Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Effective Learning to Rank Persian Web Content

Amir Hosein Keyhanipour

Assistant Professor, Computer Engineering Department, Faculty of Engineering, College of Farabi, University of Tehran, Iran. ORCID: 0000-0003-4137-9494. E-mail: keyhanipour@ut.ac.ir

Abstract

Persian language is one of the most widely used languages in the Web environment. Hence, the Persian Web includes invaluable information that is required to be retrieved effectively. Similar to other languages, ranking algorithms for the Persian Web content, deal with different challenges, such as applicability issues in real-world situations as well as the lack of user modeling. CF-Rank, as a recently proposed learning to rank data, aims to deal with such issues by the classifier fusion idea. CF-Rank generates a few click-through features, which provide a compact representation of a given primitive dataset. By constructing the primitive classifiers on each category of click-through features and aggregating their decisions by the use of information fusion techniques, CF-Rank has become a successful ranking algorithm in English datasets. In this paper, CF-Rank is customized for the Persian Web content. Evaluation results of this algorithm on the dotIR dataset indicate that the customized CF-Rank outperforms baseline rankings. Especially, the improvement is more noticeable at the top of ranked lists, which are observed most of the time by the Web users. According to the NDCG@1 and MAP evaluation criteria, comparing the CF-Rank with the preeminent baseline algorithm on the dotIR dataset indicates an improvement of 30 percent and 16.5 percent, respectively.

Keywords: Learning to rank; Persian language; CF-Rank algorithm; dotIR dataset; Information fusion.

DOI: 10.22059/jitm.2019.284726.2377

Introduction

Ranking of search results is a vital task in Web information retrieval systems deals with different challenges. Traditional ranking algorithms could not handle the huge number of the influencing

factors in the relevance of a document to a given query, as well as the dynamic nature of Web data and Web users. Learning to rank (L2R) as a novel and interesting trend incorporates machine learning algorithms to handle such difficulties. Consequently, a tremendous number of L2R algorithms have been proposed in recent years. However, there are major concerns about their applicability in real-world situations. First, L2R methods have to use a huge number of features related to the users' queries and Web documents. Preparing such a prerequisite would not be easy in practice.

Besides, search as a user-centric process, is dependent on the history of users' interactions with Web search engines. Some research works have approved that click-through data is useful in the retrieval process (Dou, Song, Yuan, & Wen, 2008) (Cen, et al., 2009) (Macdonald, Santos, & Ounis, 2012). Unfortunately, almost none of the L2R datasets include such useful data (Keyhanipour, Moshiri, & Rahgozar, 2015). Therefore, most of the state of the art L2R methods suffer from not incorporating such important data in their ranking processes.

To handle such difficulties, the CF-Rank as a newly proposed L2R algorithm introduces the click-through features concept (Keyhanipour, Moshiri, & Rahgozar, 2015). These features are categorized to be either query and document-related, or click-related. Click-through features that could be extracted from a given L2R benchmark dataset provide a very compressed and efficient representation of the primitive dataset. Construction of base classifiers at the top of each category of click-through features and application of information fusion techniques to aggregate their local decisions are other major steps of the CF-Rank. Its successful evaluation with standard L2R datasets was our motivation to customize it for the Persian language as one of the most widely used languages of the Web (W3Techs, 2019). The customization process includes finding appropriate scenarios for the generation of click-through features, as well as the proposition of some settings for the construction of base classifiers and the integration of their votes. In this way, the major contributions of this research are:

- Proposition of some effective scenarios for click-through feature generation in dotIR benchmark dataset, as the only available learning to rank dataset in the Persian language. This achievement would be more noticeable when regarding that dotIR does not include any explicit click-through data.
- Identification of the appropriate configurations for the classifier generation and classifier fusion phases of the CF-Rank algorithm.
- Evaluating the performance of the customized CF-Rank in the Persian Web in comparison with baseline ranking methods.

Literature Review

Learning to rank as a new trend has achieved noticeable attention from research communities in recent years. Subsequently, a large number of learning to rank algorithms are proposed.

Formally, L2R algorithms applied machine learning techniques to learn the optimal way of combining features extracted from query-document pairs through discriminative training (Liu, 2011). However, learning to rank techniques suffer from some substantial problems. Using a large number of features related to query-document pairs is a major challenge on the applicability of such techniques in real-world problems. Besides, search as a user-centric process is very dependent on the history of users' interactions with the search engines. In fact, the importance of users' click-through data in the enhancement of the retrieval systems, has been reported in different researches (Dou, Song, Yuan, & Wen, 2008; Cen, et al., 2009; Macdonald, Santos, & Ounis, 2012). However, this fact is not considered in the preparation of well-known L2R benchmark datasets such as Microsoft's LETOR (Qin & Liu, 2013) and Yahoo L2R Dataset (Chapelle & Chang, 2011). In fact, most of the available learning to rank datasets do not include histories of users' interactions with underlying search engines during their search sessions.

Recently, some research works have tried to handle such challenges of L2R algorithms. For example, CF-Rank has introduced the click-through features concept, as a way to compact the primitive L2R datasets while enriching them by generating some implicit click-through data. Based on the reported experimentations, it has outperformed some well-known ranking methods based on evaluation criteria such as Precision and NDCG (Keyhanipour, Moshiri, & Rahgozar, 2015). This algorithm has made the best improvement at the top of the ranked lists, which are more visited by Web users (AdvancedWebRanking, 2019).

On the other hand, the usage statistics show that the Persian language is the eighth most widely used language in Web documents (W3Techs, 2019). According to this report, the Persian language is used by 2.0 percent of all Websites whose content language is recognizable. However, a few research works are accomplished about the effective Web information retrieval for the Persian language. In (Hashemi, Yazdani, Shakery, & Naeini, 2010), a number of basic ranking algorithms are aggregated by the use of some aggregation operators. The evaluation of this algorithm is accomplished in the dotIR dataset (Darrudi, et al., 2009), as the only available benchmark dataset for the evaluation of information retrieval algorithms in the Persian language. Authors of (Khodadadian, Ghasemzadeh, Derhami, & Mirsoleimani, 2012; Derhami, Khodadadian, Ghasemzadeh, & Zareh Bidoki, 2013) have proposed a new connectivity-based ranking algorithm, called RL_Rank. Their key idea is to formulate the ranking as a reinforcement learning (RL) problem. They also have introduced a new hybrid approach using the combination of BM25 as a content-based algorithm and RL_Rank. Both proposed algorithms are evaluated by dotIR and TREC 2003 part of the LETOR (Qin, Liu, Xu, & Li, 2007) benchmark datasets. Recently in (Derhami, Paksima, & Khajeh, 2019), a ranking algorithm called RRLUFF is introduced, in which the ranking system is considered as the agent of the learning system and the selection of documents is displayed to the user as the agent's action. In the assessment of the RRLUFF algorithm, dotIR and OHSUMED part of the LETOR (Qin, Liu, Xu, & Li, 2007) benchmark datasets are utilized.

In this paper, the CF-Rank algorithm (Keyhanipour, Moshiri, & Rahgozar, 2015) is customized for the retrieval of the Persian Web content. The customization process includes three different important steps. In the first step, some scenarios are proposed for the generation of click-through features. These scenarios are dependent on the primitive features of the utilized L2R dataset. Thereafter, some configurations are investigated to construct base classifiers at the top of each category of click-through features. The final step is related to the fusion of decisions of these classifiers to find the relevance score of each query-document pair.

Review of the CF-Rank Algorithm

CF-Rank is the application of classifier fusion techniques in the L2R problem. The novelty of the CF-Rank algorithm is that it applies classifier fusion techniques on classifiers derived for the click-through features, not raw features of L2R datasets (Keyhanipour, Moshiri, & Rahgozar, 2015). Click-through features are based on the Click-through data and are made of the conversion of the primitive L2R datasets. Such features provide a compact secondary representation of a given original L2R dataset. In this way, CF-Rank includes three different steps.

The first step of the CF-Rank is click-through data concept (Joachims, 2002), which is shown to be important in the web information retrieval and learning to rank process (Dou, Song, Yuan, & Wen, 2008) (Cen, et al., 2009) (Macdonald, Santos, & Ounis, 2012). Motivated by this observation, the aim of this step is to extract data related to the users' interactions from the available data of the given primitive dataset. Click-through data are defined as a triplet <Q, R, C> consisting of the user query Q, the ranking R presented to the user, and the set C of links the user clicked on (Joachims, 2002). Click-through features are defined in the same way. They are eight features, which are defined in three categories: F_Q, F_R and F_C (Keyhanipour, Moshiri, & Rahgozar, 2015). In this step, click-through features are generated based on the available data of the given primitive dataset. These features could be categorized into three different categories and are related to either the users' queries, resultant documents or clicks of the users. The listing of click-through features is presented in Equation 1.

$$\begin{aligned} F_Q &= \{Repetition, QScore, ResultsAmount\} \\ F_R &= \{AbsoluteRank, StreamLength\} \\ F_C &= \{Specificity, Attractiveness, ClickRate\} \end{aligned} \quad (1)$$

In this setting, F_Q contains features that are related to the nature of the users' queries. The *Repetition* is related to the frequency of the query terms of the user in different parts of a Web document including URL, title and content. *QScore* indicates the score of a document with respect to a given query. It is an aggregation of query-dependent ranking techniques such as Vector Space Model and Language Models. The *ResultAmount* specifies the number of results

retrieved for a given query. Features of the category F_R are related to the sole of Web documents independent of any query. For instance, the *AbsoluteRank* feature, which indicates the absolute rank of a Web document, is heavily related to query-independent rankings such as PageRank. The *StreamLength* as an indicator of the length of a given document is a combination of the length of a document's URL, title and content. The category F_C includes features that are related to interactions of users with ranked lists of results. For example, the *Specificity* of a given document shows how much that document is specific to some particular users' queries. The *Attractiveness* of a given Web page measures the amount of the attention that users pay that page in their search sessions. In the case of the availability of users' interaction logs, it would be related to the order of the users' clicks. When such data is not available, indicators such as in-link or PageRank may be useful to estimate the *Attractiveness* of a Web document. The last feature in this category is *ClickRate*, which illustrates the number of users' clicks within different search sessions. The *ClickRate* could be assumed to be related to query-dependent and query-independent relevance scores.

In fact, this step provides an informative and compact representation of the primitive dataset. So each pair of query-document in the original dataset will be presented by a feature vector, which consists of eight click-through features:

$$\vec{F}_{8\times1}(q,d) = (Repetition, QScore, ResultsAmount, AbsoluteRank, StreamLength, Specificity, Attractiveness, ClickRate) \quad (2)$$

The noticeable point is these click-through features could be generated from any given L2R dataset even when the click-related features are not presented in the primitive dataset. This process is done via some heuristic scenarios that are heavily based on the primitive features presented in the original dataset. One of the main contributions of this research work is to identify suitable click-through feature generation scenarios for the Persian Web data. In our experimentations in the dotIR dataset, the best-investigated scenarios are listed in Table 2.

The second step of the CF-Rank is devoted to the generation of three basic classifiers on categories of click-through features, which are either query-dependent, document-related or click-associated ones. Within this step, the features of each category are used in the classifier generation process. In each category of click-through features, different information measures such as *MAP*, *NDCG*, *InfoGain*, and *OneR* are used in the classifier generation. Classifiers corresponding to each category of click-through features, simply provide a weighted sum of features of that category as their outputs. Weights of click-through features could be easily calculated by the use of a variety of quality indicators such as *MAP* (Manning, Raghavan, & Schütze, 2008), *MeanNDCG* (Manning, Raghavan, & Schütze, 2008), *InfoGain* (Mitchell, 1997) and *OneR* (Holte, 1993). Equation 3 presents this mechanism.

$$Score_Q(q,d) = \frac{\sum_{i=1}^{|Q|} fw_{Q_i} F_{Q_i}(q,d)}{\sum_{i=1}^{|Q|} w_i}, \quad Score_R(q,d) = \frac{\sum_{i=1}^{|R|} fw_{R_i} F_{R_i}(q,d)}{\sum_{i=1}^{|R|} w_i},$$
$$Score_C(q,d) = \frac{\sum_{i=1}^{|C|} fw_{C_i} F_{C_i}(q,d)}{\sum_{i=1}^{|C|} w_i}, \tag{3}$$

where: $|Q| + |R| + |C| = \#Clik\_through\ features = 8$, and parameters fw_{Q_i}, fw_{R_i} and fw_{C_i} denote the weights of click-through features which will be the mean of their *MAP*, *MeanNDCG*, *InfoGain* or *OneR* values on all query-document pairs. Clearly, the outcome of this step is dependent on the data provided by the primitive dataset. As mentioned before, in our experiments, the average of the above-mentioned measures are used on the classifier generation phase.

Finally, in the third step, primitive classifiers designed in the previous step, are aggregated by the use of information fusion techniques. The intuition behind this step is that previously generated classifiers decide about relevance scores of query-document pairs from their own perspective, which is restricted either to the nature of the query, characteristics of web page or clicks of the users. Aggregation of these local decisions seems to be promising. The aggregation is simply accomplished by the weighted sum of the votes of these classifiers. This idea is shown in Equation 4.

$$\forall q,d:\ Score(q,d) = w_Q Score_Q(q,d) + w_R Score_R(q,d) + w_C Score_C(q,d) \tag{4}$$

These weights are determined by the use of a well-known family of information fusion techniques named Ordered Weighted Averaging (OWA). In our evaluations, we have used Optimistic and Pessimistic OWA operators for finding the weights of the classifiers (Filev & Yager, 1994).

The Ordered Weighted Operators (OWA) operators were introduced by Yager (Yager, 1988). An OWA operator of dimension n is a mapping $f: R^n \to R$, that associates objects $a_1, a_2, \dots, a_n$ with weighting vector $\overrightarrow{W} = [w_1, w_2, \dots, w_n]^T$ such that:

$$\sum_{i=1}^{n} w_i = 1, \qquad \forall i: w_i \in [0,1], \qquad and\ f(a_1, a_2, \dots, a_n) = \sum_{j=1}^{n} w_j b_j \tag{5}$$

In Equation 5, b_j is the j^{th} largest element of the collection of n aggregate objects $a_1, a_2, \dots, a_n$ (Busa-Fekete, Kégl, Éltető, & Szarvas, 2013). The function value $f(a_1, a_2, \dots, a_n)$ determines the aggregated value of arguments, $a_1, a_2, \dots, a_n$. The family of Exponential OWA operators is one of the best-known solutions proposed for determining proper weights of the aggregation

arguments (Filev & Yager, 1994). Equations 6 and 7 present formulae for calculating Optimistic and Pessimistic Exponential OWA weights.

$$w_1 = \alpha;\ w_2 = \alpha(1-\alpha); w_3 = \alpha(1-\alpha)^2; \ldots; w_{n-1} = \alpha(1-\alpha)^{n-2};$$
$$w_n = (1-\alpha)^{n-1};\ 0 \le \alpha \le 1 \tag{6}$$

$$w_1 = \alpha^{n-1};\ w_2 = (1-\alpha)\alpha^{n-2}; w_3 = (1-\alpha)\alpha^{n-3}; \ldots; w_{n-1} = (1-\alpha)\alpha;$$
$$w_n = (1-\alpha);\ 0 \le \alpha \le 1 \tag{7}$$

In both equations, w_i stands for the weight assigned to the feature with i^{th} largest value and the parameter α belongs to the unit interval. Since this class of aggregation operators runs between the Max (or) and the Min (and), in (Filev & Yager, 1994), the *Orness* measure was suggested to demonstrate the type of aggregation being performed for a particular weighting vector. Filev & Yager have shown that parameter α is related to the *Orness* measure. In addition, we have: $\forall i \in [1,n]: 0 \le w_i \le 1,\ and \sum_{i=1}^{n} w_i = 1$ (Filev & Yager, 1994). Figure 1 represents a graphical overview of the steps of CF-Rank.

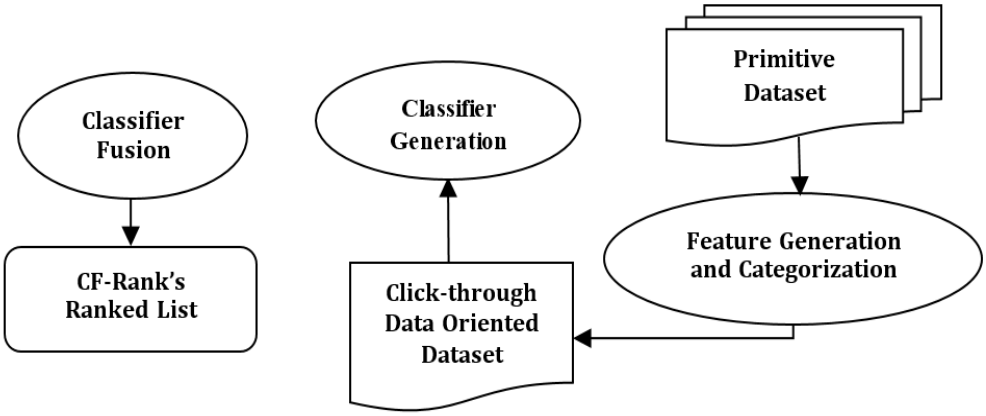

Figure 1. Steps of the CF-Rank algorithm (Keyhanipour, Moshiri, & Rahgozar, 2015)

Experimentation Settings

This section includes four different subsections. The first one introduces the dotIR benchmark dataset and describes its characteristics and structure. The second section is devoted to the scenarios suggested for extracting click-through features from dotIR dataset (Darrudi, et al., 2009). The third section presents a brief description of the evaluation criteria used in the assessment of the CF-Rank in the Persian Web data. The last section contains the details of the tentative results as well as their analysis.

1. dotIR Benchmark Dataset

The dotIR dataset is the only benchmark dataset for the evaluation of information retrieval algorithms in the Persian Web area. This dataset is developed by the University of Tehran (Darrudi, et al., 2009). It contains a set of about 8.5 million web documents gathered in 2009 from the .ir domain, as the Iranian national Web domain. DotIR contains 50 users' queries and 1,000 Web documents corresponding to each query, which form 50,000 query-document pairs. For each pair of query-document, 56 features are extracted which their listing is presented in Appendix A. There is no feature related to the search behavior of users in this dataset. Binary human judgment about the relevance level for each is pair is available.

To facilitate the evaluation of different retrieval algorithms, dotIR is divided into five different folds. Each Fold contains a training set, a validation set, and a test set. Details of these settings is presented in Table 1.

Table 1. Train setting of dotIR benchmark dataset (Darrudi, et al., 2009)

Fold	Training Set	Validation Set	Test Set
$Fold_1$	$\{S_1,S_2,S_3\}$	S_4	S_5
$Fold_2$	$\{S_2,S_3,S_4\}$	S_5	S_1
$Fold_3$	$\{S_3,S_4,S_5\}$	S_1	S_2
$Fold_4$	$\{S_4,S_5,S_1\}$	S_2	S_3
$Fold_5$	$\{S_5,S_1,S_2\}$	S_3	S_4

Each row of the dataset corresponds to a query-document pair. Figure 2 shows the general structure of the dotIR dataset. Label stands for the relevance level of the corresponding pair of query-document.

Label qid:queryID 1:F_1Value 2:F_2Value 3:F_3Value … 55:F_{55}Value 56:F_{56}Value #docid = docID

Figure 2. Structure of dotIR benchmark dataset (Darrudi, et al., 2009)

2. Click-through Feature Generation Scenarios

In order to evaluate the CF-Rank on the dotIR dataset, some click-through feature generation scenarios will be examined. In the design of these scenarios, the conceptual meaning of click-through features are tired to be compromised with features presented in the dotIR dataset. Table 2 lists the most successful scenarios.

Table 2. Click-through Feature Generation Scenarios Used in the Evaluation of the CF-Rank algorithm

Scenario ID	Click-through Feature Calculation Mechanism
IR-DF1	$Q:\begin{cases} Repetition = F_{15} \\ QScore = F_{49} \times \prod_{i=37}^{40} F_i \\ ResultAmount = 1000 \end{cases}$ $R:\begin{cases} AbsoluteRank = F_{42} \times F_{51} \\ StreamLength = F_{20} \end{cases}$ $C:\begin{cases} Specificity = F_{54} \times F_{55} \\ Attractiveness = F_{43} \times F_{46} \times F_{51} \times F_{52} \\ ClickRate = Attractiveness \times QScore \times AbsoluteRank \end{cases}$
IR-DF2	$Q:\begin{cases} Repetition = F_{15} \\ QScore = F_{49} \times \prod_{i=37}^{40} F_i \\ ResultAmount = 1000 \end{cases}$ $R:\begin{cases} AbsoluteRank = F_{42} \times F_{51} \\ StreamLength = F_{20} \end{cases}$ $C:\begin{cases} Specificity = F_{54} \times F_{55} \\ Attractiveness = \prod_{i=50}^{52} F_i \\ ClickRate = Attractiveness \times QScore \times AbsoluteRank \end{cases}$
IR-DF3	$Q:\begin{cases} Repetition = F_{15} \\ QScore = F_{49} \times \prod_{i=37}^{40} F_i \\ ResultAmount = 1000 \end{cases}$ $R:\begin{cases} AbsoluteRank = F_{42} \times F_{51} \\ StreamLength = F_{20} \end{cases}$ $C:\begin{cases} Specificity = F_{54} \\ Attractiveness = F_{43} \times F_{46} \times F_{51} \times F_{52} \\ ClickRate = Attractiveness \times QScore \times AbsoluteRank \end{cases}$
IR-DF4	$Q:\begin{cases} Repetition = F_{15} \\ QScore = F_{49} \times \prod_{i=37}^{40} F_i \\ ResultAmount = 1000 \end{cases}$ $R:\begin{cases} AbsoluteRank = F_{42} \times F_{51} \\ StreamLength = F_{20} \end{cases}$ $C:\begin{cases} Specificity = F_{54} \\ Attractiveness = \prod_{i=50}^{52} F_i \\ ClickRate = Attractiveness \times QScore \times AbsoluteRank \end{cases}$

In all of these scenarios, the same definition is used for query-dependent click-through features. This situation is also observable in result-dependent click-through features. In other words, the difference between these scenarios is limited to their definitions of click-related features. Notice that dotIR does not include any explicitly click-related feature, but we can derive click-through features related to the users' interactions with the result lists. In the category of click-through features, a similar definition is presented for the *ClickRate* feature. So the dissimilarity of the above scenarios is only related to their interpretation of the *Specificity* and *Attractiveness* features. In IR-DF3 and IR-DF4 scenarios, the *Specificity* of a given Web page is thought to be only related to the depth of that page in the tree-map of the corresponding website, which is assumed to inferable from the *"Number of slash in URL"*. In contrast, the IR-DF1 and IR-DF2 scenarios suppose that *Specificity* is calculable by the synergistic combination of *"Number of slash in URL"* and *URL-Length* features. Also in IR-DF1 and IR-DF3 scenarios, the *Attractiveness* of a document is related to hypertext-based and combinative features, while in IR-DF2 and IR-DF4 scenarios this definition is concentrated to a mixture of some hypertext-related attributes.

It must be noticed that all of the above-mentioned scenarios have a preprocessing phase on the utilized features of dotIR. In this process, employed original features are normalized by the Min-Max normalization method, and thereafter, the Dirichlet prior smoothing algorithm (Manning, Raghavan, & Schütze, 2008) is applied to the outcome. Table 3 provides a comparison of the above-mentioned scenarios for the calculation of click-through features. As it is observable, all of the scenarios utilize not more than 25 percent of dotIR features and provide a very compact representation of this dataset with only eight generated features. However, based on tentative results, they can outperform baseline ranking mechanisms. Another interesting observation is that content-based and link-based features of the dotIR dataset play a vital role in the click-through feature generation scenarios. However, content-based features have a little more influence on the formation of click-through features.

Table 3. Comparison of Click-through Feature Generation Scenarios Used on dotIR dataset

Scenario ID	% Original Features Used	# Utilized Content-based Features	# Utilized Link-based Features	# Utilized Mixture Features	#Generated Click-through Features
IR-DF1	$(14/56) \approx \%25$	6	5	3	8
IR-DF2	$(13/56) \approx \%23.2$	6	6	1	8
IR-DF3	$(13/56) \approx \%23.2$	6	4	3	8
IR-DF4	$(12/56) \approx \%21.4$	6	5	1	8

3. Evaluation Criteria

In the information retrieval literature, different measures are suggested for the comparison of ranking algorithms. In this paper, the following evaluation criteria are used:

- *P@n*: indicates the ratio of relevant documents in a list of *n* retrieved documents (Manning, Raghavan, & Schütze, 2008). The main aim of this metric is to calculate the precision of retrieval systems from the users' perspective. As users visit only the top part of the ranked list, most evaluation measures just consider *n* top documents. In this way, *P@n* is defined as:

$$P@n = \frac{\#\ relevant\ docs\ in\ top\ n\ results}{n} \tag{8}$$

- *MAP*: for a single query, *Average Precision* is defined as the average of the *P@n* values for all relevant documents.

$$AVG(q) = \frac{\sum_{j=1}^{|D_q|}(r(j) \times P@j)}{|R_q|} \tag{9}$$

In Equation 9, $r(j)$ is the relevance score assigned to the document d_j (1 when relevant to query q, and 0 otherwise); D_q is the set of retrieved documents and R_q is the set of relevant documents for the query q. Then, *MAP* calculates the mean of average precisions of each query available in the query set as shown in Equation 10:

$$MAP = \frac{\sum_{q=1}^{|Q|} AVG(q)}{|Q|} \tag{10}$$

- *NDCG@n:* evaluation criteria such as *P@n* and *MAP* consider only the binary degrees of relevance in the evaluation of pairs of queries and documents. Therefore, the quality of their analysis may not be precise or satisfactory. Assuming different levels of relevance degree for data items, *NDCG@n* could be calculated as:

$$NDCG@n = 2^{r_1} - 1 + \sum_{j=2}^{n} \frac{2^{r_j} - 1}{\log(1 + j)} \tag{11}$$

In the above equation, r_j stands for the relevance degree of the j^{th} document in the ranked list.

Results

In the evaluation of the CF-Rank algorithm in the Persian Web area, different implementations of this algorithm are used. The best implementations are listed in Table 4. It should be mentioned that although some settings are the same, however, according to Table 2, scenarios used in the extraction of click-through features are different.

Table 4. Specifications of the best implementations of CF-Rank evaluated on dotIR dataset

Configuration ID	Click-through Feature Generation Scenario	Feature Weighting Criteria	Information Fusion Technique	α
CF.IR1	IR-DF1	MAP	Pessimistic OWA	0.1
CF.IR2	IR-DF2	MAP	Pessimistic OWA	0.1
CF.IR3	IR-DF3	MAP	Pessimistic OWA	0.1
CF.IR4	IR-DF4	MAP	Pessimistic OWA	0.1

Evaluation results of different implementations of the CF-Rank algorithm on the dotIR dataset based on the precision criterion is reported in Table 5. As it could be observed, the CF.IR2 and CF.IR4 scenarios are slightly better than other implementations of the CF-Rank.

Table 5. Evaluation results of the CF-Rank on the dotIR dataset based on the precision criterion

Configuration ID	P@1	P@2	P@3	P@4	P@5	P@6	P@7	P@8	P@9	P@10
CF.IR1	0.52	0.55	0.5733	0.575	0.584	0.5567	0.5686	0.5725	0.5689	0.566
CF.IR2	0.52	0.55	0.5733	0.575	0.584	0.5567	0.5686	0.5725	0.5711	0.568
CF.IR3	0.52	0.55	0.5733	0.575	0.584	0.5567	0.5686	0.5725	0.5689	0.566
CF.IR4	0.52	0.55	0.5733	0.575	0.584	0.5567	0.5686	0.5725	0.5711	0.568

A comparison of the proposed algorithm with baseline ranking algorithms according to the precision criterion is presented in Figure 2. Based on the *P@n* criteria, the CF-Rank is substantially dominated the baseline ranking algorithms. For instance in *P@1*, which is related to the precision of the top-ranked result item, the CF-Rank, has shown an improvement of 30 percent in comparison with the best baseline ranking algorithm. This value is about 31 percent in the *P@2*.

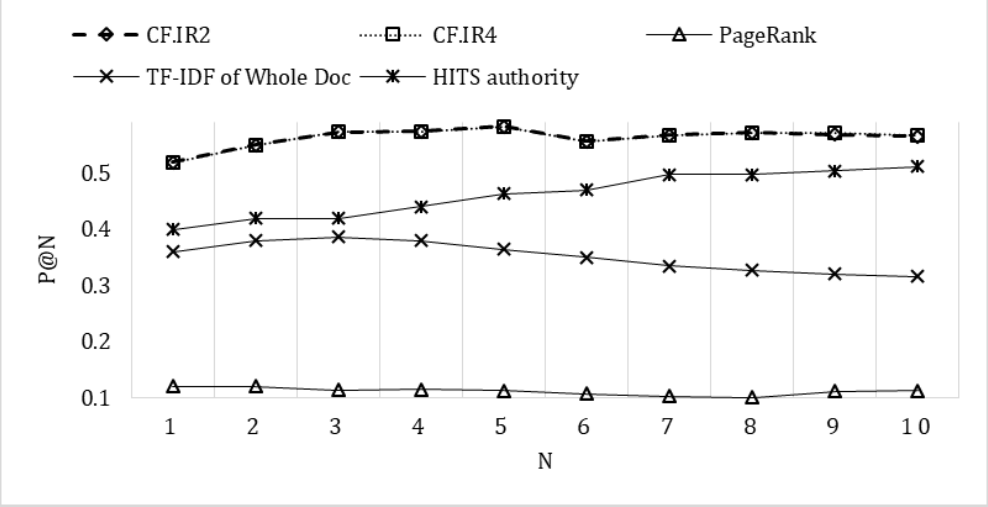

Figure 2. Comparison of the CF-Rank with baseline ranking algorithms according to the precision criterion

Figure 3 depicts the assessment results of the CF-Rank in comparison to the baseline ranking techniques on the *Mean Average Precision (MAP)* measure. According to the *MAP* measure, the CF-Rank has outperformed the best baseline algorithm by the factor of 16.5 percent.

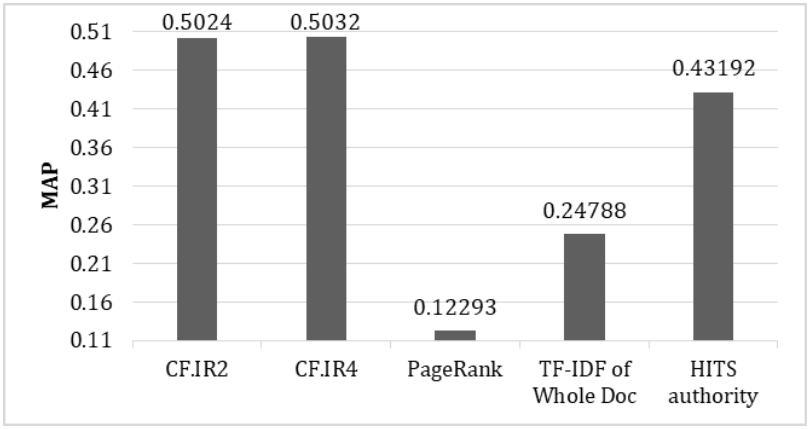

Figure 3. Comparison of the CF-Rank with baseline ranking algorithms according to the MAP measure

In order to have a comprehensive comparison of the CF-Rank algorithm, its evaluation is repeated with NDCG measure. The performance of different implementations of the CF-Rank based on the NDCG measure is reported in Table 6.

Table 6. Evaluation results of the CF-Rank on the dotIR dataset based on the NDCG criterion

Configuration ID	NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5	NDCG@6	NDCG@7	NDCG@8	NDCG@9	NDCG@10
CF.IR1	0.52	0.55	0.5668	0.5689	0.5751	0.5599	0.5665	0.5689	0.5671	0.5655
CF.IR2	0.52	0.55	0.5668	0.5689	0.5751	0.5599	0.5665	0.5689	0.5684	0.5667
CF.IR3	0.52	0.55	0.5668	0.5689	0.5751	0.5599	0.5665	0.5689	0.566	0.5655
CF.IR4	0.52	0.55	0.5668	0.5689	0.5751	0.5599	0.5665	0.5689	0.5667	0.5667

In Figure 4, a comparison of the CF-Rank algorithm with baseline ranking algorithms according to the *NDCG* criterion is represented.

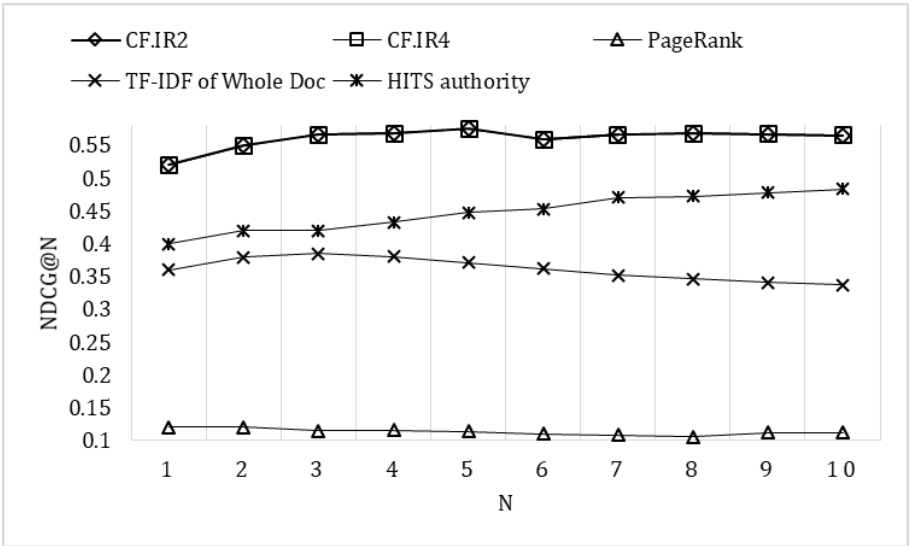

Figure 4. Comparison of the CF-Rank with baseline ranking algorithms according to the NDCG criterion

Figure 5 presents the assessment results of the CF-Rank in compare with baseline ranking techniques on the *MeanNDCG* measure.

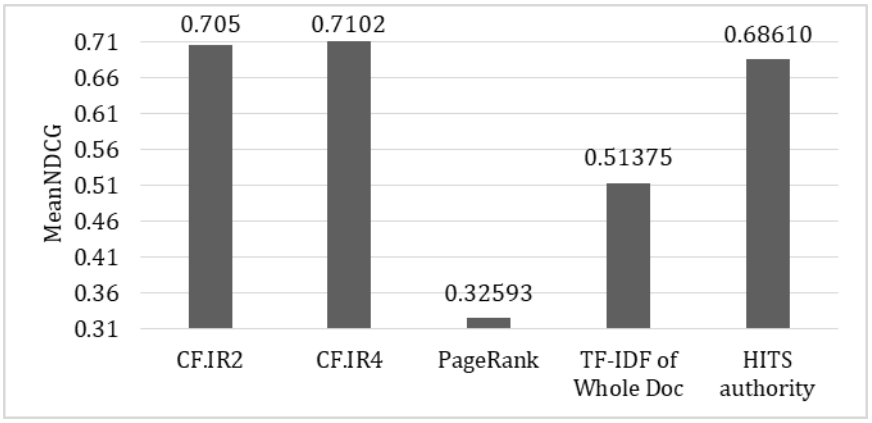

Figure 5. CF-Rank in comparison with baseline ranking algorithms based on the MeanNDCG measure

Evaluation of the CF-Rank by the use of *NDCG@n* criteria confirms the previous observation. For example, the *NDCG@1* value of the CF-Rank is 30 percent higher the same value for the preeminent baseline algorithm. In addition, CF-Rank is better than baseline rankings according to the *MeanNDCG* criterion.

As a brief, based on the tentative results, the main observations could be listed as:

- CF-Rank is a general ranking framework, which should be tailored on a given dataset by finding appropriate scenarios for the generation of the click-through features. A major contribution of this paper is the identification of some appropriate click-through feature generation scenarios for the Persian content based on the characteristics of the dotIR dataset.
- According to different evaluation criteria, CF-Rank has outperformed baseline ranking algorithms.
- The best configurations of the CF-Rank in the dotIR dataset have used the *Mean Average Precision (MAP)* criterion in the prioritization of the click-through features. It has been found to be more useful than the *OneR* and *MeanNDCG* factors. Moreover, the Pessimistic OWA as the aggregation operator has been better than Optimistic OWA in the dotIR dataset and has led to the most powerful configurations of the CF-Rank.
- Successful implementations of the CF-Rank use a very limited number (not more than 25%) of base features and generate a compact representation of the primitive dataset, which includes only eight features.
- The highest improvement of the CF-Rank is on the top-ranked results, which are usually more noticed by the users (AdvancedWebRanking, 2019).
- Based on the statistics of Table 3, both content-based and hyperlink-based primitive features are important in the click-through feature generation in the dotIR dataset.

Conclusion

Learning to rank as a new paradigm for effective Web information retrieval applies machine learning algorithms in the ranking problem. However, L2R algorithms have encountered some major challenges in real-world applications. First, they have to deal with huge benchmark datasets including a large number of various features related to queries and documents. The preparation of such datasets is a very difficult and expensive task. Although it is known that applying users' click-through data in the ranking process is useful (Dou, Song, Yuan, & Wen, 2008) (Cen, et al., 2009) (Macdonald, Santos, & Ounis, 2012), though most of the available L2R datasets do not include such data. CF-Rank as one of the recently proposed L2R algorithms aims to handle the above-mentioned problems (Keyhanipour, Moshiri, & Rahgozar, 2015). Its noticeable performance in well-known datasets was our motivation to customize this algorithm for the Persian Web content and apply it in the dotIR dataset. In this regard, here we have proposed some effective click-through feature generation scenarios in the dotIR dataset. Thereafter, base classifiers are constructed in each category of the generated click-through features. At the final step, these classifiers are fused using the OWA information fusion operators. Experimental results show an evident improvement in comparison with baseline rakings. According to the precision criterion, the CF-Rank has achieved an improvement of 30 percent against the best baseline ranking algorithm at the top of the ranked lists, which are usually more attractive for most Web users. This observation is confirmed based on other evaluation criteria such as *MAP*, *NDCG*, and *MeanNDCG*.

There are different ways to extend this research work. Finding other classifier fusion techniques is very important. Besides, the investigation of other click-through feature generation techniques based on the characteristics of the Persian Web seems to be promising. In addition, L2R datasets deal with bias of human judgments about the relevance level of query-document pairs. Studying the effect of such a bias and proposing techniques to handle such a problem is a vital task. In this way, utilization bias identification and handling algorithms (Baeza-Yates, 2018) may be beneficiary.

Appendix A: Feature set of dotIR benchmark dataset (Darrudi et al., 2009)

Feature ID	Feature Name	Feature Type
F1	Term frequency (TF) of body	Content-based
F2	TF of anchor	
F3	TF of title	
F4	TF of URL	
F5	TF of whole document	
F6	Inverse document frequency (IDF) of body	
F7	IDF of anchor	
F8	IDF of title	
F9	IDF of URL	
F10	IDF of whole document	
F11	TF×IDF of body	
F12	TF×IDF of anchor	
F13	TF×IDF of title	
F14	TF×IDF of URL	
F15	TF×IDF of whole document	
F16	Document length (DL) of body	
F17	DL of anchor	
F18	DL of title	
F19	DL of URL	
F20	DL of whole document	
F21	BM25 of body	
F22	BM25 of anchor	
F23	BM25 of title	
F24	BM25 of URL	
F25	BM25 of whole document	
F26	LMIR.ABS of body	
F27	LMIR.ABS of anchor	
F28	LMIR.ABS of title	
F29	LMIR.ABS of URL	
F30	LMIR.ABS of whole document	
F31	LMIR.DIR of body	
F32	LMIR.DIR of anchor	
F33	LMIR.DIR of title	
F34	LMIR.DIR of URL	
F35	LMIR.DIR of whole document	
F36	LMIR.JM of body	
F37	LMIR.JM of anchor	
F38	LMIR.JM of title	
F39	LMIR.JM of URL	
F40	LMIR.JM of whole document	
F41	Sitemap based term propagation	Mixture of Content-based and Hyperlink-based
F42	Sitemap based score propagation	
F43	Hyperlink base score propagation weighted in-link	
F44	Hyperlink base score propagation weighted out-link	
F45	Hyperlink base score propagation uniform out-link	
F46	Hyperlink base feature propagation weighted in-link	
F47	Hyperlink base feature propagation weighted out-link	
F48	Hyperlink base feature propagation uniform out-link	
F49	HITS authority	Hyperlink-based
F50	HITS hub	
F51	PageRank	
F52	In-link number	
F53	Out-link number	
F54	Number of slash in URL	
F55	Length of URL	
F56	Number of child page	

References

AdvancedWebRanking. (2019, July). *Google Organic CTR History, Fresh CTR averages pulled monthly from millions of keywords.* Retrieved May 15, 2019, from https://www.advancedwebranking.com/ctrstudy/

Baeza-Yates, R. (2018). Bias on the Web. *Communications of the ACM, 61*(6), 54-61.

Busa-Fekete, R., Kégl, B., Éltető, T., & Szarvas, G. (2013). Tune and mix: learning to rank using ensembles of calibrated multi-class classifiers. *Machine Learning, 93*(2-3), 261–292.

Cen, R., Liu, Y., Zhang, M., Zhou, B., Ru, L., & Ma, S. (2009). Exploring relevance for clicks. *The 18th ACM conference on Information and knowledge management* (pp. 1847-1850). ACM.

Chapelle, O., & Chang, Y. (2011). Yahoo! Learning to Rank Challenge Overview. *The Learning to Rank Challenge*, (pp. 1-24).

Darrudi, E., Hashemi, H. B., AleAhmad, A., Zare Bidoki, A., Habibian, A., Mahdikhani, F., & Rahgozar, M. (2009). *dotIR collection for Persian web retrieval.* University of Tehran. Retrieved May 15, 2019, from http://dbrg.ut.ac.ir/webir/files/Papers/WebIR.pdf

Derhami, V., Khodadadian, E., Ghasemzadeh, M., & Zareh Bidoki, A. (2013). Applying reinforcement learning for web pages ranking algorithms. *Applied Soft Computing*, 1686–1692.

Derhami, V., Paksima, J., & Khajeh, H. (2019). RRLUFF: Ranking function based on reinforcement learning using user feedback and web document features. *AI and Data Mining*. Retrieved May 15, 2019, from http://jad.shahroodut.ac.ir/article_1446.html

Dou, Z., Song, R., Yuan, X., & Wen, J.-R. (2008). Are click-through data adequate for learning web search rankings? *17th ACM Conference on Information and Knowledge Management Conference* (pp. 73-82). ACM.

Filev, D., & Yager, R. R. (1994). Learning OWA operator weights from data. *The Third IEEE Conference on Fuzzy Systems* (pp. 468-473). IEEE.

Hashemi, H. B., Yazdani, N., Shakery, A., & Naeini, M. P. (2010). Application of ensemble models in web ranking. *The 5th International Symposium on Telecommunications*, (pp. 726-731).

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning, 11*, 63-91.

Joachims, T. (2002). Optimizing search engine using clickthrough data. *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 132-142). ACM.

Keyhanipour, A., Moshiri, B., & Rahgozar, M. (2015). CF-Rank: Learning to rank by classifier fusion on click-through data. *Expert Systems with Applications, 42*, 8597-8608.

Khodadadian, E., Ghasemzadeh, M., Derhami, V., & Mirsoleimani, A. (2012). A novel ranking algorithm based on reinforcement learning. *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*, (pp. 546-551).

Liu, T.-Y. (2011). *Learning to rank for information retrieval.* Springer-Verlag.

Macdonald, C., Santos, R. L., & Ounis, I. (2012). On the usefulness of query features for learning to rank. *The 21st ACM International Conference on Information and Knowledge Management* (pp. 2559-2562). ACM.

Makvana, K., Patel, J., Shah, P., & Thakkar, A. (2018). Comprehensive analysis of personalized web search engines through information retrieval feedback system and user profiling. *The International Conference on Advanced Informatics for Computing Research*, (pp. 155-164).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *An introduction to information retrieval.* Cambridge, England: Cambridge University Press.

Mitchell, T. (1997). *Machine Learning.* McGraw Hill.

Qin, T., & Liu, T.-Y. (2013). Introducing LETOR 4.0 datasets. *arXiv, abs/1306.2597*. Retrieved May 15, 2019, from http://arxiv.org/abs/1306.2597

Qin, T., Liu, T.-Y., Xu, J., & Li, H. (2007). LETOR: Benchmark dataset for research on learning to rank for information retrieval. *The ACM SIGIR 2007 Workshop on Learning to Rank for Information Retrieval* (pp. 3-10). ACM.

W3Techs. (2019, July). *Usage of content languages for websites.* W3Techs. Retrieved May 15, 2019, from https://w3techs.com/technologies/overview/content_language/all

Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetic, 18*, 183-190.

Bibliographic information of this paper for citing:

Keyhanipour, Amir Hosein (2019). Effective learning to rank Persian web content. *Journal of Information Technology Management*, 11(4), 92-109.