Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

9-2009

Joint ranking for multilingual web search

Wei GAO
Singapore Management University, weigao@smu.edu.sg

Cheng NIU

Ming ZHOU

Kam-Fai WONG

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Databases and Information Systems Commons

Joint Ranking for Multilingual Web Search[*]

Wei Gao[1], Cheng Niu[2], Ming Zhou[2], and Kam-Fai Wong[1]

[1] The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China
{wgao,kfwong}@se.cuhk.edu.hk
[2] Microsoft Research Asia, No.49, Zhichun Road, Beijing 100190, China
{chengniu,mingzhou}@microsoft.com

Abstract. Ranking for multilingual information retrieval (MLIR) is a task to rank documents of different languages solely based on their relevancy to the query regardless of query's language. Existing approaches are focused on combining relevance scores of different retrieval settings, but do not learn the ranking function directly. We approach Web MLIR ranking within the learning-to-rank (L2R) framework. Besides adopting popular L2R algorithms to MLIR, a joint ranking model is created to exploit the correlations among documents, and induce the joint relevance probability for all the documents. Using this method, the relevant documents of one language can be leveraged to improve the relevance estimation for documents of different languages. A probabilistic graphical model is trained for the joint relevance estimation. Especially, a hidden layer of nodes is introduced to represent the salient topics among the retrieved documents, and the ranks of the relevant documents and topics are determined collaboratively while the model approaching to its thermal equilibrium. Furthermore, the model parameters are trained under two settings: (1) optimize the accuracy of identifying relevant documents; (2) directly optimize information retrieval evaluation measures, such as mean average precision. Benchmarks show that our model significantly outperforms the existing approaches for MLIR tasks.

1 Introduction

Search across multiple languages is desirable with the increase of many languages over the Web. Multilingual information retrieval (MLIR) for web pages however remains challenging because the documents in different languages have to be compared and merged appropriately. It is hard to estimate the cross-lingual relevancy due to the information loss from query translation.

Recently, machine learning approaches for ranking, known as learning-to-rank (L2R), have received intensive attention [2,4,5,20]. The learning task is to optimize a ranking function given the data consisting of queries, the retrieved documents and their relevance judgments made by human. Given a new query, the learned function is used to predict the order of the retrieved documents.

However, there is little research to adapt the state-of-the-art ranking algorithms for MLIR. Existing techniques usually combine query translation and

---

[*] This work was done while the first author visiting Microsoft Research Asia.

monolingual retrieval to derive a relevancy score for each document. Then the relevancy scores from different settings are normalized to be comparable for final combination and ranking [10,15,17]. Such approaches do not directly incorporate any feature to the MLIR relevancy, hence does not work well for multilingual Web search where a large number of relevancy features can be utilized.

Multilingual L2R aims to optimize a unique ranking function for documents of different languages. This can be done intuitively by representing documents within a unified feature space and being approached as a monolingual ranking task. Nevertheless, information loss and misinterpretation from translation makes the relevancy features between query and individual documents (especially in the target language) inaccurate, rendering the multilingual ranking a more difficult problem.

In this work, we propose to leverage the relevancy among candidate documents to enhance MLIR ranking. Because similar documents usually share similar ranks, cross-lingual relevant documents can be leveraged to enhance the relevancy estimation for documents of different languages, hence complement the inaccuracies caused by query translation errors. Given a set of candidate documents, multilingual clustering is performed to identify their salient topics. Then a probabilistic graphical model, called Boltzmann machine (BM) [1,8], is used to estimate the joint relevance probability of all documents based on both of the query-document relevancy and the relevancy among the documents and topics. Furthermore, we train our model by two means: (1) optimizing the accuracy of identifying relevant documents; (2) directly optimizing IR evaluation measures. We show significant advantages of our method for MLIR tasks.

2   Related Work

MLIR is a task to retrieve relevant documents in multiple languages. Typically, the queries are first translated using a bilingual dictionary, machine translation software or a parallel corpus, which is followed by a monolingual retrieval. A re-ranking process then proceeds to merge different ranked lists of different languages appropriately. Existing work is focused on how to combine the incomparable scores associated with each result list. The scores are normalized with the methods like Min-Max [3], Z-score [15], CORI [16], etc., and combined by CombSUM [3] or logistic regression [15] to generate the final ranking score.

Although some work [15,17] involve learning, they are still focused on adjusting the scores of documents from different monolingual result lists, ignoring the direct modeling of various types of features for measuring MLIR relevancy. Recently, Tsai el al. [18] presented a study of learning a merge model by learning the unique ranking function for different features, demonstrating the advantages of L2R for MLIR ranking. Although related to their work, our approach focuses on a new model that can leverage the relevancy among documents of different languages in addition to the commonly used relevancy features for the query and individual documents.

3   Learning for MLIR Ranking

The learning framework for MLIR ranking aims to learn a unique ranking function to estimate comparable scores for documents of different languages. An important step is to design a unified multilingual feature space for the documents. Based on these features, existing monolingual L2R algorithms can be applied for MLIR ranking. We will give details about constructing the multilingual feature space in Section 5. In this section, we introduce the learning framework.

Suppose that each query $q \in Q$ (Q is a given query set) is associated with a list of retrieved documents $D_q = \{d_i\}$ and their relevance labels $L_q = \{l_i\}$, where l_i is the rank label of d_i and may take one of the m rank levels in the set $R = \{r_1, r_2, \ldots, r_m\}$ ($r_1 \succ r_2 \succ \ldots \succ r_m$, where $\succ$ denotes the order relation). So the training corpus can be represented as $\{q \in Q | D_q, L_q\}$.

For each query-document pair (q, d_i), we denote $\Phi : \mathbf{f}(q, d_i) = [f_k(q, d_i)]_{k=1}^K$ as the feature vector, where f_k is one of the relevancy feature functions for (q, d_i). The goal is to learn a ranking function $F : \Phi \to \Re$ ($\Re$ is the real value space) to assign a relevance score for the feature vector of each retrieved document. Specifically, a permutation of integers $\pi(q, D_q, F)$ is introduced to denote the order among the documents in D_q ranked by F, and each integer $\pi(d_i)$ refers to the position of d_i in the result list. Then the objective of ranking is formulated as searching for an optimal function: $\hat{F} = \operatorname{argmin}_F \sum_q E(\pi(q, D_q, F), L_q)$ which minimizes an error function E that represents the disagreement between $\pi(q, D_q, F)$ and the desirable rank order given by L_q over all the queries.

The ranking function and error function have different forms in different ranking algorithms. The standard probabilistic classification (e.g., Support Vector Classifier) or metric regression (e.g., Support Vector Regression) can be used for ranking by predicting rank labels or scores of the documents. Most of the popular ranking models like Ranking SVM (large-margin ordinal regression) [5], RankBoost [4], RankNet [2], etc., aim to optimize the pair-wise loss based on the order preference and classify the relevance order between a pair of documents. More recently, SVM-MAP [20] is proposed to directly optimize IR evaluation measure – Mean Average Precision (MAP).

Under this framework, existing monolingual ranking algorithms can be applied for multilingual ranking in a similar way as [18] using FRank.

4   Joint Ranking Model for MLIR

Although monolingual ranking algorithms can be applied for MLIR, the information loss caused by query translation makes it a more difficult task. To complement the query-document relevancy, we propose a joint ranking model to additionally exploit the relationship among documents of different languages. If two documents are bilingually correlated or similar, and one of them is relevant to the query, it is very likely that the other is also relevant. By modeling the similarity, relevant documents in one language may help the relevance estimation of documents in a different language, and hence can improve the overall relevance

estimation. This can be considered as a variant of pseudo relevance feedback. In our study, Boltzmann machine (BM) [1,8] is used to estimate the joint relevance probability distribution because it is well generalized to model any relationship among objects.

4.1 Boltzmann Machine (BM) Learning

BM is a undirected graphical model that makes stochastic predictions about which state values its nodes should take [1]. The global state $\mathbf{s}$ of the graph is represented by a vector $\mathbf{s} = [s_1 s_2 \ldots s_n]$, where $s_i = \pm 1$ is the state of the node i and n is the total number of graph nodes. The system's energy under a global state is defined as $E(\mathbf{s}) = -\frac{1}{2} \sum_{ij} w_{ij} s_i s_j - \sum_i \theta_i s_i$, where w_{ij} is the edge weight between node i and j, θ_i is the threshold of node i. After some enough time of the dynamics process, the system will reach a thermal equilibrium, where the probability to find the graph in global state depends only on the states of each node and its neighbors, and follows the Boltzmann distribution, i.e., $P(\mathbf{s}) = \frac{1}{Z} \exp(-E(s))$, where $Z = \sum_{\mathbf{s}} \exp(-E(\mathbf{s}))$ is the normalization function over all possible states.

The training of a machine is to resolve the weights and thresholds in such a way that the Boltzmann distribution approximates the target distribution $\tilde{P}(\mathbf{s})$ as close as possible. The difference between the two distributions is measured by *Kullback-Leibler (K-L) Divergence* [9]: $K(\tilde{P}||P) = \sum_{\mathbf{s}} \tilde{P}(\mathbf{s}) \log \frac{\tilde{P}(\mathbf{s})}{P(\mathbf{s})}$. The objective is to minimize the divergence using gradient descent. The weight updating rules of the following form can be obtained:

$$\Delta w_{ij} = \alpha(< s_i s_j >_{clamped} - < s_i s_j >_{free}) \tag{1}$$

$$\Delta \theta_i = \alpha(< s_i >_{clamped} - < s_i >_{free}) \tag{2}$$

where α is the learning rate, and $< . >_{clamped}$ and $< . >_{free}$ denote the expectation values of the node states obtained from the "clamped" and "free-running" stages in training respectively. In clamped stage, states are fixed to the patterns in training data; in free-running stage, states are changed based on the model's stochastic decision rule. The procedure alternates between the two stages until the model converges.

4.2 Joint Relevance Estimation Based on BM

For each query, one can intuitively represent the retrieved documents as nodes, the correlations between them as edges, and the rank label of each document as node state. Then each BM naturally corresponds to the instances of one query. However, the number of edges is quadratic to the number of documents with this representation. This is unacceptable for Web search where hundreds of candidate documents will be returned. Our idea is to first discover the salient topics using a clustering technique, and the direct document connections are replaced by the edges between documents and topics. In particular, only some top largest clusters are kept so that the size of the graph's connectivity is linear with the number of documents.

For the salient topics, we perform multilingual clustering on the retrieved documents of each query q (see Sect. 4.3). We denote q's salient topic set as $T_q = \{t_j\}$. Then T_q and D_q correspond to different types of nodes in the graph. The topic nodes are regarded hidden units because their states (rank labels) are not explicitly provided, while the document nodes are output units as their rank labels will be the output of ranking. Though a document belongs to one topic at most, edges exist between a document node and every topic node, representing the strength of their correlation.

For each q, we denote $\mathbf{sd}_q = [sd_i]$ and $\mathbf{st}_q = [st_j]$ as the state vectors of the document and topic nodes respectively, then the energy of the machine becomes:

$$E(\mathbf{s}, q) = E(\mathbf{sd}_q, \mathbf{st}_q, q) = -\sum_i \Theta \cdot \mathbf{f}(q, d_i) sd_i - \frac{1}{2} \sum_{i,j} \mathcal{W} \cdot \mathbf{g}(d_i, t_j) sd_i st_j \quad (3)$$

where $\mathbf{f} = [f_x(q, d_i)]_{x=1}^{X}$ and $\mathbf{g} = [g_y(d_i, t_j)]_{y=1}^{Y}$ are the X-dimension feature vector of query-document relevancy on document nodes and the Y-dimension document-topic relevancy on edges respectively, and Θ and $\mathcal{W}$ are their corresponding weight vectors. Then the probability of the global state $P(\mathbf{s}, q) = P(\mathbf{sd}_q, \mathbf{st}_q, q)$ follows Boltzmann distribution (see Sect. 4.1).

4.3 Multilingual Clustering for Identifying Salient Topics

For clustering and measuring the relevancy among documents, some translation mechanism has to be employed for comparing the similarity of documents in different languages. We use the cross-lingual document similarity measure described in [12] for its simplicity and efficiency. The measure is a cosine-like function with an extension of TF-IDF weights for the cross-lingual case, using a dictionary for keyword translation. The measure is defined as follows:

$$sim(d_1, d_2) = \frac{\sum_{(t_1, t_2) \in T(d_1, d_2)} tf(t_1, d_1) idf(t_1, t_2) tf(t_2, d_2) idf(t_1, t_2)}{\sqrt{Z'}} \quad (4)$$

where Z' is given as

$$Z' = \left[\sum_{(t_1, t_2) \in T(d_1, d_2)} (tf(t_1, d_1) idf(t_1, t_2))^2 + \sum_{t_1 \in \overline{T}(d_1, d_2)} (tf(t_1, d_1) idf(t_1))^2 \right] \times$$

$$\left[\sum_{(t_1, t_2) \in T(d_1, d_2)} (tf(t_2, d_2) idf(t_1, t_2))^2 + \sum_{t_2 \in \overline{T}(d_2, d_1)} (tf(t_2, d_2) idf(t_2))^2 \right]$$

$T(d_1, d_2)$ denotes the sets of word pairs where t_2 is the translation of t_1, and t_1 (t_2) occurs in document d_1 (d_2). $\overline{T}(d_1, d_2)$ denotes the set of terms in d_1 that have no translation in d_2 ($\overline{T}(d_1, d_2)$ is defined similarly). $idf(t_1, t_2)$ is defined as the extension of the standard IDF for a translation pair (t_1, t_2): $idf(t_1, t_2) = \log \left(\frac{n}{df(t_1) + df(t_2)} \right)$, where n denotes the total number of documents in two languages and df is the word's document frequency. In our work, the cross-lingual

document similarity is measured as such, and the monolingual similarity is calculated by the classical cosine function. K-means algorithm is used for clustering. We introduce only k largest clusters into the graph as salient topics, where k is chosen empirically ($k = 6$ achieves best results in our case) based on the observation that minor clusters are usually irrelevant to the query.

Eq. (4) is also used to compute the edge features, i.e., the relevancy between documents and salient topics. The edge features for each document-topic pair are defined as 12 similarity values based on the following combinations considering three aspects of information: (1) language — monolingual or cross-lingual similarity depending on the languages of two documents concerned; (2) field of text — the similarity is computed based on title, body or title+body; and (3) how to do the average for the value — average the similarity values with all the documents in the cluster or compute the similarity between the document and the cluster's centroid.

4.4   BM Training as a Classifier

The training is to adjust the weights and thresholds in such a way that for each query the predicted probability of document relevancy, i.e., $P(\mathbf{sd}_q, q) = \sum_{\mathbf{st}_q} P(\mathbf{sd}_q, \mathbf{st}_q, q)$, approximates to the target distribution $\tilde{P}(\mathbf{sd}_q, q)$ as closely as possible, where $\tilde{P}(\mathbf{sd}_q, q) = \begin{cases} 1, & \text{if } \mathbf{sd}_q = L_q; \\ 0, & \text{otherwise} \end{cases}$ is obtained from the training data. By minimizing the *K-L Divergence*, we obtain the updating rules

$$\Delta\theta_x = \alpha \sum_{q,i} f_x(q, d_i) \left(< sd_i >_{clamped} - < sd_i >_{free} \right) \tag{5}$$

$$\Delta w_y = \alpha \sum_{q,i,j} g_y(d_i, t_j) \left(< sd_i st_j >_{clamped} - < sd_i st_j >_{free} \right) \tag{6}$$

which have the similar forms as Eq. (1)–(2).

The training procedure alternates between the clamped and the free stages, which needs to repeat several times with different initial weight values to avoid local optima. Unlike an output unit whose state is fixed to its human label in the clamped phase, the state value of a hidden unit (i.e., a topic) is decided by the model in both stages. Note that the exact estimation of the expectation values $< . >_{clamped}$ and $< . >_{free}$ requires enumerating all the possible state configurations. So we use Gibbs sampling [19], a Markov Chain Monte Carlo method, to approximate their values for efficiency.

4.5   BM Inference for MLIR Ranking

For a new query q and the retrieved documents D_q, the relevance probability of a document $d_i \in D_q$ can be estimated by $P(sd_i, q) = \sum_{\mathbf{sd}_q \backslash sd_i, \mathbf{st}_q} P(\mathbf{sd}_q, \mathbf{st}_q, q)$. Then it is straightforward to determine $\hat{l}_i = \text{argmax}_{sd_i} P(sd_i, q)$ as the rank label for ranking and use the value of $P(\hat{l}_i, q)$ to break the tie. However, exact

estimation of $P(sd_i, q)$ is time-consuming since an enumeration of all the possible global states is needed again. For the efficiency of online prediction, we use mean field approximation [6] for the inference. Mean field theory has solid foundation based on variational principle. Here we simply present the procedure of the mean field approximation for BM, and leave the formal justifications to [6].

In mean field approximation, the state distribution of each node only relies on the states of its neighbors which are all fixed to their *average state value*. So given the machine, we have the following:

$$P(sd_i = r) = \frac{\exp\left[\sum_j \mathcal{W} \cdot \mathbf{g}(d_i, t_j) < st_j > r + \Theta \cdot \mathbf{f}(q, d_i)r\right]}{\sum_r \exp\left[\sum_j \mathcal{W} \cdot \mathbf{g}(d_i, t_j) < st_j > r + \Theta \cdot \mathbf{f}(q, d_i)r\right]} \tag{7}$$

$$P(st_j = r) = \frac{\exp\left[\sum_i \mathcal{W} \cdot \mathbf{g}(d_i, t_j)r < sd_i >\right]}{\sum_r \exp\left[\sum_i \mathcal{W} \cdot \mathbf{g}(d_i, t_j)r < sd_i >\right]} \tag{8}$$

$$< sd_i >= \sum_r P(sd_i = r)r \quad (9) \qquad < st_j >= \sum_r P(st_j = r)r \quad (10)$$

where Eq. (7) computes the relevance probability of a document given the average rank labels of all the topics. Similarly, Eq. (8) computes the relevance probability of a topic given the average rank labels of all the documents. Eq. (9) and (10) estimate the average rank labels given the probability distributions computed by Eq. (7) and (8).

Eq. (7)–(10) are called mean field equations, and can be solved using the following iterative procedure for a fixed-point solution:

1. Assume an average state value for every node;
2. For each node, estimate its state value probability using Eq. (7) and (8) given the average state values of its neighbors;
3. Update the average state values for each node using Eq. (9) and (10);
4. Go to step 2 until the average state values converge.

Each iteration requires $O(|T_q| + |D_q|)$ time, being linear to the number of nodes.

4.6   BM Training with MAP Optimization

In the previous sections, BM is optimized for the rank label prediction. However, rank label prediction is just loosely related to MLIR accuracy since the exact relevance labels are not necessary to derive the correct ranking orders. In [20], ranking model directly optimizing IR evaluation measure reports the best ranking performance. Hence, we will train our model in a similar way, i.e., optimizing the MAP of MLIR.

MAP is the mean of average precision over all the queries. We know that the predicted ranking order is produced by $\pi(q, D_q, F)$. Then the average precision for q is defined as $AvgP_q = \frac{\sum_{i=1}^{n(q)} p_q(i)y_i}{\sum_{i=1}^{n(q)} y_i}$, where $n(q)$ is the number of retrieved documents, y_i is assigned with 1 or 0 depending on $d_{i'}$ is relevant or not ($d_{i'}$

is the document ranked at the i-th position, i.e., $\pi(d_{i'}) = i$), and $p_q(i)$ is the precision at the rank position of i: $p_q(i) = \frac{1}{i}\sum_{j<i} y_j$.

Instead of simply maximizing MAP, we maximize the following objective:

$$MAP - C\sum_y ||w_y||^2 - C\sum_x ||\theta_x||^2 \tag{11}$$

where the last two terms are L-2 regularization terms representing the complexity of the model. So it is a tradeoff between the model's accuracy and complexity controlled by C. SVM-MAP [20] used a similar function to minimize a linear combination of the same L-2 norm with the hinge loss relaxation of MAP loss.

Because MAP is not a continuous function with the weights of the BM, Powell's Direction Set Method [13], which does not involve derivation computations, is used for the optimization. To achieve the optimal performance, Powell's method is repeatedly called many times with different initial values of the BM's weights. One particular set of the initial values is the weights learned when the BM is trained to optimize classification accuracy in Sect. 4.4. The mean field approximation (Sect. 4.5) is used in model inference as well.

5   Experiments and Results

We evaluated the proposed MLIR ranking algorithms. The experiments are conducted on two datasets: (1) TREC5&6 English-Chinese CLIR data; (2) Chinese and English multilingual Web search data. The baseline is the ranking score combination algorithm, referred to as *ScoreComb* below. Specifically, different ranking algorithms including Ranking SVM and SVM-MAP are used to learn ranking functions for Chinese and English documents separately. Then the scores are combined by a log linear model following [15,17].

Three prevalent L2R algorithms, i.e., SVC (SVM classifier with probability estimation), RSVM (Ranking SVM), and SVM-MAP, are used to compare the performance of the MLIR ranking. These algorithms represent three typical categories of ranking schemes: (1) SVC is a typical classification-based ranking algorithm; (2) RSVM is the state-of-the-art ranking algorithm based on pair-wise preference order classification; (3) SVM-MAP is a ranking algorithm directly optimizing IR relevancy measure. We used the source codes of LibSVM[1], SVM-Light[2] and SVM-map[3] to run SVC, RSVM and SVM-MAP, respectively.

The proposed BM classifier (BMC) and BM classifier with MAP optimizer (BMC-MAP) are also performed for comparison. In order to directly assess the contribution of the relevancy among documents, we reduced BMC and BMC-MAP into the conventional log linear models by simply removing the hidden units and the edges. This produces two more corresponding systems to compare, namely LOG and LOG-MAP.

---

[1] http://www.csie.ntu.edu.tw/∼jlin/libsvm
[2] http://svmlight.joachims.org/
[3] http://projects.yisongyue.com/svmmap/

5.1 Experiments on TREC CLIR Data

We study the contribution of cross-lingual document similarity on CLIR. The CLIR task of TREC5&6 is defined as using English queries to retrieve Chinese documents. Although the multilingual result merge is not required, it is valuable to study the effectiveness to improve cross-lingual relevance estimation for Chinese documents by exploiting the relevant documents of English. Because the joint ranking model requires English retrieval, we additionally index the English TIPSTER corpus from LDC. We use query CH1-28 (TREC5 topics) for training and CH29-54 (TREC6 topics) for testing.

Three free machine translation engines are used to translate English queries into Chinese, and then an Okapi-BM25 (BM25) model [14] is employed for Chinese document retrieval based on the combined query translations. For learning the ranking models, we implement 25 commonly used query-document relevancy features in the literature [11] based on the translated queries, including the scores of TFIDF, BM25, and language modeling IR, etc.

To create BM for joint relevance ranking, 500 English documents are retrieved from TIPSTER using the original query, and are ranked by the BM25 scores. Since there is no relevancy annotation for English documents, we choose 20 documents and assign them one of the following two labels: 0 for the last 10 documents in the result; 1 the top 10. During both training and inference, the states of the English document nodes are fixed to one of the above values. This assumes that top-10 (last-10) English documents are relevant (irrelevant).

The CLIR results are given in Table 1 using average precision (AP) and 11-point precision-recall measures. Since no multilingual result merge is involved, the BM25 score between the translated query and the Chinese document is used as the baseline. Obviously, all the learning algorithms outperform the BM25 baseline. Furthermore, SVM-MAP outperforms RSVM and SVC, and BMC-MAP outperforms BMC, implying that directly optimizing IR measure is also critical to CLIR ranking.

Table 1. TREC6 CLIR performance by 11-point precision-recall and AP measure

recall	BM25	SVC	RSVM	SVM-MAP	LOG	BMC	LOG-MAP	BMC-MAP
0	0.658	0.736	0.788	0.798	0.715	0.796	0.797	0.815
0.1	0.495	0.476	0.531	0.598	0.475	0.583	0.592	0.591
0.2	0.411	0.393	0.427	0.486	0.391	0.469	0.480	0.502
0.3	0.345	0.354	0.385	0.414	0.349	0.412	0.411	0.423
0.4	0.289	0.324	0.346	0.368	0.324	0.367	0.366	0.376
0.5	0.251	0.282	0.299	0.316	0.281	0.312	0.315	0.323
0.6	0.203	0.222	0.241	0.245	0.214	0.247	0.241	0.269
0.7	0.164	0.174	0.200	0.185	0.175	0.183	0.182	0.220
0.8	0.074	0.099	0.101	0.086	0.099	0.088	0.084	0.107
0.9	0.010	0.020	0.027	0.016	0.018	0.017	0.016	0.030
1.0	0.002	0.007	0.012	0.006	0.004	0.007	0.006	0.008
AP	0.249	0.253	0.280	0.301	0.250	0.299	0.299	0.314

We further conducted t-test, which shows that BMC significantly outperforms LOG ($p = 0.009$) and RSVM ($p = 0.011$). This indicates the effectiveness of utilizing monolingual IR results to enhance CLIR. The AP improvement from SVM-MAP and LOG-MAP to BMC-MAP is not as large as from LOG to BMC. This less enhancement may be caused by optimizing Eq. (11). Different from SVM-MAP training which achieves global optimum, BMC-MAP training only achieves a sub-optimal solution. However, although suffered from under-training, BMC-MAP still significantly outperforms SVM-MAP by 4.15% ($p = 0.032$).

5.2 MLIR Experiments on Web Search Data

Multilingual Web Search Data. Our Web search data consists of queries and returned web pages from query logs of a commercial search engine. There are two separate monolingual query logs for English and Chinese. The retrieved web pages are annotated with ratings from 0 (irrelevant) to 5 (perfect) by human labelers. For each web page of a given query, query-dependent features are extracted from the query combined with four different sources: the anchor text, the URL, the document title and the body. Some query-independent features are also extracted, such as PageRank. There are 352 such features in total for each one of the two languages.

For the multilingual ranking, we manually select 1,000 queries which are in the English query log and their translations are in the Chinese log. Based on these queries and their labeled results, we construct a bilingual ranking corpus: Given an English query, the corresponding Chinese and English web pages associated with the rank labels are put together. This brings 17,791 Chinese and 32,049 English pages in total. In addition, the edge features specific to our joint model, i.e., the 12 similarities measuring the correlations between documents and salient topics, are also computed. All the model parameters are tuned on a development set with 197 queries and 803 queries are used for 4-fold cross validation.

Experiments on Multilingual Ranking. The results of MAP, precision@1, 5,10 and NDCG@1,5,10 (NDCG — Normalized Discounted Cumulative Gain [7]) are presented in Fig. 1. Apparently, all the models learned using the multilingual feature space outperform the *ScoreComb* baseline. The t-test shows that all improvements are statistically significant ($p < 0.05$). This confirms the advantage of the L2R approaches which directly learn a ranking function from features.

By optimizing the ranking order of document pairs, RSVM is usually believed to perform better than SVC. This is confirmed by our MLIR results. Similar as the TREC result, BMC achieves comparable results with RSVM, implying that classification-based ranking algorithms, by making use of the relevancy among individual documents, can perform equally well with the state-of-the-art ranking models. Interestingly, SVM-MAP underperforms RSVM. This may be because SVM-MAP cannot exploit the fine-grained 6-level relevance while RSVM can.

BMC-MAP outperforms all other models. In terms of MAP, it outperforms the baseline by 30.22% ($p = 0.003$), SVC by 15.12% ($p = 0.006$), BMC by 5.33% ($p = 0.029$), RSVM by 3.90% ($p = 0.023$), and SVM-MAP by 7.40% ($p = 0.009$).
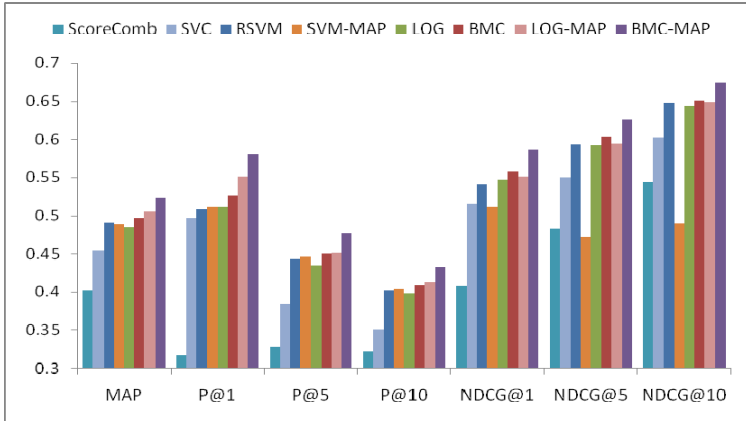
Fig. 1. Comparison of ranking results using multilingual Web search data

Table 2. The comparison results of using and without using clusters in BM models

	MAP	P@1	P@5	P@10	NDCG@1	NDCG@5	NDCG@10
LOG	0.484	0.511	0.435	0.397	0.546	0.591	0.641
BMC	0.497	0.527	0.451	0.409	0.557	0.604	0.651
LOG-MAP	0.504	0.552	0.452	0.413	0.551	0.594	0.649
BMC-MAP	0.523	0.580	0.478	0.432	0.587	0.626	0.674

Table 2 shows the enhancement from the joint ranking model by comparing BMC (BMC-MAP) with LOG (LOG-MAP). The p-value on MAP difference is 0.04 between BMC and LOG, and is 0.027 between BMC-MAP and LOG-MAP, implying the significant contribution of the inter-document relevancy.

6 Conclusion and Future Work

We studied to rank web pages of different languages based on their relevancy to the query using the learning-to-rank framework. By constructing a unified multilingual feature space, popular L2R algorithms are applied to MLIR ranking, and significantly outperform the score combination baseline. For further improvement, a joint ranking model is proposed to exploit document similarities in addition to the commonly used query-document relevancy. This new model first uncovers salient topics among retrieved documents, and then collaboratively identifies relevant documents and topics using their content similarities. Significant ranking enhancement is achieved. Our model is a generic ranking mechanism. Besides the content similarity, any types of relationship among web pages from different languages, such as structural similarity, hyperlink relation, etc., will be used to improve ranking in our future work.

References

1. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.A.: Learning Algorithm for Boltzmann Machines. Cognitive Science 9, 147–169 (1985)
2. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to Rank using Gradient Descent. In: Proc. of ICML, pp. 89–96 (2005)
3. Fox, E.A., Shaw, J.A.: Combination of Multiple Searches. In: Proc. TREC-2 (1994)
4. Freund, Y., Iyer, R., Schapire, R., Singer, Y.: An Efficient Boosting Algorithm for Combining Preferences. Journal of Machine Learning Research 4, 933–969 (2004)
5. Herbrich, R., Graepel, T., Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. In: Advances in Large Margin Classifiers. MIT Press, Cambridge (2000)
6. Jaakkola, T.S.: Variational Methods for Inference and Estimation in Graphical Models. Ph.D. Thesis, MIT (1997)
7. Järvelin, K., Kekäläinen, J.: IR Evaluation Methods for Retrieving Highly Relevant Documents. In: Proc. of ACM SIGIR, pp. 41–48 (2000)
8. Ko, J., Luo, S., Nyberg, E.: A Probabilistic Graphical Model for Joint Answer Ranking in Question Answering. In: Proc. of ACM SIGIR, pp. 343–350 (2007)
9. Kullback, S.: Information Theory and Statistics. John Wiley & Sons Press, NY (1959)
10. Lin, W.-C., Chen, H.-H.: Merging Mechanisms in Multilingual Information Retrieval. In: Peters, C., Braschler, M., Gonzalo, J. (eds.) CLEF 2002. LNCS, vol. 2785, pp. 175–186. Springer, Heidelberg (2003)
11. Liu, T.-Y., Xu, J., Qin, T., Xiong, W.Y., Li, H.: LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval. In: Proc. of ACM Workshop on Learning to Rank for Information Retrieval, Amsterdam, The Netherland (2007)
12. Mathieu, B., Besancon, R., Fluhr, C.: Multilingual Document Clusters Discovery. In: Proc. of RIAO, pp. 1–10 (2004)
13. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C: The Art of Scientific Computing (§10.5). Cambridge University Press, Cambridge (1992)
14. Robertson, S.E., Walker, S., Hancock-Beaulieu, M.M., Gatford, M.: OKAPI at TREC-3. In: Proc. of TREC-3, pp. 109–128 (1995)
15. Savoy, J., Berger, P.Y.: Selection and merging strategies for multilingual information retrieval. In: Peters, C., Clough, P., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B. (eds.) CLEF 2004. LNCS, vol. 3491, pp. 27–37. Springer, Heidelberg (2005)
16. Si, L., Callan, J.A.: Semi-supervised Learning Method to Merge Search Engine Results. ACM Transaction on Information Systems 21(4), 457–491 (2003)
17. Si, L., Callan, J.A.: CLEF 2005: Multilingual retrieval by combining multiple multilingual ranked lists. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, pp. 121–130. Springer, Heidelberg (2006)
18. Tsai, M.-F., Wang, Y.-T., Chen, H.-H.: A Study of Learning a Merge Model for Multilingual Information Retrieval. In: Proc. of ACM SIGIR, pp. 195–202 (2008)
19. Walsh, B.: Markov Chain Monte Carlo and Gibbs Sampling. Lecture Notes for EEB 596z (2002),
   http://nitro.biosci.arizona.edu/courses/EEB596/handouts/Gibbs.pdf
20. Yue, Y., Finley, T., Radlinski, F., Joachims, T.: A Support Vector Method for Optimizing Average Precision. In: Proc. of ACM SIGIR, pp. 271–278 (2007)