A Novel Connectivity-based Reinforcement Learning Algorithm for Ranking Linked Data on the Social Semantic Web

Babak Farhadi
Department of Computer Engineering,
University of Tehran
Tehran, Iran

ABSTRACT
In social web zone, semantic web (web of data), in particular Linked Data (LD), has made it possible to link previously disconnected social datasets and services via common semantic definitions of terms (vocabularies, ontologies). In addition, semantic entities can be extracted from user-generated content items by web mining, Natural Language Processing (NLP) techniques and another Named Entity Recognition (NER) systems, and hence these content items can be connected together through common semantic definitions. In this regard, the social semantic web aims to overcome some of the essential restrictions through a combination of social web frameworks with semantic web standards, thereby creating a technology platform enabling semantically enhanced social spaces where communities and individuals participate in building distributed interoperable information. In this paper, a new ranking algorithm for LD on the social semantic web is offered, using Reinforcement Learning (RL) notions. The proposed algorithm is mapping of the connectivity-based PageRank algorithm, form web of documents to web of data with formulation of ranking as an RL problem. Experimental results demonstrate using RL concepts leads considerable improvements in PageRank raking algorithms.

Keywords
Social Semantic Web, Reinforcement Learning; Semantic Web; LOD Dataset.

1. INTRODUCTION
The social web has been widely pursued, allowing social participation and interaction through the creation of social spaces. Unfortunately, these social spaces are experiencing restrictions in terms of data reuse, interconnectivity, collaboration functionality, and usability. Many act as data silos which limit various opportunities for added value if they were easily connectable and could be connected to. Fortunately, critical semantic web technologies and standards are maturing in parallel with the social web. The social spaces of the social web can be combined with semantic web technologies to accelerate the next stage of the web and to accredit new applications in terms of knowledge discovery and data mining. This intersection of the semantic web and the social web is termed the "social semantic web" [1]. The social semantic web aims to overcome some of the fundamental limitations by a combination of social web frameworks with semantic web standards, thereby creating a technology platform enabling semantically enhanced social spaces where individuals and communities participate in building distributed interoperable information. It is a two-way street: the semantic web can help the social web and vice versa. The semantic web has suffered from a chicken-and-egg problem in the past, whereby it has been difficult to gather semantically rich data for semantic web applications to use; however users of the social web are creating semantically rich data every second. In the reverse direction, different heterogeneous platforms and social web clients can benefit from having interoperable semantic representations of social data to provide integrated views on this data and improved data exchange [2, 3, 4]. The social semantic web can be a platform for both personal and professional collaborative exchange with reusable community contributions. Via the use of semantic web data, search able and interpretable content is added to existing social web collaborative infrastructures, and intelligent use of this content can be made within (and between) these semantically enhanced social spaces allowing the vision of semantic data on the web to be realized to its greatest possible advantage. Some typical application areas for the social web are wikis, blogs/microblogs, and social networks, but can include any spaces where content is being created, annotated, and shared [1]. Each of these can be enhanced with machine-readable data to not only provide more functionality internally, but also to build an overall interconnected set of social spaces. This offers a number of possibilities in terms of increased automation and information diffusion that are not easily realizable with current social web applications. In this regards, the social semantic web can be used to bring together data from heterogeneous social websites through common representations and interlinkages [1]. In this paper, we propose a new algorithm for ranking Linking Open Data (LOD) cloud [5] based on Resource Description Framework (RDF) graphs on the social semantic web. In Figure 1, the linked datasets as of august 2014 are presented.

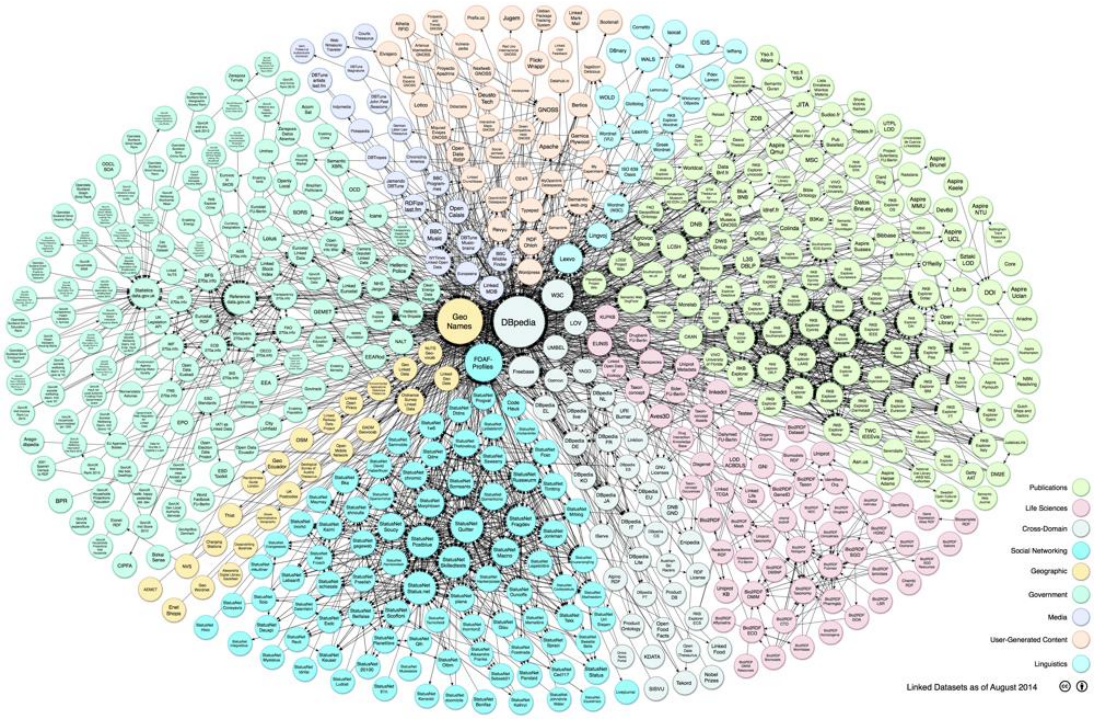

Fig 1: The linked datasets as of august 2014 [5].

The objective is specify the score of each dataset based on paths which can be reached to that dataset from other datasets as well as the output-degree (number of output links) of datasets in the traverse paths. Consider a random agent who transfers between datasets randomly. After meeting a dataset; the agent selects next dataset by choosing randomly one of the links in that dataset. The aforesaid process can be conversed as a Markov Decision Process (MDP) [6] problem where the target is policy evaluation. The foundation of RL in this problem are defined as follows: 1) *states*: datasets (on the

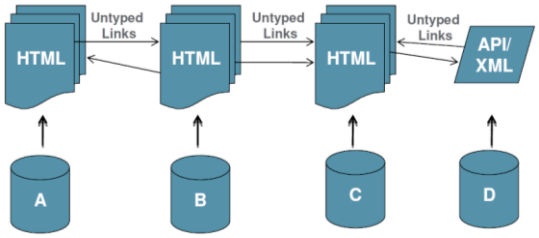

Fig 2: web of documents.

LOD), 2) *Actions*: output links on each dataset, 3) *Policy*: the agent selects the next dataset by choosing randomly one of the output links in current dataset. 4) *Reward*: reverse of the output-degree of the source dataset. 5) *Value function*: the total amount of rewards that agent can expect to cumulate during cruising through datasets to attain that dataset. Based on the above definitions, value function of each dataset is remarked as the score of the dataset. The proposed approach is called LD_Rank.

2. RANKING ALGORITHMS OF WEB OF DOCUMENTS

The principal elements of web of documents are following: 1) *Primary objects*: documents, 2) *Links*: between documents (or parts of them), 3) *Degree of structure in data*: fairly low, 4) *semantics of contents*: Implicit and 5) *Designed for*: human consumption (see Figure 2).

The ranking algorithms web of documents are divided in two main categories of content-based and connectivity-based algorithms. The content-based algorithms are based on matching words in documents. TF-IDF [7] and BM25F [8] are samples of these algorithms. Connectivity-based algorithms use links between pages on the web of documents. Generally, links carry information which can be used to evaluate the significance of pages and the relevance of pages to the user query. These algorithms are divided into two basic categories "query-independent" and "query-dependent". The most important instance of query-independent algorithms is PageRank [9]. Query-independent algorithms exert the complete web graph and compute the score of pages on the web of documents (offline), whereas query-dependent algorithms such as HITS [10] wrap the structure of a query-specific graph (online).

In here, we extend the connectivity-based PageRank as a mostly used and well-known algorithm, form web of documents to web of data with formulation of ranking as an RL problem.

2.1 PageRank Algorithm of Web of Documents

PageRank is a famous ranking algorithm used by Google search engine. It models the users' browsing behaviors as a random agent model. In this model, a user cruising on the web through randomly clicking links on the visited pages on the web of documents and sometimes jumps to other page at random. In this algorithm, deduction of time the agent spends on a page is determined as the score of that page [11].

About the PageRank algorithm of web of documents, the score of a web page such as i, $(wpr(wp_i))$ can be approximated through the following recursive formula [9]:

$$wpr(wp_i) = \left(\frac{1-d}{n} + d \times \sum_{wp_j \in p(wp_i)} \frac{wpr(wp_j)}{ol(wp_j)} \right) \quad (1)$$

where $wpr(wp_i)$ and $wpr(wp_j)$ show score of web pages i and j, respectively. d is the damping factor, n is the total number of pages on the web of documents and $p(i)$ and $ol(wp_j)$ are the set of pages pointed to page i and the output-degree (number of output links) of the page j, respectively. Since the web graph is not a strongly connected graph (SCG), the attendance of the damping factor is necessary, so damping factor used to warranty the convergence of PageRank and eliminate the effects of pages with no output-link.

3. MAPPING PAGERANK ALGORITHM FROM WEB OF DOCUMENTS TO WEB OF DATA

The fundamental elements of web of data are divided into five sections. 1) *Primary objects*: things (or description of things), 2) *Links*: between things, 3) *Degree of structure in data*: High (based on RDF data model), 4) *semantics of contents and links*: Explicit and 5) *Designed for*: Both machines and humans (see Figure 3).

The proposed approach about the mapping PageRank algorithm from web of documents to web of data is introduced in two situations of equally and unequally weighted links.

3.1 Equally Weighted Links Situation (regardless of the link type)

In the first proposed situation, Equation (2) shows ranking dataset ds_i on the web of data. Where d is damping factor and demonstrates the possibility of remaining in ds_i. The parameter of n is the total number of datasets. $p(ds_i)$ and $nl(ds_j, ds_i)$ illustrate the set of datasets pointed to dataset ds_i and the number of links from dataset ds_j to dataset ds_i respectively. $ol(ds_j)$ indicates the total number of output-links from ds_j.

$$dsr(ds_i) = \left(\frac{1-d}{n} + d \times \sum_{ds_j \in p(ds_i)} \frac{|nl(ds_j, ds_i)|}{|ol(ds_j)|} \times dsr(ds_j) \right) \quad (2)$$

3.2 Unequally Weighted Links Situation (considering the link type)

Equation (3) is suggested for ranking datasets where unequally weights are intended for the links on the web of data. Where d is damping factor and demonstrates the possibility of remaining in ds_i. The parameter of n is the total number of datasets. $p(ds_i)$ and $wnl(ds_j, ds_i)$ show the set of datasets pointed to dataset ds_i and the weighted links from dataset ds_j to dataset ds_i, respectively. $ol(ds_j)$ indicates the total number of output-links from ds_j.

$$dsr(ds_i) = \left(\frac{1-d}{n} + d \times \sum_{ds_j \in p(ds_i)} \frac{|wnl(ds_j, ds_i)|}{|ol(ds_j)|} \times dsr(ds_j) \right) (3)$$

so that $|nl_{lt}(ds_j, ds_i)|$ shows the number of links with type lt from dataset ds_j to dataset ds_i. lw illustrates link weight of type lt.

$$wnl(ds_j, ds_i) = \sum_{lt \in (citation\ attributes)} lw_{lt} \left| nl_{lt}(ds_j, ds_i) \right| \quad (4)$$

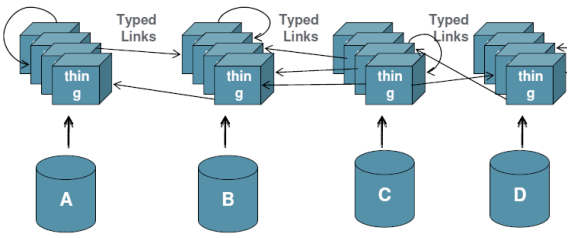

Fig 3: web of data.

4. THE PROPOSED ALGORITHM

LD_Rank algorithm inspired from RL concepts. So in this section, we first review RL notations. Thence, the LD_Rank algorithm is introduced.

4.1 Reinforcement learning

Reinforcement learning, one of the Machine Learning (ML) techniques, learns by interactive in dynamic environment. As well as, it is a strong tool in determining effective states in states space. In an RL problem, the learner is called the agent who learns by its interaction with the environment and it acquires knowledge through reward or punishments of an action undertaken [6].

In an agent-based system with RL, 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, defined by $\pi(s, a)$, is the probability of selecting action a when agent is concerned with states. 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 [6]. Reinforcement signal is a scalar signal and it demonstrates 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, denoted as follows [6]:

$$V^{\pi} = E_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | S_t = s \right\} \qquad 0 \le \gamma \le 1 \ (5)$$

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

4.2 LD_Rank algorithm

In our algorithm, we use link structure of datasets on the LOD and define ranking in shape of RL problem. The proposed approach is named LD_Rank. In LD_Rank algorithm, an agent is considered as a surfer (cruiser) and each dataset as a state. In each dataset on the LOD (state), the surfer (agent) clicks on one of the available links in that dataset with a uniform probability, and goes to the next state. Thus, an agent's action is to click on one of the links randomly with a uniform probability. In the other words, when surfer selects next dataset by clicking randomly on one of the links in the current dataset, the policy π is equal to $1/ol$(current state), where ol(current state) is the output-degree of the current dataset. The reward is given when a transition occurs from a current state (ds_j) to other state (ds_i) denoted by:

$$r_{ji} = \frac{1}{ol(ds_j)} \quad (6)$$

where $ol(ds_j)$ is the output-degree of dataset j. Therefore, dataset with less out degree gives more reward to its children.

We define the score of dataset i to be the expected value of sum of discounted rewards that agent cumulates during traveling via datasets to attain dataset i. Afterwards agent adds the received reward r_{ji} to the discounted cumulated rewards. Accordingly, score of dataset i is probability of attaining it from other datasets multiplied through sum of the transition reward and discounted cumulated rewards. The score of dataset is defined as follows:

$$dsr_{t+1}(ds_i) = \sum_{ds_j \in p(ds_i)} \left(\left(\frac{pprob(ds_j)}{ol(ds_j)}\right) \times (r_{ji} + \gamma dsr_t(ds_j))\right) \quad (7)$$

where $dsr_{t+1}(ds_i)$ is rank of dataset i in time $t+1$ and $dsr_t(ds_j)$ shows the rank of dataset j in time t, $p(ds_i)$ is the set of datasets on the LOD that point to dataset i, $pprob(ds_j)$ is the presence probability of the agent into dataset j. $ol(ds_j)$ is the output-degree of dataset j and r_{ji} the reward for transition from dataset j to i denoted by Equation (6). So, the rank of dataset ds depends on the output-degree and rank of the datasets pointing to dataset ds.

Algorithm: LD_Rank

//DS: all datasets on the LOD

//pprob: presence probability of the agent at dataset j

//dsr: LD_Rank vector

//ε: A small positive number

Initialize dsr, pprob vectors

$\delta \leftarrow 0$

while $(\delta > \varepsilon)$

For every dataset i ∈ DS

$$pprob_{new}(ds_i) = \left(1 - d/n + d \times \sum_{ds_j \in p(ds_i)} pprob(ds_j)/ol(ds_j)\right)$$

End for

$\delta \leftarrow \|pprob_{new} - pprob\|$

$pprob \leftarrow pprob_{new}$

End while

$\delta \leftarrow 0$

while $(\delta > \varepsilon)$

For every dataset ds ∈ DS

$r_{ji} = 1/ol(ds_j)$

$$dsr_{new}(ds_i) = \sum_{ds_j \in p(ds_i)} ((pprob(ds_j)/ol(ds_j)) \times (r_{ji} + \gamma dsr(ds_j)))$$

End for

$\delta \leftarrow \|dsr_{new} - dsr\|$

$dsr \leftarrow dsr_{new}$

End while

Algorithm 1: The proposed LD_Rank.

The value of $pprob(ds_j)/ol(ds_j)$ is the probability of attaining dataset i from dataset j. It is equal to presence probability of the agent at state j multiplied by selection probability of dataset i when agent is in state j. Whereas the agent selects one of the links by uniform probability distribution, the selection probability of dataset i from j is equal to one divided by output-degree of dataset j. $dsr(ds_j)$ is the rank of dataset j that presents cumulated discounted rewards the agent has received till getting to dataset j. Thus, rank of dataset i based on Equation (7) depends on the output-degree and rank of the datasets pointing to i. Using the policy evaluation idea [6] in the LD algorithm, we propose a practical approach to estimate the rank of each LOD dataset. As Equation (7) illustrates LD_Rank is computed recursively like PageRank. The pseudo code in Algorithm 1 demonstrates our LD_Rank procedure. Eventually, we will have the LD_Rank vector and LOD datasets sorted in the descent order. With regards to the pseudo code, it is clear that the time complexity of LD_Rank is linear.

4.2.1 LD_Rank convergence

In this section, we prove convergence of LD_Rank algorithm.

Lemma 1. In LD_Rank algorithm (Equation (7)), $dsr(ds_i)$ converges.

Proof. The rank scores in LD_Rank are computed recursive through Equation (7). It has to be considered that rank score of LOD datasets with zero input-degree (input-degree of a dataset is equal the number of links from other datasets to the dataset) are not changed in iterations and their final amounts are equal their initial values. However, some LOD datasets with zero input-degree have output-links to other datasets; Hence, rank score of other datasets are affected via the rank score of these datasets. Thus, Equation (7) is rewritten as follows:

$$dsr_{t+1}(ds_i) = \sum_{ds_j \in p(ds_i)} \left(\left(\frac{pprob(ds_j)}{ol(ds_j)}\right) \times \left(r_{ji} + \gamma dsr_t(ds_j)\right)\right) + \sum_{ds_k \in p'(ds_i)} \left(\left(\frac{pprob(ds_k)}{ol(ds_k)}\right) \times \left(r_{ki} + \gamma dsr(ds_k)\right)\right) \quad (8)$$

where $p(ds_i)$ is the set of LOD datasets with non-zero input-degree that point to dataset i and $p'(ds_i)$ is the set of LOD datasets with zero input-degree that point to dataset i. Amount of the second term of the left side in Equation (8) is constant. In the other words, it is not updated during iterations. This amount for i-th LOD dataset is defined as $k(ds_i)$:

$$dsr_{t+1}(ds_i) = \sum_{ds_j \in p(ds_i)} \left(\left(\frac{pprob(ds_j)}{ol(ds_j)}\right) \times \left(r_{ji} + \gamma dsr_t(ds_j)\right)\right) + k(ds_i) \quad (9)$$

In here, we denote the matrix M and the vectors X, dsr, K as follows: M is a $n' \times n'$ matrix that each element is denoted as:

$$m(ij) = \begin{cases} \left(\frac{pprob(ds_j)}{ol(ds_j)}\right) & ds_j \in p(ds_i) \\ 0 & otherwise \end{cases} \quad (10)$$

where $pprob(ds_j)$ is the presence probability of agent at LOD dataset j. This probability is constant during LD_Rank computations, since it is independently computed sooner. $ol(ds_j)$ is output-degree of LOD dataset j. n' is the total number of datasets with non-zero input-degree.

X is a $n' \times 1$ vector that its element is as:

$$x(ds_j) = \frac{1}{ol(ds_j)} \quad (11)$$

Table 1. Overview of two LOD benchmark datasets.

	# triples	# days	total # hits	# plain hits	# RDF hits	# HTML hits	SPARQL
DBpedia	109,734,227 (979,769)	112	56,946,461 (508,450)	15,257,263 (136,225)	6,348,225 (56,680)	15,435,125 (137,813)	13,408,518 (119,718)
SWC	79,136 (659)	120	6,196,200 (51,635)	1,341,497 (11,179)	231,750 (1,931)	1,143,867 (9,532)	654,120 (5,451)

dsr is vector containing the score of LOD datasets. K is $n' \times 1$ vector that that i-th its elements is $k(ds_i)$. γ is the discount factor that $0 < \gamma < 1$.

Based on the mentioned definitions, we can rewrite Equation (9) as a matrix form [12]:

$$dsr = \gamma M dsr + MX + K \qquad (12)$$

As observe γM is the coefficient of vector dsr, and MX and K are to constant vector. With respect to Equation (10) all elements on main diagonal of matrix M is 0 as well other elements are less than 1. Therefore, $\|\gamma M\|_{\infty} < 1$ according to the convergence theorem of iterative methods [13], It can be concluded $dsr(ds_i)$ in Equation (7) converges.

5. EVALUATION AND EXPRIMENTAL RESULTS

To evaluate the proposed methods, they are assessed experimentally on well-known LOD benchmark datasets based on standard criteria.

5.1 Benchmark Datasets

We conducted some experiments on DBpedia and SWC (aka "Semantic Web Dog Food") benchmark datasets. All two datasets differ greatly with regards to several of their basic characteristics, such as size (in number of RDF triples), connectedness in the LOD cloud, functionality beyond serving of LD and etc. They together provide us with proper coverage of the various types of benchmark datasets which make up the web of data (on the LOD).

In here, we will give an introduction to each benchmark dataset. We individualize requests to the SPARQL Protocol And RDF Query Language (SPARQL) endpoints of each benchmark dataset and three related kinds of Uniform Resource Identifiers (URI) which all reflect the identical resource, in the sense that the plain resource URI is the identifier of a non-information resource [14] such as "WWW2012", while the related RDF and HTML document URI are identifiers for information resources, or representations in different formats about WWW2012 (see Figure 4). We used the SPARQL GUI in [2, 3, 4] for testing out SPARQL queries on LOD benchmark datasets which we created by loading in RDF from files and/or remote URIs.

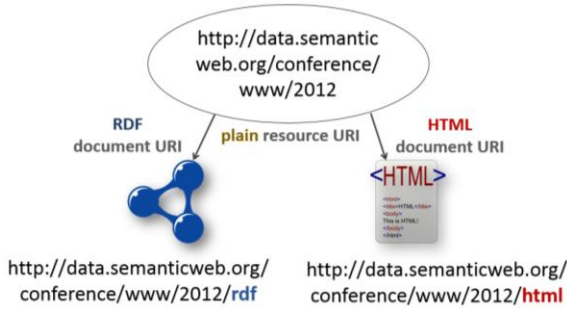

Fig 4: Plain resource, RDF and HTML representations.

5.1.1 DBpedia semantic data collection

The largest benchmark dataset in our evaluation is the DBpedia [15], which provides LD based on an extraction of structured data from Wikipedia. Due to its wide coverage in background knowledge NEs such as people, places, species and etc., DBpedia can be considered a hub into the Web of LD, in that it is used as a point of reference through many other LOD datasets. The DBpedia benchmark dataset serves both RDF and HTML documents about its resources. For DBpedia, we had access to server log files dating from 01/08/2015-21/11/2015 (See Table 1).

5.1.2 Semantic Web Dog Food

The smallest benchmark dataset in our assess in terms of RDF triples (~80,000 RDF) is served by the Semantic Web Conference (SWC) metadata site. SWC holds RDF data about a number of large, international conferences in the Web and Semantic Web zone, such as WWW, ISWC and ESWC, as well as a growing number of workshops. For each such event, detailed data about papers, authors, events and other NEs is provided, both as RDF and as HTML documents. For this benchmark dataset, we had access to server log files dating from 01/08/2015-29/11/2015 (See Table 1).

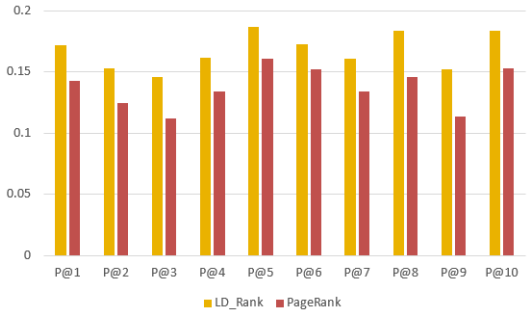

Fig 5: Comparison of LD_Rank with PageRank in the P@n measure on DBpedia benchmark.

5.2 Evaluation Measures

In order to assess the proposed algorithm, we use two well-known and related LOD benchmark datasets and use two common evaluation measures which are widely used in Information Retrieval (IR), namely Precision at n (P@n) [16] and Mean Average Precision (MAP) [16]. Their definitions are summarized as follows:

a) Precision at n (P@n): This criterion illustrates the ratio of top relevant documents to total number of documents (n) in presented outcomes. In fact, it shows system accuracy [16]:

$$P@n = \#\ of\ relevant\ in\ top\ n\ results/n \qquad (13)$$

b) 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 through Equation (14) [16]:

$$AP = \sum_{i=1}^{n} \frac{(P@i.rel(i))}{\#total\ relevant\ docs\ for\ one\ query} \qquad (14)$$

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

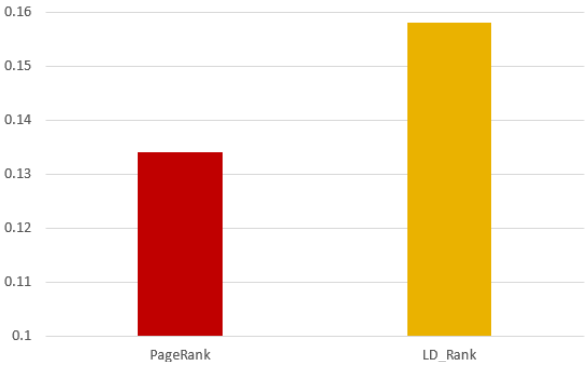

Fig 6: Comparison of LD_Rank with PageRank in the MAP measure on DBpedia benchmark.

5.3 Experimental Results

The first experimental outcomes compare LD_Rank with PageRank (mapped to web of LD) as a well-known connectivity-based ranking algorithm. In the experiments, the factor γ in LD_Rank was set to 0.9 and the damping factor in PageRank was set to 0.85. The outcomes of evaluation on DBpedia LOD benchmark dataset are shown in Figures 5–6.

Figure 5 illustrate the obtained P@n. As shown, the obtained values for LD_Rank are higher than those for PageRank. Figure 6 shows that LD_Rank obtains improvement over the PageRank in terms of MAP measure.

Graphical evaluations of outcomes on SWC benchmark dataset are depicted in Figures 7–8 in terms of P@n and MAP measures, respectively. About the Figure 7, the values grabbed for LD_Rank are higher than those for PageRank. Figure 8 shows that LD_Rank exceed PageRank in performance.

A close look at the outcomes demonstrates that LD_Rank is a suitable algorithm for ranking of the datasets on the LOD. The results signify that LD_Rank algorithm makes larger improvements on DBpedia benchmark dataset in compared to SWC benchmark dataset. It should be noticed that LD_Rank and PageRank are two connectivity-based ranking algorithms and they are influenced via connectivity features of LOD datasets.

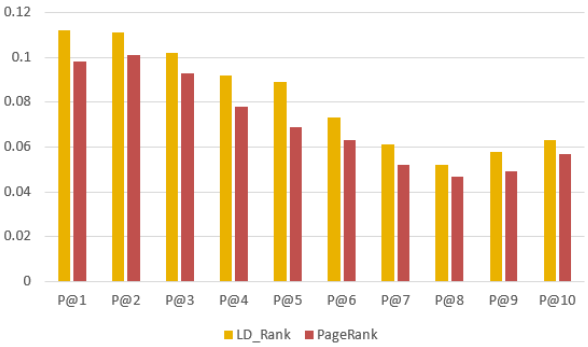

Fig 7: Comparison of LD_Rank with PageRank in the P@n measure on SWC benchmark.

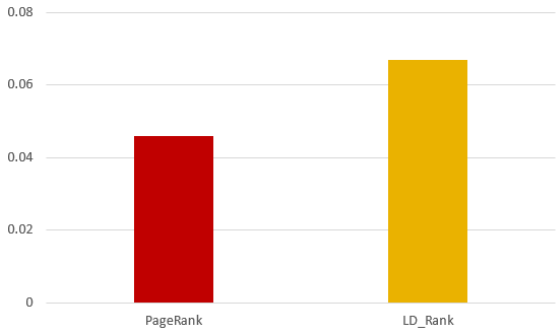

Fig 8: Comparison of LD_Rank with PageRank in the MAP measure on SWC benchmark.

```
Algorithm: DS_Rank(LinkMatrix, NumNode)
//Input: LinkMatrix, NumNode
//Output: rank
for (i = 0; i ≤ NumNode; i + +) do
  rank[i] ← 1
end for
for (i = 0; i ≤ NumNode; i + +) do
  for (j = 0; j ≤ NumNode; j + +) do
    numAllLink[i] ← LinkMatrix[i][j]
  end for
end for
for (iteration = 1; iteration ≤ NumIteration; iteration + +) do
  rankPrevIteration ← NewFloat[NumNode]
  for (p = 0; p ≤ NumNode; p + +) do
    rankPrevIteration[p] ← rank[p]
  end for
  for (i = 0; i ≤ NumNode; i + +) do
    rank[i] ← 1 − alpha
    for (j = 0; j ≤ NumNode; j + +) do
      if (LinkMatrix[j][i] ≠ 0) then
        rank[i] ← rank[i] + alpha * (rankPrevIteration[j] *
        LinkMatrix[j][i]/numAllLink[j])
      end if
    end for
  end for
end for
```

Algorithm 2: Ranking LOD datasets in web of data.

6. CONCLUSION

In this paper, using the RL notations, we first proposed LD_Rank algorithm which is a novel connectivity-based algorithm for ranking datasets on the web of LD. This algorithm considers rank definition of a LOD dataset as an RL problem where the reward for transition from current LOD dataset to the next LOD dataset is proportionate to the reverse of the output-degree of the current LOD dataset. In fact, LD_Rank models the user who cruises the social semantic web by accumulating transition rewards to obtain rank of each LOD dataset. Experimental outcomes showed that LD_Rank can attain much better outcomes than PageRank in standard criteria. The linear complexity of the LD_Rank signifies the scalability of this algorithm on large datasets. Therefore,

LD_Rank can be used either as a connectivity-based ranking algorithm in semantic web search engines like Swoogle. Also we observed that LD_Rank behaves differently on various benchmark datasets (it makes larger improvements on DBpedia benchmark dataset in comparison to SWC benchmark dataset). Hence, it can be concluded that LD_Rank has high performance in LOD. As future works, we plan to explore positive outcomes of appropriate RDF ranking in different applications such as Question Answering [17] and NE disambiguation [18] in the context of property tagging.

```
Algorithm: DS_Rank(LinkMatrix, NumNode, α, NumAttr)
//Input: LinkMatrix, NumNode, α, NumAttr
//Output: rank
for (i = 0; i ≤ NumNode; i + +) do
  rank[i] ← 1
end for
for (i = 0; i ≤ NumNode; i + +) do
  for (j = 0; j ≤ NumNode; j + +) do
    for (t = 0; t ≤ NumNode; t + +) do
      numAllLink[i][t] ← numAllLink[i][t] + LinkMatrix[i][j][t]
    end for
  end for
end for
for (iteration = 1; iteration ≤ NumIteration; iteration + +) do
  rankPrevIteration ← NewFloat[NumNode]
  for (p = 0; p ≤ NumNode; p + +) do
    rankPrevIteration[p] ← rank[p]
  end for
  for (i = 0; i ≤ NumNode; i + +) do
    rank[i] ← 1 − alpha
    for (j = 0; j ≤ NumNode; j + +) do
      for (t = 0; t ≤ NumNode; t + +) do
        if (LinkMatrix[j][i][t] ≠ 0) then
          rank[i] ← rank[i] + alpha * (rankPrevIteration[j] *
          LinkMatrix[j][i][t] * α[t]/numAllLink[j][i][t])
        end if
      end for
    end for
  end for
end for
```

Algorithm 3: Ranking LOD datasets with weighted links in web of data.

7. REFERENCES

[1] J. Breslin, A. Passant, and S. Decker, The social semantic web: Springer Science & Business Media, 2009.

[2] B. Farhadi, "Enriching Subtitled YouTube Media Fragments via Utilization of the Web-Based Natural Language Processors and Efficient Semantic Video Annotations," Global Journal of Science, Engineering and Technology, pp. 41-54, 2013.

[3] B. Farhadi, "Creating a Semantic Academic Lecture Video Search Engine via Enrichment Textual and Temporal Features of Subtitled YouTube EDU Media Fragments," International Journal of Computer Applications, vol. 96, pp. 13-18, 2014.

[4] B. Farhadi and M. B. Ghaznavi Ghoushchi, "Creating a Novel Semantic Video Search Engine through Enrichment Textual and Temporal Features of Subtitled YouTube Media Fragments," in 3rd International conference on Computer and Knowledge Engineering (ICCKE), 2013, pp. 70-78.

[5] [Online]. Available: http://lod-cloud.net

[6] R. Suttan and A. Barto, "Reinforcement Learning: An Introduction, MIT Press," ed, 1998.

[7] G. Salton, "The SMART retrieval system—experiments in automatic document processing," 1971.

[8] R. Blanco, P. Mika, and S. Vigna, "Effective and efficient entity search in RDF data," in The Semantic Web–ISWC 2011, ed: Springer, 2011, pp. 83-97.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation algorithm: bringing order to the web," in 7th World Wide Web Conference, 1998.

[10] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," Journal of the ACM (JACM), vol. 46, pp. 604-632, 1999.

[11] P. Boldi, M. Santini, and S. Vigna, "PageRank as a function of the damping factor," in Proceedings of the 14th international conference on World Wide Web, 2005, pp. 557-566.

[12] V. Derhami, E. Khodadadian, M. Ghasemzadeh, and A. M. Z. Bidoki, "Applying reinforcement learning for web pages ranking algorithms," Applied Soft Computing, vol. 13, pp. 1686-1692, 2013.

[13] S. Berger Henengouwen, Engineering Numerical Analysis, Cybered Incorporated, 1998.

[14] I. Jacobs and N. Walsh, "Architecture of the world wide web," 2004.

[15] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, Dbpedia: A nucleus for a web of open data: Springer, 2007.

[16] A. Moffat and J. Zobel, "Rank-biased precision for measurement of retrieval effectiveness," ACM Transactions on Information Systems (TOIS), vol. 27, p. 2, 2008.

[17] A. Dessi and M. Atzori, "A machine-learning approach to ranking RDF properties," Future Generation Computer Systems, 2015.

[18] B. Farhadi, "NER-FL: A Novel Named Entity Recognizer of Farsi Language using the Web-Based Natural Language Processors and Semantic Annotations," International Journal of Computer Applications, vol. 98, 2014.