Online Learning to Rank for Cross-Language Information Retrieval

Razieh Rahimi
Department of Computer Science
Georgetown University
Washington D.C.
razieh.rahimi@georgetown.edu

Azadeh Shakery
School of Electrical and Computer Engineering
College of Engineering, University of Tehran
School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
shakery@ut.ac.ir

ABSTRACT

Online learning to rank for information retrieval has shown great promise in optimization of Web search results based on user interactions. However, online learning to rank has been used only in the monolingual setting where queries and documents are in the same language. In this work, we present the first empirical study of optimizing a model for Cross-Language Information Retrieval (CLIR) based on implicit feedback inferred from user interactions. We show that ranking models for CLIR with acceptable performance can be learned in an online setting, although ranking features are noisy because of the language mismatch.

CCS CONCEPTS

•Information systems →Users and interactive retrieval; Learning to rank; Multilingual and cross-lingual retrieval;

KEYWORDS

Online learning; Learning to rank; Cross-language information retrieval

1 INTRODUCTION

Leveraging user interactions to optimize Web search results has attracted considerable attention in recent years. This is because offline evaluation of retrieval models based on the Cranfield paradigm does not necessarily generalize to actual users and other time periods [18], while online optimization based on user interactions allows learning of personalized ranking function [3].

An online learning to rank algorithm for information retrieval integrates user feedback in optimizing the parameters of a ranking function. In contrast, the common approach in learning to rank algorithms is to optimize parameters of a retrieval model based on manually annotated training data, thus these algorithms perform offline learning. Online systems for IR examine a new setting of retrieval parameters at each iteration and update the parameters based on user feedback on the provided ranking. The goal of an online system for IR is to maximize cumulative performance of result lists presented to the user in the learning process, referred to as *online performance*. This objective function is to ensure that users do not experience low-quality results during maximizing the performance of a ranking function. On the other hand, *final performance* refers to the performance of the learned ranking function on test data in both online and offline cases.

Although user interactions are exploited to optimize systems for monolingual information retrieval in several studies (e.g., [9, 11, 16, 20, 21, 26]), to the best of our knowledge, these interactions have not been specifically used to optimize models for CLIR. In this paper, we investigate online optimization of systems for CLIR.

Implicit feedback inferred from user interactions with a retrieval system is inherently noisy, which makes online optimization of ranking functions challenging [9]. In addition to the noisy nature of such feedback, features of a cross-lingual ranking function are noisy. This happens because some statistics in ranking features such as *term frequency* and *document frequency* of a term, cannot be computed directly in CLIR, and are estimated using translation models. The noisier nature of systems for CLIR makes learning from user interactions more challenging. In this work, we study the suitability of online learning of a ranking function for CLIR.

In this paper, we address three research questions: **(1)** How does the *final* performance of an *online* learning to rank algorithm compare to that of an *offline* learning to rank algorithm for CLIR? This comparison demonstrates how a learned ranking model for CLIR based on user interactions compare to that based on explicit manual judgments. **(2)** How does the *final* performance of an online learning to rank algorithm for *CLIR* compare to that for *monolingual IR*? **(3)** How does the *online* performance of an online learning to rank algorithm for *CLIR* compare to that for *monolingual IR*? The second and third comparisons specifically reveal the impact of noisier ranking features in CLIR on the performance of an online learning to rank algorithm.

We demonstrate that, although the cross-lingual environment is noisier than the monolingual one, online learning to rank algorithms can be successfully adopted to learn personalized ranking functions.

2 RELATED WORK

The purpose of our study is to examine online learning to rank for CLIR, which is not yet explored to the best of our knowledge. However, there are several studies related to our work, which form two groups: (1) work on using (offline) learning to rank algorithms for CLIR, and (2) work on online learning to rank for monolingual information retrieval.

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan

DOI: http://dx.doi.org/10.1145/3077136.3080710

Learning to rank algorithms have been widely used for ad hoc IR and many IR applications [13]. For multilingual information retrieval, as a sub-task of IR, there are a number of methods based on learning to rank algorithms [7, 23, 24]. These methods mainly focus on learning to merge result lists retrieved separately for each language in multilingual IR. However, this step is not required in CLIR where all documents are in the same language, but different from the query language. Gao et al. [6] learn a ranking function based on bilingual features for monolingual IR. In another line, Azarbonyad et al. [1] define cross-lingual features to learn a ranking function for CLIR, however parameters are learned in an offline setting.

Online learning to rank for IR. Learning to rank approaches for IR in an online setting try to optimize search results based on user interactions. The challenging point is that retrieval systems cannot estimate the utility value for a provided ranking from user interactions. Ordinal feedback for some document pairs [12] or lists of documents [17] can only be inferred. The latter feedback can be obtained using interleaved comparison methods including balanced interleave [17], and multileave comparison [21]. In particular, interleaving methods facilitate comparison of utility values for result lists of two or more rankers. Using such methods, online optimization of ranking functions is modeled as a *dueling bandits problem* in [26]. Using interleaved comparison methods allows listwise learning of ranking functions in an online setting. In another line, Hofmann et al. [11] investigate online learning to rank based on pairwise document preferences. They finally improve the performance of online learning to rank by balancing exploration and exploitation for both listwise and pairwise learning. In another study, Hofmann et al. [9] investigate how the learning speed can be increased by reusing historical interaction data of users.

Online learning of parameter values is also adopted to optimize the base BM25 ranker using user interactions [20]. In addition to optimization of parameters with continuous values, interleaved comparison methods are used for online evaluation of a finite set of rankers, which is formulated as k-armed dueling bandit problems in [25]. However, all these methods are examined for monolingual IR.

Algorithm 1 DBGD algorithm [26]

Require: γ, δ, w_1
1: **for** query q_t ($t = 1 \dots T$) **do**
2: Sample unit vector u_t uniformly.
3: $w'_t \leftarrow w_t + \delta u_t$
4: Compare w_t and w'_t
5: **if** w'_t wins **then**
6: $w_{t+1} \leftarrow w_t + \gamma u_t$
7: **else**
8: $w_{t+1} \leftarrow w_t$
9: **return** w_{T+1}

3 ONLINE LEARNING TO RANK FOR IR

In this section, we describe the problem of online optimization of a ranking function. Online learning to rank for information retrieval is modeled as a reinforcement learning problem in which the retrieval system repeatedly interacts with the user to learn an approximately optimal ranking function by maximizing the cumulative reward. The cumulative reward is calculated over an infinite horizon of time steps using [11]:

$$C = \sum_{t=1}^{\infty} \gamma^{t-1} r_t, \tag{1}$$

where r_t is the reward received at time t, and is weighted by the discount rate $\gamma \in [0, 1)$ to place more emphasis on immediate rewards. Reward r_t is computed using a retrieval performance measure such as average precision or Normalized Discounted Cumulative Gain (NDCG).

Online optimization of retrieval functions parameterized by a weight vector $\boldsymbol{w}$, is formulated as dueling bandits problem for continuous parameter space [26]. The Dueling Bandit Gradient Descent (DBGD) algorithm to learn the weight vector in this problem is shown in Algorithm 1. The algorithm works as follows. At each timestep t, the system receives query q_t and provides a ranked list to the user. This ranked list is generated by interleaving methods, which compare two retrieval functions by providing the user with a combination of their respective rankings. One ranking is generated by using the current best estimate of the weight vector, maintained in w_t. The other exploratory ranking is produced by perturbation of w_t along a random direction u_t. The user interacts with the result list, and click information is used to determine the winner. If the exploratory list wins the comparison, the current best weight vector is updated by moving along u_t. This process repeats continuously.

Herein, the goal is online learning of a weight vector for linear combination of ranking features in CLIR. In particular, the ranking model f for CLIR is a linear function of the form

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\top}\boldsymbol{x}, \tag{2}$$

where $\boldsymbol{x}$ denotes the ranking features for CLIR and $\boldsymbol{w}$ is a weight vector. The goal is online learning of $\boldsymbol{w}$ based on users' implicit feedback using Algorithm 1. Line 4 of this Algorithm compares two ranking models. For this step, we employ probabilistic interleave method [10].

4 EXPERIMENTS AND RESULTS

Table 1: Dataset properties.

Year	Data collection	Document language	Query language	Experiment name
2002	Los Angeles Times 1994	English	French	2002:En-Fr
			Italian	2002:En-It
			Spanish	2002:En-Es
	Le Monde 1994 French SDA 94	French	English	2002:Fr-En
	La Stampa 1994 Italian SDA 94	Italian	English	2002:It-En
2003	Los Angeles Times 1994 Glasgow Herald 1995	English	French	2003:En-Fr
			Spanish	2003:En-Es

Datasets. Evaluations are carried out against test collections from ad-hoc cross-language track in CLEF-2002 and CLEF-2003 campaigns. We use English, French, and Italian collections with query sets in multiple languages for the experiments reported here, which represent different language pairs and different translation directions. Test collections and their languages as well as query languages are shown in Table 1. In addition, the query set in the

Table 2: Learning features for CLIR.

Feature	Description	Category
1	$\sum_{q_i \in q} \log(1 + \text{CLTF}(q_i, d))$	Q-D
2	$\sum_{q_i \in q} \log(1 + \frac{\text{CLTF}(q_i, d)}{\lvert d \rvert})$	Q-D
3	$\sum_{q_i \in q} \log(\text{CLIDF}(q_i))$	Q
4	$\sum_{q_i \in q} \log(1 + \frac{\lvert C \rvert}{\text{CLTF}(q_i, C)})$	Q
5	$\sum_{q_i \in q} \log(1 + \frac{\text{CLTF}(q_i, d)}{\lvert d \rvert} \cdot \text{CLIDF}(q_i))$	Q-D
6	$\sum_{q_i \in q} \log(1 + \frac{\text{CLTF}(q_i, d)}{\lvert d \rvert} \cdot \frac{\lvert C \rvert}{\text{CLTF}(q_i, C)})$	Q-D
7	PSQ score	Q-D
8	LMIR with DIR smoothing	Q-D
9	LMIR with JM smoothing	Q-D
10	LMIR with ABS smoothing	Q-D
11	$\lvert d \rvert$	D

language of each test collection is used to provide monolingual baseline for CLIR performance. We index the TEXT and TITLE fields of documents in test collections for retrieval.

Preprocessing. Diacritic characters are mapped to the corresponding unmarked characters. Stopwords are removed. Next, we use Snowball stemmers for all languages.

Translation Models. We build a word-to-word translation model for each language pair using the *Europarl Corpus* [22]. Statistical translation models (IBM model 1) are obtained using the GIZA++ toolkit. Before word alignment, the aforementioned preprocessing steps are done on both sides of each parallel corpus. Obtained translation probabilities are then linearly normalized by selecting the top 3 translations for each word.

Learning features. For feature extraction, we first sample some documents for each query. We use the BM25 and *Probabilistic Structured Query* (PSQ) [5] models to rank all documents with respect to each query in monolingual and cross-lingual settings, respectively. After ranking, the top 1,000 documents for each query are selected for feature extraction. We extract 11 features for each query-document pair, which are shown in Table 2 for the cross-lingual setting similar to [1]. In this setting, frequency of query term q_i in document d as well as document frequency of q_i are estimated using translation models as follows [5]:

$$\text{cltf}(q_i, d) = \sum_{w \in v_d} p(q_i, w) \times \text{tf}(w, d), \tag{3}$$

$$\text{cldf}(q_i) = \sum_{w \in v_d} p(q_i, w) \times \text{df}(w), \tag{4}$$

where v_d represents the vocabulary set of the document language, and $p(q_i, w)$ shows the translation probability of word w to q_i in the respective translation model. Inverse document frequency of a term is computed as $\text{clidf}(w) = \log \frac{N+1}{\text{cldf}(w)}$, where N is the total number of documents in the document collection. The PSQ model for CLIR uses the estimates in Eqs. 3 and 4 in the BM25 model to rank documents. Parameters of the BM25 model are set as $k_1 = 1.2$, $k_3 = 7$, and $b = 0.75$ [5, 15]. Features 8, 9, and 10 are calculated by integration of translation models in the query language model [14], and smoothing parameters of these features are set as $\mu = 2,000$, $\lambda = 0.1$, and $\delta = 0.7$, respectively [15]. For monolingual query-document pairs, we extract the same set of features, where cltf and clidf are respectively replaced by tf and idf, and feature 7 is the BM25 score. Finally, we perform query-based normalization for each feature.

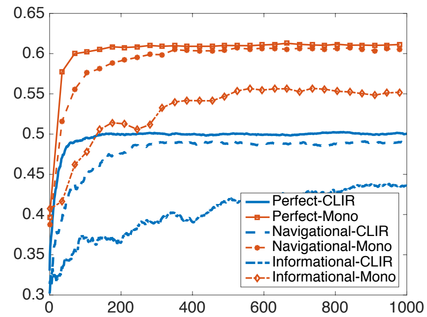

Figure 1: Final performance (NDCG@10) over iterations for 2002:En-Fr dataset and all click models.

Simulation of user clicks. User clicks are generated based on the dependent click model [8] which generalizes the cascade model to multiple clicks. We instantiate this click model based on click and stop probabilities similar to the instantiations used in [11]. These instantiations simulate three levels of increasing noise in user's feedback. The *perfect* click model provides reliable feedback, and is used to obtain an upper bound on the performance. The other two models, *navigational* and *informational*, are realistic user models. To the best of our knowledge, there is no study to specifically model user clicks in cross-lingual search sessions. However, using click models for monolingual IR seems reasonable, since instantiations are based on the purpose of the search, which is independent of the query language, and result lists in CLIR contain documents in one language, similar to monolingual IR.

Experimental setup. Online learning experiments are done using the *lerot* framework [19]. We split each query set into 5 parts to perform 5-fold cross validation. The discount factor in Eq. 1 is set as $\gamma = 0.995$, and all experiments are run for 1000 iterations [11]. We repeat all experiments 25 times and average results over folds and repetitions.

Results and Discussion. We first report the *final performance* of the learned ranking function for CLIR after 1,000 iterations for each dataset in Table 3. To gain more insights into the obtained final performance of the online learning to rank algorithm (the DBGD algorithm) for CLIR, we provide two performance in Table 3; **(1)** performance of ListNet [2], one of the representative listwise algorithms for offline learning to rank, since the DBGD algorithm performs listwise learning [9]. The ListNet algorithm uses manual relevance judgments in learning, while the DBGD algorithm learns from relative comparisons of two ranking functions based on user clicks. The results of ListNet, therefore, determine the level of final performance that can be expected from online learning to rank algorithms. For our experiments, we use the RankLib [4] implementation of the ListNet algorithm with the default parameters. The results in Table 3 show that online learning to rank can be successfully adopted for the CLIR setting, since online optimization outperforms the offline learning. However, the improvements are not statistically significant. **(2)** final performance of online learning

Table 3: Final performance of the online learning to rank algorithm (perfect click model) in comparison with the supervised ranking algorithm in terms of NDCG@10.

	2002:En-Es	2002:En-Fr	2002:En-It	2002:Fr-En	2002:It-En	2003:En-Es	2003:En-Fr
Online L2R for CLIR	0.377	0.366	0.377	0.472	0.330	0.423	0.476
ListNet for CLIR	0.321	0.309	0.300	0.406	0.280	0.390	0.407
Online L2R for Monolingual IR	0.475			0.471	0.439	0.478	

Table 4: Online performance in terms of cumulative NDCG over 1000 iterations.

Click model	IR model	2002:En-Es	2002:En-Fr	2002:En-It	2002:Fr-En	2002:It-En	2003:En-Es	2003:En-Fr
Perfect	Cross-lingual	58.753	58.817	58.685	76.628	54.531	66.167	74.059
	Monolingual	78.322			79.562	74.492	77.232	
Navigational	Cross-lingual	53.935	51.938	51.754	68.411	49.956	59.741	65.927
	Monolingual	72.447			72.385	69.955	71.180	
Informational	Cross-lingual	38.471	37.618	38.760	53.207	37.841	41.768	47.707
	Monolingual	55.685			58.847	55.755	57.790	

to rank for the monolingual setting of test collections. One metric to evaluate the performance of CLIR is the percentage compared to the performance of monolingual IR [14]. Therefore, the results of the DBGD algorithm for the monolingual setting of test collections are also reported in Table 3, which show that online learning to rank for the CLIR cases achieves reasonable percentage of that for the monolingual ones, and even performs equivalently for 2002:Fr-En dataset (higher performance of CLIR than monolingual IR for some cases is also reported in [14]). Figure 1 shows the learning curves for 2002:En-Fr dataset, final performance of the learned ranking function at each iteration, for different click models in both monolingual and cross-lingual settings. In both settings, the noisier the click model, the lower the final performance. The learning curve for each click model in the cross-lingual setting has almost the same trend as the one in the monolingual setting.

Table 4 reports the *online performance* using Eq. 1 obtained by different click models for each dataset. The results in the table include the online performance for the CLIR setting in comparison with that for the monolingual setting, which show that the online performance in the CLIR setting achieves acceptable percentage of that in the monolingual setting. The results thus demonstrate that users would not experience low-quality results.

5 CONCLUSION AND FUTURE WORK

In this paper, we studied the optimization of retrieval functions for CLIR based on users' implicit feedback. We demonstrated that although the cross-lingual environment is noisier than monolingual one, the online learning to rank algorithm DBGD can be successfully adopted for learning of personalized ranking models. There are several possible directions for future work. A promising line is to reuse historical data to accelerate the learning speed in the cross-lingual setting. We also would like to investigate how user interactions with monolingual search results can be integrated in the learning process of models for CLIR.

6 ACKNOWLEDGMENTS

This research was in part supported by a grant from the Institute for Research in Fundamental Sciences (No. CS1396-4-51). Any opinions, findings, conclusions, or recommendations expressed in this paper are of the authors, and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Hosein Azarbonyad, Azadeh Shakery, and Heshaam Faili. 2012. Using Learning to Rank Approach for Parallel Corpora Based Cross Language Information Retrieval. In *ECAI*. 79–84.

[2] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *ICML*. 129–136.

[3] Yiwei Chen and Katja Hofmann. 2015. Online Learning to Rank: Absolute vs. Relative. In *WWW*. ACM, 19–20. DOI: http://dx.doi.org/10.1145/2740908.2742718

[4] Van Dang. 2016. https://people.cs.umass.edu/~vdang/ranklib.html. (2016).

[5] Kareem Darwish and Douglas W. Oard. 2003. Probabilistic structured query methods. In *SIGIR*. ACM, 338–344. DOI: http://dx.doi.org/10.1145/860435.860497

[6] Wei Gao, John Blitzer, Ming Zhou, and Kam-Fai Wong. 2009. Exploiting Bilingual Information to Improve Web Search. In *ACL '09*. 1075–1083.

[7] Wei Gao, Cheng Niu, Ming Zhou, and Kam-Fai Wong. 2009. Joint Ranking for Multilingual Web Search. In *ECIR*. 114–125.

[8] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient Multiple-click Models in Web Search. In *WSDM*. 124–131. DOI: http://dx.doi.org/10.1145/1498759.1498818

[9] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. 2013. Reusing Historical Interaction Data for Faster Online Learning to Rank for IR. In *WSDM*. 183–192. DOI: http://dx.doi.org/10.1145/2433396.2433419

[10] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2011. A Probabilistic Method for Inferring Preferences from Clicks. In *CIKM*. 10.

[11] Katja Hofmann, Shimon Whiteson, and Maarten Rijke. 2013. Balancing Exploration and Exploitation in Listwise and Pairwise Online Learning to Rank for Information Retrieval. *Inf. Retr.* 16, 1 (Feb. 2013), 63–90.

[12] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *KDD*. 133–142. DOI: http://dx.doi.org/10.1145/775047.775067

[13] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* 3, 3 (March 2009), 225–331. DOI: http://dx.doi.org/10.1561/1500000016

[14] Jian Yun Nie. 2010. *Cross-Language Information Retrieval.* Morgan & Claypool.

[15] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. 2010. LETOR: A Benchmark Collection for Research on Learning to Rank for Information Retrieval. *Inf. Retr.* 13, 4 (2010).

[16] Filip Radlinski and Thorsten Joachims. 2005. Query Chains: Learning to Rank from Implicit Feedback. In *KDD*. ACM, 239–248.

[17] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. 2008. How Does Clickthrough Data Reflect Retrieval Quality?. In *CIKM*. 10.

[18] Mark Sanderson. 2010. Test Collection Based Evaluation of Information Retrieval Systems. *Found. Trends Inf. Retr.* 4, 4 (2010).

[19] Anne Schuth, Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2013. Lerot: An Online Learning to Rank Framework. In *LivingLab*. 4.

[20] Anne Schuth, Floor Sietsma, Shimon Whiteson, and Maarten de Rijke. 2014. Optimizing Base Rankers Using Clicks: A Case Study using BM25. In *ECIR*.

[21] Anne Schuth, Floor Sietsma, Shimon Whiteson, Damien Lefortier, and Maarten de Rijke. 2014. Multileaved Comparisons for Fast Online Evaluation. In *CIKM*.

[22] Jrg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *LREC*.

[23] Ming-Feng Tsai, Hsin-Hsi Chen, and Yu-Ting Wang. 2011. Learning a Merge Model for Multilingual Information Retrieval. *Inf. Process. Manage.* 47, 5 (Sept. 2011), 635–646.

[24] Nicolas Usunier, Massih-Reza Amini, and Cyril Goutte. 2011. Multiview Semi-supervised Learning for Ranking Multilingual Documents. In *ECML PKDD*. Springer-Verlag, 443–458.

[25] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. 2009. The k-armed dueling bandits problem. In *COLT*.

[26] Yisong Yue and Thorsten Joachims. 2009. Interactively Optimizing Information Retrieval Systems As a Dueling Bandits Problem. In *ICML*. 8.