores to each training vector $V_{q,r}$. Such numeric labels would then be converted, in a straightforward manner, to labels for appropriate choices of training data formats (pointwise, pairwise, or list-wise, as outlined earlier). We use the sampling-based approach for labeled data creation used in [10], whereby only a fixed sample of results (we set sample size to be 10) are obtained from each resource per training query. For every query-resource pair, we set the numeric score to the number of relevant retrieved documents in the sampled subset for the query.
LtR Models in LTRo: Having defined the construction of training vectors and associated scores, it is then simple to deploy any LtR algorithm for the task. We experimented with all the LtR models available in the RankLib[4] package, and did not find any perceivable difference in performance across them. Thus, we consistently employ the latest list-wise LtR technique from the RankLib library, i.e., co-ordinate ascent [17], in LTRo.
Testing phase: For every new query (i.e., query from the test set), the LTRo model ranks the resources in the order of relevance to the query. We select the top-$k\%$ of all resources to route the query to. k is a parameter for the approach that may be varied; we experiment with values of k from $\{5\%, 10\%, 20\%, 30\%, 40\%, 50\%\}$ and report average of the evaluation measures across these values of k.

5 Experimental Study

We experimentally analyze LTRo against baseline approaches on several standard testbeds for P2P IR. We start by describing the setup and the baselines, and then go on to analyzing the experimental results.
Setup: We use several standard P2P IR testbeds from [18] in our evaluation. Each of these are based on the WT10g dataset[5], and model a variety of real-world data distributions with varying number of peers and varying skew of documents between peers. The characteristics of the various testbeds are summarized in Table 1. TREC 2000 and 2001 web track topics for the WT10g corpus are used as queries along with their ground truth relevance judgements. We selected 10,000 training query from 1.6 million known-item queries[6] leading to a choice of 18.82% single-term queries, 47% two-term queries, 19.7% three-term queries and the remaining 13.32% comprising four terms or more. The COMBMNZ [19] merging algorithm is used to combine the results from peers. We use the TREC 2001 query topics from 451-550 (these were excluded from training) as our test queries, thus replicating the setup from [4,5].
Baseline Methods: We have not come across supervised query routing methods that are specifically targeted to the clustered P2P IR architecture. Thus, we compare against the regression method from [2] (denoted as LR) as well as against a simple multi-layer perceptron based learner (MLP). In order to enable quantify the enhanced performance of the supervised approaches, we also report results from Taily [20], a recent unsupervised query routing method.
Experimental Results: Table 2 summarizes the comparative retrieval effectiveness of LTRo against the baseline approaches on each of the six testbeds, in terms of Precision (@top-1000), Recall (@1000), Precision@10 and MAP. The LTRo method is

[4] https://sourceforge.net/p/lemur/wiki/RankLib/
[5] http://ir.dcs.gla.ac.uk/test_collections/wt10g.html
[6] http://boston.lti.cs.cmu.edu/callan/Data/P2P

Table 1: Test-beds general properties

Characteristics	ASISWOR	ASISWR	DLWOR	DLWR	UWOR	UWR
# Peers	11680	11680	1500	1500	11680	11680
# Docs	1692096	1788248	1692096	1740385	1692096	1788896
Avg. Docs in Peer	144.87	153.1	1128.54	1160.26	144.87	153.16

Table 2: LTRo Retrieval effectiveness: two-paired statistically significant bootstrap t-test; p $\leq$ 0.01 are denoted as • compared to Taily method.

DL*	**DLWOR test-bed**				**DLWR test-bed**			
Method	**Precision**	**Recall**	**P@10**	**MAP**	**Precision**	**Recall**	**P@10**	**MAP**
Taily	0.02815	0.52157	0.16050	0.08944	0.02519	0.48203	0.02367	0.02786
LR	0.04017	**0.57933**	0.22633	0.13322	**0.03687**	**0.54805**	0.05650	0.06389
MLP	**0.04028**	0.57889	0.22683	0.13135	0.03679	0.54706	0.05650	0.06389
LTRo	0.03972$^\bullet$	0.56756$^\bullet$	**0.23015**$^\bullet$	**0.13470**$^\bullet$	0.03668$^\bullet$	0.53493$^\bullet$	**0.05767**$^\bullet$	**0.06506**$^\bullet$
LTRo-LR	**(-3.77%)**	**(-16.32%)**	**(+7.48%)**	**(+3.46%)**	**(-1.57%)**	**(-15.22%)**	**(+3.71%)**	**(+3.47%)**
LTRo-MLP	**(-4.8%)**	**(-18.77%)**	**(+6.04%)**	**(+7.55%)**	**(-0.94%)**	**(-14.23%)**	**(+3.71%)**	**(+3.48%)**
ASIS*	**ASISWOR test-bed**				**ASISWR test-bed**			
Taily	0.02581	0.46064	0.15833	0.07046	0.01934	0.37833	0.01733	0.02042
LR	0.04356	0.54821	0.24500	0.12134	**0.03965**	0.52124	**0.06400**	0.06027
MLP	0.04354	0.54891	0.24400	0.12099	0.03954	0.52066	**0.06400**	0.06021
LTRo	**0.04412**$^\bullet$	**0.55917**$^\bullet$	**0.24600**$^\bullet$	**0.12454**$^\bullet$	0.03959$^\bullet$	**0.52462**$^\bullet$	0.06317$^\bullet$	**0.06194**$^\bullet$
LTRo-LR	**(+2.76%)**	**(+6.73%)**	**(+1.07%)**	**(+5.34%)**	**(-0.28%)**	**(+2.46%)**	**(-1.72%)**	**(+4.18%)**
LTRo-MLP	**(+2.85%)**	**(+6.27%)**	**(+2.16%)**	**(+5.98%)**	**(+0.25%)**	**(+2.89%)**	**(-1.72%)**	**(+4.33%)**
U*	**UWOR test-bed**				**UWR test-bed**			
Taily	0.02797	0.49474	0.18783	0.10229	0.02374	0.43882	0.01400	0.02451
LR	0.08842	0.73835	0.47133	0.32898	0.08488	0.71121	0.12317	0.14347
MLP	0.08842	0.73908	0.47233	0.32908	0.08485	0.71135	0.12317	0.14342
LTRo	**0.08856**$^\bullet$	**0.74407**$^\bullet$	**0.47400**$^\bullet$	**0.33283**$^\bullet$	**0.08498**$^\bullet$	**0.71152**$^\bullet$	**0.12783**$^\bullet$	**0.14663**$^\bullet$
LTRo-LR	**(+0.24%)**	**(+2.33%)**	**(+0.98%)**	**(+1.7%)**	**(+0.17%)**	**(+0.12%)**	**(+4.31%)**	**(+2.71%)**
LTRo-MLP	**(+0.23%)**	**(+2.03%)**	**(+0.61%)**	**(+1.66%)**	**(+0.22%)**	**(+0.07%)**	**(+4.31%)**	**(+2.76%)**

seen to outperform others in 75% of the metrics (18/24), and closely trails the leading method in the other cases (except for the DLWOR testbed, where the difference is more perceivable). The improvements achieved over the baseline approaches have also been indicated in the table. The results indicate that LTRo should be the method of choice for supervised query routing. This shows the effectiveness of going beyond binary relevance labeling and consequent usage of learning-to-rank approaches for the query routing problem in P2P IR.

6 Conclusions and Future work

In this paper, we considered the applicability of learning to rank methods for query routing within the clustered P2P IR architecture. Accordingly, we modeled the query routing problem within the learning to rank framework, and empirically evaluated it against state-of-the-art supervised and unsupervised algorithms for query routing. Our empirical analysis illustrates the superiority of our LtR approach, codenamed LTRo, in a large majority of scenarios, thus indicating that LTRo should be the method of choice for supervised query routing for clustered P2P IR.

References

1. Callan, J.P., Lu, Z., Croft, W.B.: Searching distributed collections with inference networks. In: Proc. of SIGIR, New York, NY, USA, ACM (1995) 21–28
2. Arguello, J., Callan, J., Diaz, F.: Classification-based resource selection. In: Proceedings of CIKM '09, New York, NY, USA, ACM (2009) 1277–1286
3. Klampanos, I.A., Jose, J.M.: An evaluation of a cluster-based architecture for peer-to-peer information retrieval. In: DEXA. (2007) 380–391
4. Alkhawaldeh, R., Jose, J., Padmanabhan, D.: Evaluating document retrieval methods for resource selection in clustered p2p ir. CIKM (2016)
5. Alkhawaldeh, R., Jose, J., Padmanabhan, D.: Clustering-based query routing in cooperative semi-structured peer to peer networks. ICTAI (2016)
6. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: From pairwise approach to listwise approach. Technical Report MSR-TR-2007-40, Microsoft Research (April 2007)
7. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. Annals of Statistics **29** (2000) 1189–1232
8. Wu, Q., Burges, C.J., Svore, K., Gao, J.: Ranking, boosting, and model adaptation. Technical Report MSR-TR-2008-109, Microsoft Research (October 2008)
9. Arguello, J., Diaz, F., Callan, J., Crespo, J.F.: Sources of evidence for vertical selection. In: Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '09, New York, NY, USA, ACM (2009) 315–322
10. Hong, D., Si, L., Bracke, P., Witt, M., Juchcinski, T.: A joint probabilistic classification model for resource selection. In: Proceedings of SIGIR '10, New York, NY, USA, ACM (2010) 98–105
11. Cetintas, S., Si, L., Yuan, H.: Learning from past queries for resource selection. In: Proceedings of CIKM '09, New York, NY, USA, ACM (2009) 1867–1870
12. Yuwono, B., Lee, D.L.: Server ranking for distributed text retrieval systems on the internet. In: DASFAA, World Scientific Press (1997) 41–50
13. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
14. Hiemstra, D.: Using language models for information retrieval. Univ. Twente (2001)
15. Dinçer, B.T., Kocabas, I., Karaoglan, B.: Irra at trec 2009: Index term weighting based on divergence from independence model. In: TREC. (2009)
16. Amati, G., Van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems (TOIS) **20**(4) (2002) 357–389
17. Metzler, D., Bruce Croft, W.: Linear feature-based models for information retrieval. Inf. Retr. **10**(3) (June 2007) 257–274
18. Klampanos, I.A., Poznański, V., Jose, J.M., Dickman, P., Road, E.H.: A suite of testbeds for the realistic evaluation of peer-to-peer information retrieval systems. In: Proc. of ECIR. (2005) 38–51
19. Shaw, J.A., Fox, E.A.: Combination of Multiple Searches. In: Text REtrieval Conference. (1994) 243–252
20. Aly, R., Hiemstra, D., Demeester, T.: Taily: Shard selection using the tail of score distributions. In: Proceedings of SIGIR '13, New York, NY, USA, ACM (2013) 673–682