A Novel Ranking Algorithm Based on Reinforcement Learning

Elahe Khodadadian[1], Mohammad Ghasemzadeh[2], and Vali Derhami[3]
Department of Electrical and Computer Engineering
Yazd University
Yazd, Iran
[1]khodadadi@stu.yazduni.ac.ir
{[2]m.ghasemzadeh, [3]vderhami}@ yazduni.ac.ir

S.Ali Mirsoleimani
Department of Electrical and Computer Engineering
Shiraz University
Shiraz, Iran
ali.mirsoleimani@gmail.com

Abstract— **People are very interested in finding most related web pages when they are looking for some information about a topic in the web. This is the responsibility of the search engine to use effective ranking methods to find the most related web pages for a given query. In this paper, we propose a novel ranking method which is based on connectivity and also benefits from the Reinforcement Learning (RL) concepts. RL problems are structured around estimating value functions. In our algorithm, each web page is considered as a state and its score is as value function of the state. So, the key elements in our method are agent, value function and considered interstate transition rewards. Also we inspected a hybrid ranking algorithm which combined results of several basic ranking algorithms. The proposed algorithms are evaluated by using well known benchmark data sets and they are analyzed according to concerning criteria. Experimental results show that applying reinforcement learning method leads to considerable improvements and the hybrid algorithm can outperform all other basic ranking algorithms**.

Keywords-ranking method; Reinforcement Learning; value function; reward; agent

I. INTRODUCTION

Nowadays World Wide Web (WWW) is considered to be the best source of information. The main reasons for its importance are; easy access, low-cost and being responsive to users' requests in the shortest time. Search engines are the predominant tools for finding and get access to the contents on the web. Whenever users seek information, enter their query in search engine and it searches through web pages and return a list of relevant ones.

The search engines first, retrieve web pages by a crawler (sometimes also known as a spider). In fact, a crawler visits a web page, and follows all the links provided in that page. This operation leads to constructing a web graph (A web graph consists of nodes and edges, where nodes stand for web pages and edges show the links which are available from each page to other pages). After collecting web pages, content of each page are analyzed to determine how it should be indexed (e.g. words are extracted from the titles, headings, or special fields). The purpose of an index is to allow information to be found as quickly as possible. Ranking is the final stage. In this stage millions of web pages were recorded in the previous stage are sifted to find matching cases for a specified query and sorting them based on the users' requests or preferences. Due to the huge size of the web, it is very common that a large number of relevant results are returned for a given query. Moreover, studies have shown that users don't have the time and the patience to go through all of them to find the ones which they are interested in. They often consider the top 10 or 20 results. Therefore, a better and efficient ranking algorithm is required. This will enable search engines to present the best related pages to users in response to their queries.

In this paper, we propose a ranking algorithm inspired from reinforcement learning concepts [1] and it called RL_Rank. Reinforcement learning is a branch of machine learning that deals with optimal sequential decision making. The key elements in an RL problem are reward and value function [1]. In our method, we consider a user as an agent and each web page as a state. The score of each page is also defined as the value function of the state. In this algorithm, we consider a random surfer (agent) starts from a random page (state). After visiting a web page, she selects the next page by clicking randomly on one of the links in that page. Giving reward in transition from each page to a next page corresponds to inverse of the out-degree (number of hyperlinks pointing to other pages from the current page) of the source page. The score of each page (value function of state) is the total amount of rewards the surfer can expect to accumulate during traveling through pages to reach that page.

Also we introduce a hybrid ranking algorithm which is based on benefiting the results of BM25 and RL_Rank algorithms. This idea applies from the concept of Meta search engines with the difference that Meta search engines merge the results of other search engines while here we merge the results of some ranking algorithms. Generally, we combine some ranking algorithms to achieve a better ranking algorithm and bring the higher quality pages to the top of the result list.

The remainder of this paper is organized as follow: Ranking algorithms are discussed in Section 2. Section 3 introduces our proposed ranking algorithms. Experimental analysis and their results are presented in Section 4. Finally, in section 5 we summarize our main contributions and discuss some possible further improvements on our proposed method.

II. RANKING ALGORITHMS

Ranking algorithms divided into two categories namely content-based and connectivity-based algorithms.

Content-based algorithms usually work based on matching words in documents. In otherwords, for each query the documents with the most similar content to the query will be selected as the more relevant. Examples of these algorithms are vector space [2], TF-IDF [3] and BM25 [4]. These algorithms are suitable for well structured environments such as digital libraries, rather than the web pages which usually include large number of unstructured contents.

Connectivity-based algorithms use links between web pages. Links carry information which can be used to evaluate the importance of pages and the relevancy of pages to the user query. These algorithms are divided into two major classes "query-independent" and "query-dependent". Instances of query-independent algorithms are PageRank [5], HostRank [6] and DistanceRank [7]. These algorithms use the entire web graph and compute the score of web pages offline, whereas query-dependent algorithms such as HITS [8] involve the construction of a query-specific graph.

This research is mostly concerned with connectivity algorithms that are offline. Among these algorithms, PageRank is at the center of our attention. Therefore, we describe PageRank which is the most widely used web pages ranking algorithms. Then we review BM25 algorithm which is used in our hybrid algorithm.

A. PageRank Algorithm

PageRank is a popular ranking algorithm used by Google search engine. PageRank models the users' browsing behaviors as a random surfer model. In this model, a user who surfs the web by randomly clicking links on the visited pages and sometimes jumps to another page at random. In this algorithm, fraction of time the surfer spends on a page is defined as the score of that page [9]. PageRank measures the importance of web pages as follows: if a page has many links to other pages, it can be concluded that many people are interested in that page and the page should be considered an important one. PageRank takes the backlinks (incoming links to a web page) into account and propagates the ranking through links: a page has a higher rank if the sum of the ranks of its backlinks is higher. The score of a page such as p ($S(p)$) can be approximated by the following recursive formula [5]:

$$S(p) = d \times \sum_{j \in B(p)} \frac{S(j)}{O(j)} + \frac{1-d}{n} \tag{1}$$

Where $S(p)$ and $S(j)$ show score of pages p and j respectively. d is the damping factor, n is the total number of pages. $B(p)$ is the set of pages pointed to page p and $O(j)$ the out-degree of the page j.

The presence of the damping factor is necessary, because the web graph is not a strongly connected graph (SCG), so it used to guarantee the convergence of PageRank and remove the effects of sink pages (pages with no out-link).

B. BM25 Algorithm

BM25 is a ranking algorithm based on content. It is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document. Given a query Q, containing keywords $q_1,...,q_n$, the BM25 score of a document D is [4]:

$$S(D,Q) = \sum_{i=1}^{n} IDF(q_i) \frac{f(q_i,D)(k_1+1)}{f(q_i,D)k_1(1-b+b\frac{|D|}{avgdl})} \tag{2}$$

where $f(q_i,D)$ is q_i's term frequency in the document D, $|D|$ is the length of the document D in words, and $avgdl$ is the average document length in the text collection from which documents are drawn. k_1 and b are free parameters, usually chosen, in absence of an advanced optimization, as $k_1 \in [1.2, 2.0]$ and b = 0.75. $IDF(q_i)$ is the IDF (inverse document frequency) weight of the query term q_i. It is usually computed as:

$$IDF(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \tag{3}$$

where N is the total number of documents in the collection and $n(q_i)$ is the number of documents containing q_i.

III. PROPOSED ALGORITHM

RL_Rank algorithm inspired from reinforcement learning concepts. So in this section, we review reinforcement learning concepts. Afterwards, RL_Rank and hybrid algorithms are described in the details.

A. Reinforcement Learning

Reinforcement learning as one of the machine learning techniques is learning by interactive in dynamic environment. Also, it is a powerful tool in determining effective states in states space. In an RL problem, the learner is called the agent who learns through its interaction with the environment and it acquires knowledge through reward or punishments of an action undertaken [1].

In an agent-based system with reinforcement learning, at each time step t, the agent is involved with a state called current state and selects an action from a set of possible actions. The policy, denoted by $\pi(s,a)$, is the probability of selecting action a when agent is concerned with state s. Afterwards, the environment goes to next state (s_{t+1}), and the agent receives reinforcement signal $r_{t+1} = (r(s_t,a_t))$ that is called a reward [1]. Reinforcement signal is a scalar signal and it indicates the intrinsic desirability of the action. Then, agent updates value function of the state. The state value function under policy π is expected value of the sum of received discounted rewards, defined as follows [1]:

$$V^{\pi} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^{k}\, r_{t+k+1} \,\middle|\, S_t = s\right\} \quad 0 \le \gamma \le 1 \tag{4}$$

where γ is a discount factor that determines the present value of the future rewards that can be achieved over time. $E_{\pi}\{0\}$ denotes the expected value and r_{t+k+1} is a reward that agent receives during transition between states.

B. RL_Rank Algorithm as Reinforcement Learning

In RL_Rank algorithm, we consider an agent as a surfer that moves between pages, randomly. Each web page is considered as a state. In each page (state), the surfer (agent) clicks on one of the available links in that page with a uniform probability, and goes to the next state. Therefore, an agent's action is to click on one of the links randomly with a uniform probability. The reward is given when a transition from a current state (j) to another state (p) defined by:

$$r_{jp} = \frac{1}{O(j)} \tag{5}$$

where $O(j)$ is the out-degree of page j. As the denominator, it means that the page with less out-degree shares more value between its children. Thus, each child receives more rewards.

We define the score of each page p to be the expected value of sum of discounted rewards that agent accumulates during traveling through pages to reach page p. Then agent adds the received reward r_{jp} to the discounted accumulated rewards. Therefore, score of page p is probability of reaching it from other pages multiplied by sum of the transition reward and discounted accumulated rewards. The value function formula that is considered as score of page is as follows:

$$R_{t+1}(p) = \sum_{j \in B(p)} \left(\left(\frac{prob_{t+1}(j)}{O(j)}\right) \times (r_{jp} + \gamma R_t(j)) \right) \tag{6}$$

where $S_{t+1}(p)$ is score of page p in time $t+1$ and S_t (j) shows the score page j in time t, $B(p)$ is the set of pages that point to page p, $prob_{t+1}(j)$ is the presence probability of the agent at page j in time t+1, $O(j)$ is the out-degree of page j and r_{jp} the reward for transition from page j to p defined by "(5)". Therefore, the score of page p depends on the out-degree and score of the pages pointing to page p.

The value of $prob_{t+1}(j)/O(j)$ is the probability of reaching page p from page j. It is equal to presence probability of the agent at page j multiplied by selection probability of page p when agent is in page j.Since the agent selects one of the links by uniform probability distribution; the selection probability of page p from j is equal inverse of the out-degree of page j. As we mentioned before, the fraction of time spent on a page by user defined as PageRank of that page [9]. So, presence probability of page j $(prob_{t+1}(j))$ is calculated as "(1)".

As "(6)" shows RL_Rank is computed recursively like PageRank. The process iterates to converge. Finally, we will have the RL_Rank vector and pages are sorted in the descending order.

C. Hybrid Algorithm

Each of content-based and connectivity-based algorithms behaves satisfactory for specific environments, but in general they are not efficient when used separately. It seems that we can obtain better performance and overcome their drawbacks by using a combination of them. It is obvious that the properties of the page play an important role in the quality of the page. Therefore, weighted combination of different ranking algorithms (such as PageRank, BM25, TF_IDF, HITS and etc) as properties of page can be effective [10]. The score of the page by hybrid algorithm (weighted combination of some algorithms) is calculated as follows:

$$S(p) = \sum_{i=1}^{n} w_i S_i(p) \tag{7}$$

Where $S(p)$ and $S_i(p)$ are scores of page p by hybrid algorithm and ith algorithm, respectively. w_i is weight of ith algorithm. i varies from 1 to n (n is the number of algorithms which are used in hybrid algorithm).

We combine results of RL_Rank and BM25 algorithms and we have assigned a fixed weight to any of them by Borda method [11, 12]. In fact, the normalized accuracy of each algorithm considered as their weight.

IV. EVALUATION AND EXPERIMENTAL RESULTS

In order to assess the proposed algorithm, we compared it with PageRank algorithm on well known benchmark in the standard criteria. So, we first describe evaluation measures, benchmark data sets and then report the experimental results.

A. Evaluation Measures

For proposed of evaluation, we use a number of evaluation measures commonly used in information retrieval (IR), namely, Precision at n (P@n) [13], Mean Average Precision (MAP) [13] and Normalized Discount Cumulative Gain (NDCG) [14]. Their definitions are as follows:

- *Precision at n (P@n)*: this criterion indicates the ratio of top relevant documents to total number of documents (n) in presented results. In fact, it indicates system accuracy:

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

- *Mean Average Precision (MAP)*: Average Precision (AP) corresponds to the average of P@n values for all relevant documents of a given query and is computed by following equation:

$$AP = \frac{\sum_{i=1}^{n} (P@i.rel(i))}{\#\,total\ relevant\ docs\ \ for\ one\ query} \tag{9}$$

where n is the number of retrieved documents, and *rel(i)* is a binary function on the relevance of the ith document. If ith document is a relevant page, *rel(i)* will be equal to 1, otherwise it is 0. Finally, MAP is obtained by computing the average of AP values over the set of queries.

- *Normalized Discount cumulative Gain (NDCG)*: The above mentioned criteria (P@n and MAP) can only provide binary judgment: "relevant" or "irrelevant", whereas NDCG can perform multiple levels of relevance judgments. For a single query, the NDCG value of a ranking list at position n is computed as:

$$NDCG@n = \sum_{i=1}^{n} \frac{2^{r_j}}{\log(1+i)} \tag{10}$$

where r_j is the rating of the jth document in the ranking list.

B. Experimental Results

For the purpose of evaluation, we used *LETOR* benchmark data set [15] which is derived from the existing English test collection. Moreover, we evaluate our solutions by a newly constructed web test collection on Iran web, *dotIR* benchmark [16], which is released by Iran Telecommunication Research Center (ITRC) [17]. These benchmark data sets are constructed for research on information retrieval and are publicly available. *LETOR* constructed based on the existing data sets and query sets, namely, the "Gov" and OHSUMED corpora. We use 50 TREC-2003 queries on the "Gov" corpus. There are totally 1,053,110 pages in *TREC-2003* and *dotIR* data set contains 997,462 web pages, 50 queries. The both data sets consist of the contents of the web pages, queries, and human judgments on the retrieved documents with respect to the queries.

In the experiments, the factor γ in RL_Rank was set to 0.5 and the damping factor in PageRank was set to 0.85. "Fig. 1", "Fig. 2" and "Fig. 3" summarized results of experiment on dotIR benchmark. This results show that the performance of RL_Rank is better than PageRank algorithm in terms of all measures. "Fig. 3" illustrates 26% increases in ranking quality of RL_Rank in comparison to PageRank algorithm.

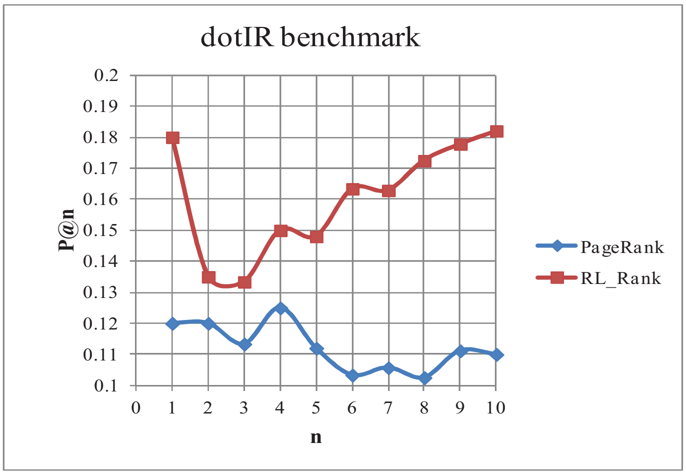

Figure 1. Comparison of RL_Rank with PageRank in the P@n.

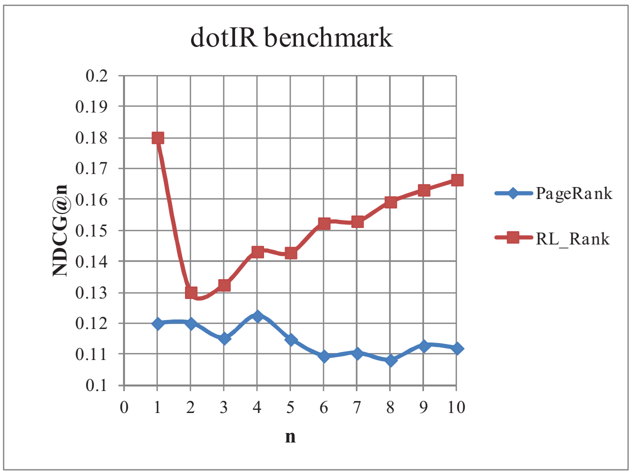

Figure 2. Comparison of RL_Rank with PageRank in the NDCG@n.

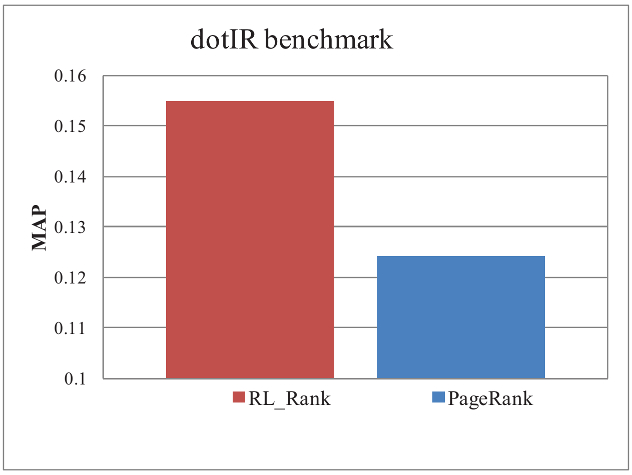

Figure 3. Comparison of RL_Rank with PageRank in the MAP.

Graphical analyses of results on TREC-2003 benchmark data set are depicted in "Fig. 4", "Fig. 5" and "Fig. 6". The results on TREC-2003 benchmark data set show the obtained values for RL_Rank are higher than PageRank algorithm in terms of all measures. Also, we observed about 7% improvement.

A close look at the results indicates that RL_Rank is a suitable algorithm for ranking of the web pages. Evaluation of RL_Rank algorithm signifies that it makes larger improvements on dotIR data set in compare to TREC-2003 data set. It should be noted that RL_Rank and PageRank are two connectivity-based ranking algorithms and they are influenced by connectivity features of web pages; especially out-degree of pages. Hence, we inspected statistical characteristics of two graph (dotIR and TREC-2003) and we observed that these two graphs of data sets have great different, e.g. Mean of out-degree web pages in dotIR is about 40 whereas this feature in TREC-2003 is about 10. This difference reflects dotIR graph is denser than TREC-2003 graph. Hence, it can be concluded that RL_Rank has high performance in denser web graphs.

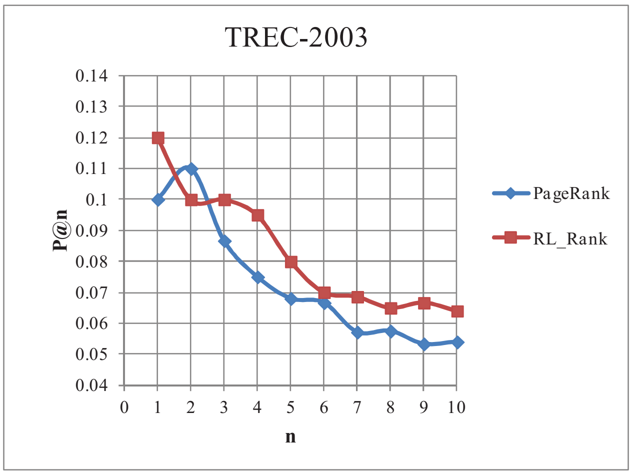

Figure 4. Comparison of RL_Rank with PageRank in the P@n.

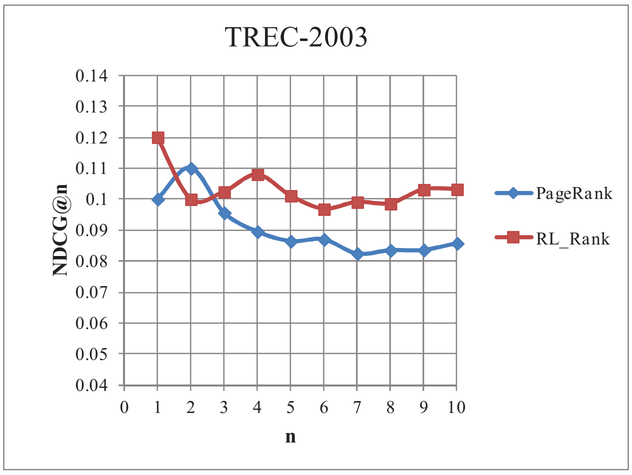

Figure 5. Comparison of RL_Rank with PageRank in the NDCG@n.

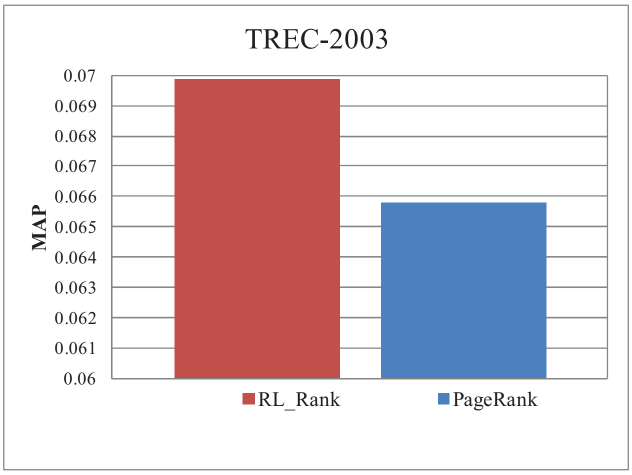

Figure 6. Comparison of RL_Rank with PageRank in the MAP.

The second part of experiments is about evaluation of hybrid algorithm. Combination of RL_Rank and BM25 algorithms was called CRLBM. For demonstrating the superiority of RL_Rank to PageRank, we also combined PageRank with BM25 and it was called CPRBM. In hybrid algorithm, weights for all participating algorithms are assigned. Therefore, by using this operator all of the algorithms will have chance to affect the final aggregation value. Therefore a fixed weight to any of them is assigned in our experiments, weight of RL_Rank and PageRank in "(7)" were set to 0.15479 and 0.143 respectively. Weight of BM25 in CRLBM and CPRBM was set to 0.84521 and 0.857, respectively.

"Fig. 7" and "Fig. 8" summarized results of evaluation both hybrid algorithms on dotIR benchmark data set in terms of P@n and NDCG@n measures. This results show that both hybrid algorithms are much better than all other basic ranking algorithms. Also, the combination of BM25 with RL_Rank (CRLBM) outperforms combination with PageRank (CPRBM). Therefore, we can conclude RL_Rank algorithm generally is superior to PageRank.

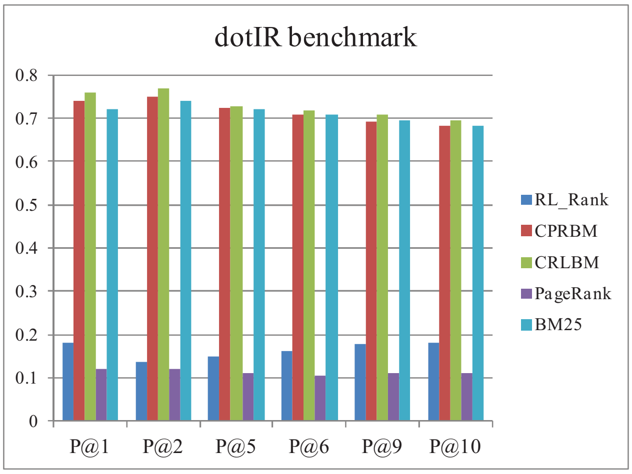

Figure 7. Evaluation of hybrid algorithms in the P@n.

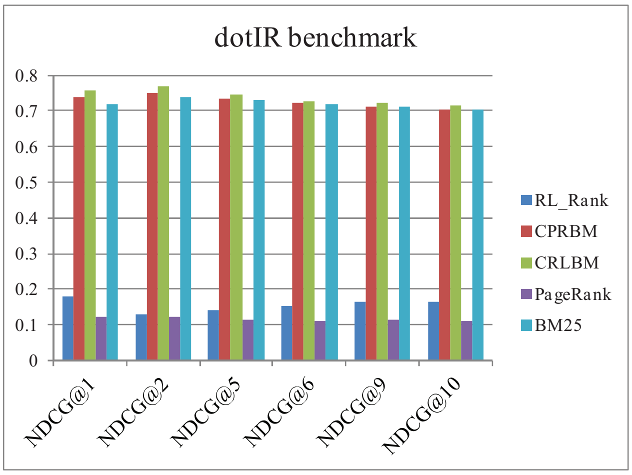

Figure 8. Evaluation of hybrid algorithms in the NDCG@n.

V. CONCLUSION

In this paper, using the reinforcement learning concept, we proposed RL_Rank algorithm which is a novel connectivity-based algorithm for ranking web pages. This algorithm considers rank determination of a page as an RL problem where the reward for transition from current page to the next page is proportional to the inverse of the out-degree of the current page. In fact, RL_Rank models the user who surfs the web by accumulating transition rewards to obtain rank of each page. Experimental results showed that it can present much better results than PageRank in standard criteria. Also we saw that RL_Rank behaves differently on different data sets (it makes larger improvements on dotIR data set in compare to TREC-2003 data set). Hence, it can be concluded that RL_Rank has high performance in denser web graphs. Also, we use RL_Rank algorithm in a hybrid algorithm. We observed that the hybrid algorithms are effective and their results are satisfying. Generally, it is obvious that RL_Rank outperforms PageRank algorithm. As future works we are going to define another rewards, combine another algorithms for higher efficiency and we will obtain weights in hybrid algorithm in a learning process.

ACKNOWLEDGMENT

This research was supported by Iran National Science Foundation (INSF). The authors would like to thank INSF for its support.

REFERENCES

[1] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An introduction*, Cambridge, MA :MIT Press, 1998, p. 322.

[2] G. Salton and C. Buckley. “Term-weighting approaches in automatic text retrieval”. *Journal of Information Processing and Management*, vol. 24, pp. 513-523, 1998.

[3] G. Salton. *The SMART Retrieval System-Experiments in Automatic Document Processing*. USA, Prentice-Hall, 1971, p.556.

[4] S.E. Robertson and S. Walker. “Some Simple Effective Approximations to the 2-poisson Model for Probabilistic Weighted Retrieval.” In *Proc. of SIGIR conference on Research and development in information retrieval*, 1994, pp. 232-241.

[5] L. Page, S. Brin, R. Motavni and T. Winogard. “The PageRank Citation Algorithm: Bringing Order to the Web”. *Tech.Rep, Stanford Digital Library Technologies Project,*1998.

[6] A.M. Zareh Bidoki and N. Yazdani. “DistanceRank: An Intelligent Ranking Algorithm for Web Pages”. *Journal of Information Processing and Management*, vol.44, pp. 877-892, 2008.

[7] G. R. Xue, Q. Yang, H. J. Zeng, Y. Yu and Z. Chen. “Exploiting the Hierarchical Structure for Link Analysis.” In *Proc. of SIGIR conference on Research and development in information retrieval*, 2005, pp. 186-193.

[8] J.M. Kleinberg. “Authoritative Sources in a Hyperlinked Environment”. *Journal of the ACM,* vol. 46, pp. 668-677, 1999.

[9] P. Boldi, M. Santini, and S. Vigna. “PageRank as a Function of the Damping Factor.” In *Proc. of World Wide Web Conference*, 2005, pp. 557-566.

[10] A.M. Zareh Bidoki, M. Azadnia, N. Yazdani, A.M. Keihanipour. “Adaptive Combinational Web Ranking Algorithm Using Coarse-grained and Fine-grained “. In *13th Conference on Iran computer Society*, 2009, p.12.

[11] J.C. Borda. M´emoire sur les ´elections au scrutin. In *Histoire de l'Academie Royale des Sciences*, 1781.

[12] D. Coppersmith, L. Fleischer, and A. Rudra. “Ordering by weighted number of wins gives a good ranking for weighted tournaments.” In *Proc. of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2006, pp. 776-782.

[13] A. Moffat and J. Zobel. “Rank-biased Precision for Measurement of Retrieval”. *Journal of ACM Transactions on Information Systems*, vol. 27, pp. 1-27, 2008.

[14] M. Gordon and P. Pathak. “Finding Information on the World Wide Web: The Retrieval Effectiveness of Search Engines”. *Journal of Information Processing and Management*, vol. 5*,* pp. 141-18, 1999.

[15] E. Darrudi, H. BaradaranHashemi, A. Aleahmad, A. Habibian, A.M. ZarehBidoki, A. Shakery and M. Rahgozar. “A Standard Web Test Collection for ir Domain”. *Tech.Rep*, *Iran Telecommunication Research Center*, 2009.

[16] T. Qin, T.Y. Liu, J. Xu and H. Li. “LETOR: A Benchmark Collection for Research on Learning to Rank for Information Retrieval”. *Journal of Information Retrieval*, vol. 13, pp. 346-374, 2010.

[17] E. Darrudi, H. BaradaranHashemi, A. Aleahmad, A. Habibian, A.M. ZarehBidoki, A. Shakery and M. Rahgozar. “dotIR benchmark collection.” Internet: http://ece.ut.ac.ir/dbrg/webir , [Feb.18, 2010]