DBtrends: Exploring Query Logs for Ranking RDF Data

Edgard Marx
AKSW, University of Leipzig
Leipzig, Germany
marx@informatik.uni-leipzig.de

Amrapali Zaveri
Stanford Center for Biomedical Informatics Research
Stanford University, USA
amrapali@stanford.edu

Diego Moussallem
AKSW, University of Leipzig
Leipzig, Germany
moussallem@informatik.uni-leipzig.de

Sandro Rautenberg
Midwestern State University (UNICENTRO)
Guarapuava, PR, Brazil
srautenberg@unicentro.br

ABSTRACT

Many ranking methods have been proposed for RDF data. These methods often use the structure behind the data to measure its importance. Recently, some of these methods have started to explore information from other sources such as the Wikipedia page graph for better ranking RDF data. In this work, we propose DBtrends, a ranking function based on query logs. We extensively evaluate the application of different ranking functions for entities, classes, and properties across two different countries as well as their combination. Thereafter, we propose MIXED-RANK, a ranking function that combines DBtrends with the best-evaluated entity ranking function. We show that: (i) MIXED-RANK outperforms state-of-the-art entity ranking functions, and; (ii) query logs can be used to improve RDF ranking functions.

1. INTRODUCTION

Over the last year we see an emerging necessity in developing ranking functions in order to facilitate the content access. This necessity became very evident in the Semantic Web domain with the arising of the emerging large structured datasets. Although a big portion of this datasets is free available, users can not easily consume it. During the last years many ranking functions where designed with a view to address specific or broad range of purposes such as entity summarization [3] document retrieval [9, 21, 6] and entity linking [4] among others. This ranking functions usually explore statistics [6] or the structure of the data [9, 18] to measure its relevance. However, a fundamental principle in Semantic Web is that the resources represent concepts in the real world. Therefore, there are a huge amount of features and indicators that can be used to measure how important a piece of information is. For example, to measure the relevance of country to a person and/or a policy action, one can use the GDP (Gross Domestic Product) or HDI (Human Development Index). Furthermore, the relevancy is highly tied to the context. For instance, a public policy coordinator can choose to use the HDI in an ascending order to decide welfare policies while a emigrant can use the same index in descending order to decide where to move. Another important observation is that the relevance can change across time.

Presently, ranking algorithms have started to become more personalized. This means that instead of using only the data structure itself, approaches have begun to use third-party information, i.e. information that cannot be found in the data itself. For instance, one can use the location, language or the previously visited web sites and their frequency. That information helps enhance the rank of the query results [14, 16].

According to [1, 26], a *good* measure of the importance of a piece of information is it's occurrence in real users query. Hence, query logs are highly useful for ranking information. The central idea of using query logs is that it allows to extract the users interests across time. As users interest tend to change over time, query logs provide a better idea about the resource relevance when compared with other methods that use only graph-based metrics. Thereafter, query logs can also be used to generate a more personalized ranking e.g. users from different countries may search for different things. In this work, we propose two ranking functions for RDF data: DBtrends, a ranking function that uses external information to rank resources in the dataset, more precisely, the query logs, and; MIXED-RANK, a ranking function that uses a combination of DBtrends and the best-evaluated ranking function for RDF data. We extend the previously introduced Spearman's footrule to deal with heterogeneous rankings [15]. Moreover, we provide an extensive evaluation between main property, class and entity ranking functions in a standard benchmark for measuring RDF ranking functions [15].

The rest of the paper is structured as follows: The related work is discussed in Section 2. Thereafter, we introduce an extension of the Spearman's footrule for evaluating heterogeneous ranks in Section 3. Section 4 introduces the rank-

ing function based on query logs. The evaluation, results and performance of different ranking functions for entities, properties and classes are presented in Section 5. Finally, we conclude with our plans for future work in Section 6.

2. RELATED WORK

Dynamic and Static Ranking Functions. Ranking functions have been studied for a long time as they are useful for measuring the relevance of a certain feature. Ranking functions can be dynamic or static. Dynamic ranks change according to a given third-party information. That is the case of the ranks designed for information retrieval that use the occurrence of terms in a given query to measure the relevance of the resource in the knowledgebase [9, 21, 6]. There are also dynamic ranks designed for the task of entity linking introduced by *Cheng et al.* [4], which use a target text for suggesting the possible linking candidates. Another example of dynamic rank is LDRANK [1], a query-based algorithm for ranking RDF resources. LDRANK is used for generation of semantic snippets that uses a combination of explicit and implicit relationships inferred from RDF resources. The explicit relationship is extracted through a PAGE-RANK like algorithm applied to the RDF graph. The implicit relation is inferred from the text of the resource web page.

Static ranks are those that can be derived from a particular data structure or information and do not change. That is the case of, for example, Page-Rank [18], DBpedia Page-Rank [23] (DB-RANK) and RELIN [3]. Static ranks are the basis of wide range of applications such as Search Engines [2], Linked Data browsers [5], Link Discovery [17] and Machine Learning [24].

Ranking algorithms for RDF data are usually designed for three main features: (1) classes, (2) properties and (3) entities (objects or individuals). Ranking for entities is the most common type of ranking available. For instance, a query "persons" can return thousands of entities if applied to the DBpedia knowledge base, but not all the information can be useful. In such a way, entity ranks can help search engines sort the resulting set according to its relevance.

Hits [12] rank is based on the number of incoming and outgoing edges of a given node. Hits is designed to measure the number of incoming links of a page, which indicates its authority and the hub, which is the sum of all the authorities of the pages pointed by the outgoing links TripleRank [8] uses Hits and applies it to (1) properties and entities and shows that it produces better results for faceted navigation than using the incoming links measure. *Page-Rank* [18] is based on the probability of randomly finding a page in a network by following a path starting from any other page. The concept of Page-Rank can be applied to any graph network. For instance, *DBpedia Page-Rank* [23] is a variant of the original Page-Rank algorithm where the rank of a DBpedia entity corresponds to the rank of its Wikipedia page.

Entities can have a large number of properties, but a big portion of them might not be interesting to users. To deal with this problem of so-called *entity summarization*, approaches implementing different types and levels of abstractions have been introduced. *RELIN* [3] is a ranking function that explores a variant of the random surfer model [18]—used in previous methods [7, 25, 19]—revised by a more specific notion of centrality designed for property ranking. That is, a computation of the central elements involving similarity (or relatedness) among them as well as their informativeness. The similarity is given by a combination of the relatedness between their properties and their values. Relatedness is the probability that a property-value pair appears together. The smaller the probability of a property-value pair, the more information its occurrence provides. For instance, there is a higher probability of its occurrence of the property-value pair (*country*, *United States*), in United States cities than the pair *largest city* and *United States*. That is, there are fewer entities containing the property-value (*largest city*, *United States*) than the property-value (*country*, *United States*). Taking this into consideration, the property-value pair (*country*, *United States*) is more general to the United States cities than the pair (*largest city*, *United States*). The authors also implemented a baseline called *RandomRank*, which trivially generates a random ranking of property-value pairs. Also, Roa-Valverde et al. [20] provides a systematic review on rank approaches for the Web of Data. However, none of these methods take into account the users needs which can be better identified by using the users performed queries [14, 16].

Statistics. Another type of measure that can be used for ranking is the dataset statistics. Some of the dataset statistics are, for example, the number of instances of a certain resource. Another measure is the number of references, predicates as well as incoming or outgoing links. These statistics are specifically useful for ranking entities.

However, as the Semantic Web usually deals with real world entities, some approaches have introduced rankings using statistics coming from external sources [3]. That is, those statistics are outside the dataset. This approach can be applied to knowledge bases because an entity usually refers to a real world resource. Thus, for instance, one can extract statistics related with the resource's web page —i.e. the incoming and outgoing links. This possibility opens new perspectives for RDF ranking. That is, the use of information coming from other source that is not in the graph structure to rank resources in the graph. Notice that statistics can also be dynamic or static.

Currently, there are many available search engines that are able to crawl big portion of the Web as Google, Yahoo and Bing. Apart from that, they can find related content to a given query. By crawling a large volume of the Web, the search engines can also became a big source of information that can help when ranking resources. One source of information is, for instance, the number of available Web documents with a particular term or sentence. Another source of information is the query log. For instance, *Google Trends*[1] is a public web platform containing the historical search index of a particular term in Google Knowledge Graph topics, Search interest, trending YouTube videos, and Google News articles. The Google Trends index is based on how often a particular term is searched for in relation to the total search-

[1] `https://www.google.com/trends/`

volume across various regions of the world, and in various languages. Google Trends index can be used to either rank entities, properties, or their correlations.

Benchmarks. Previous works [3, 4] evaluate ranking function by using the best rank selection. That is, first different ranking functions were applied to a target data producing different ranks. The produced ranks were then shown to humans, who select the most relevant one. This method did not allow reproducing the experiments as it was not possible to use the same users to evaluate other ranks. Thus, the DBtrends benchmark [15] was introduced. The DBtrends benchmark contains classes, properties and entities ranking profiles of 60 users, out of which 30 are North Americans and 30 Indians. Moreover, it also contains both the when (time) and where (location) the profiles were generated, which allows us to measure trends as well as cultural influences across different countries. The Indians and Americans were chosen because they represent the two major working groups in Amazon Mechanical Turk. The benchmark was built upon four tasks: First, in Task 1, the user was asked to rank 20 entities extracted from the top five entities of the top four DBpedia classes. Thereafter, the user had to rank the most instantiated classes and predicates of the highest ranked entity in the Task 1. Finally, the user had to score her confidence in performing the previous ranking tasks.

Rank Similarity Functions. There are several works in measuring similarity between two ranks. However, the problem of rank similarity is different from rank quality. A rank quality is a measure that indicates how *good* is a rank, for example Discounted Cumulative Gain [10], Rank similarity is focused on finding how distant or close two ranks are and it is computed by comparing the position of the elements. There are two approaches to evaluate the similarity between two ranks: Spearman's Footrule [22] and Kendall rank correlation coefficient [11]—usually referred to as Kendall's tau coefficient. Both rank similarity functions are designed to measure the distance among ranks containing the same set of elements. Spearman's Footrule measures the distance between an element belonging to two different ranks. Kendall's tau coefficient computes the number of swap (*Bubble sort*) operations necessary to sort the first rank according to the second. However, the correlation between two ranks (r_β, r_γ) of Spearman's Footrule S_F is bounded by Kendall's tau coefficient [13] K, that is:

$$\forall r_\beta, r_\gamma \; K(r_\beta, r_\gamma) \leq S_F(r_\beta, r_\gamma) \leq 2K(r_\beta, r_\gamma) \tag{1}$$

3. AN EXTENDED HETEROGENEOUS RANK SIMILARITY FUNCTION

The result of a ranking function is a rank, that is, a sequence of elements sorted in a particular order. Spearman's Footrule [22] is a distance measure function designed to measure similarities among homogeneous ranks. However, Spearman's Footrule is not defined for heterogeneous ranks, in other words, ranks that contain different sets of individuals—for instance, $\Gamma = \{a, b, c\}$ and $\Lambda = \{a, b\}$. Therefore, we propose a new similarity ranking function based on Spearman's Footrule for measuring similarity among heterogeneous ranks.

According to the Spearman's Footrule, the similarity between two ranks is measured by a summation of the difference among the positions of the elements in the two ranks r_β and r_γ. The Spearman's Footrule is formally defined by the function S_F as follows:

$$\begin{aligned} F(r) =& (F | r = (f_1, f_2, ..., f_n) : f \in F) \\ S_F(r_\beta, r_\gamma) =& \sum_{f_\beta \in F(r_\beta)} | \, r_\beta^{-1}(f_\beta) - r_\gamma^{-1}(f_\beta)| \end{aligned} \tag{2}$$

The maximal distance between two ranks in Spearman's Footrule can be obtained by induction in a very trivial process and will not be discussed here. However, the maximal distance between two ranks is given by the function $S_{F_{max}}$ that receives a rank size and computes the maximal distance. To overcome the problem of measuring heterogeneous ranks, we proposed a variant of Spearman's Footrule [15]. The difference from the original formula is that it consists of the sum of the position of the element of the highest rank that does not intersect, which can be formally defined as follows[2]:

$$\begin{aligned} D(r_\beta, r_\gamma) =& D_\cap(r_\beta, r_\gamma) + D_{\not\cap}(r_\beta, r_\gamma) \\ D_\cap(r_\beta, r_\gamma) =& \sum_{f_\beta \in F(r_\beta) \cap F(r_\gamma)} | \, r_\beta^{-1}(f_\beta) - r_\gamma^{-1}(f_\beta)| \\ D_{\not\cap}(r_\beta, r_\gamma) =& \begin{cases} \sum_{f_\beta \notin F(r_\beta) \cap F(r_\gamma)} r_\beta^{-1}(f_\beta) & \text{else if } |r_\beta| > |r_\gamma| \\ \sum_{f_\gamma \notin F(r_\beta) \cap F(r_\gamma)} r_\gamma^{-1}(f_\gamma) & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

The proposed extension of the original Spearman's Footrule can be divided into three cases: (1) when the ranks are homogeneous, (2) when they intersect and (3) when they do not intersect. The simplest cases are the homogeneous and without intersection ones. When the ranks are homogeneous, the distance can be measured by the function $S_{F_{max}}$. In other cases, the distance is given by an arithmetic progression of the size of the biggest rank. The arithmetic progression is used as an anchor because an arithmetic progression of a rank with n entries is higher than the maximal distance of Spearman's Footrule.

$$\sum_{i=1}^{|F(r)|} i > S_{F_{max}}(|F(r)|) \tag{4}$$

Apart from that, the similarity function for heterogeneous ranks uses the position of the element in the biggest rank. The biggest distance between two ranks occurs when they do not have elements in common. Thus, a maximal distance between two ranks is defined by the function d_{max} as follows:

$$D_{max}(r_\beta, r_\gamma) = \begin{cases} S_{F_{max}}(|F(r_\beta)|) & \text{if } r_\beta \equiv r_\gamma \\ D_{\not\cap}(r_\beta, r_\gamma) & \text{otherwise} \end{cases} \tag{5}$$

[2]Herein, we define the length of a list r as a natural extension of a cardinality set denoted by $|r|$.

However, there are some scenarios one can argue towards it's completeness. Let us suppose that we have three ranks $\Gamma = \{a, b, c\}$, $\Lambda = \{a, b\}$, $\Theta = \{e, f\}$. According to the proposed function, $D(\Gamma, \Lambda) = D(\Theta, \Lambda)$. The problem is that while Γ—that has one disjoint elements, Θ has two disjoint element. Thus, we propose an extension where the different elements in the sets is taken in consideration. In the proposed extension, we redefine $D_{\not\cap}(r_\beta, r_\gamma)$ as follows:

$$D_{\not\cap}(r_\beta, r_\gamma) = \sum_{f_\beta \notin F(r_\beta) \cap F(r_\gamma)} r_\beta^{-1}(f_\beta) + \sum_{f_\gamma \notin F(r_\beta) \cap F(r_\gamma)} r_\gamma^{-1}(f_\gamma) \quad (6)$$

4. DBTRENDS

The idea behind the proposed ranking function is to use previous performed user queries to track the relevance of an RDF concept across time. We follow the assumption that user performed queries can be used to predict and better estimate the users intention and thereby the relevance of an RDF resource [1, 26]. As a users opinion can change across time, it can also be used to maintain the rank updated to the users interest. Big search engines can generate high confident query based ranks, because they have the required amount of data to generate it. In case of Google, this information is publicly available through Google Trends.

Moreover, the work of [1, 6, 9, 21] shows that is possible to map query terms to RDF concepts. Thus, we estimate the relevance of the resource by tracking its occurrence in the query logs. We labeled this rank as DBtrends and the process of generate it is as follows:

- First, the labels of the entities are extracted and used to acquire the search history in query logs e.g. Google Trends (see (1)-(2) in Figure 1). In Google Trends, the search history can be filtered by time, geo-location (globally, countries and states) and category (Health, Games and Finance etc.);
- Thereafter, the entity ranks are used as a base to propagate the rank to the classes (see (3)-(4) in Figure 1). In this step, the rank is distributed from entities to the most abstract class.

In the RDF model, classes C and entities E form a hierarchical graph. Thus, the ranks of the entities can propagate until the class is in the highest hierarchy. In order to do that, a *Breadth First Search* (BFS) algorithm is applied starting from the entities. The proposed ranking function assumes that the importance of a class is given by the data instantiated with it, and, generic classes are less expressive in defining something and therefore, less important. Thus, an importance of the class is measured by the (1) importance of the data in it, and, (2) its hierarchical position. For ranking classes, we use the average rank of resources instantiated with the classes R_i divided by two. Given the function $GoogleTrends(r)$ used for extracting the Google Trends search index of a given resource, the proposed rank is defined by the *DBtrends* function as follows:

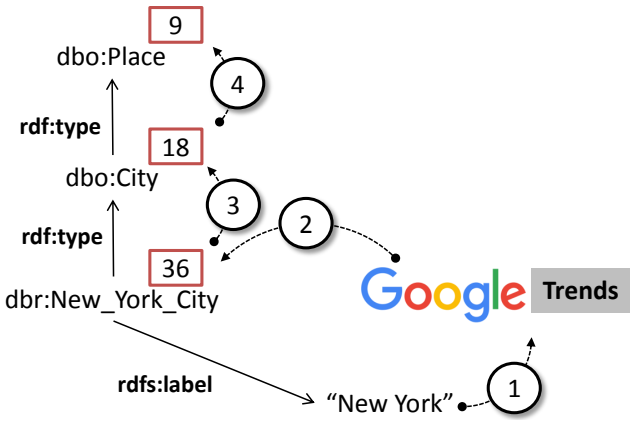

Figure 1: The DBtrends rank workflow, which consists of trend extraction (1)-(2) and propagation (3)-(4).

$$DBtrends(r) = \begin{cases} \frac{1}{2|R_i|} \sum_{r_i \in R_i} DBtrends(r_i) & \text{if } r \in C \\ GoogleTrends(r) & \text{if } r \in E \end{cases}$$

Although the DBtrends rank uses statistics of query logs of Google available via Google Trends, the same principle can also be applied to use indexes extracted from other sources (e.g. PAGE-RANK, SHARED-LINKS and SEO-PA in Table 1).

5. EVALUATION

In this section, we describe the evaluation performed for benchmarking different entity, property and class ranking functions. The benchmark selected for evaluation was the DBtrends benchmark [15], since it is publicly available and allows reproducibility. Furthermore, it allows the benchmarking of a wide range of RDF ranking functions such as classes, properties and entities. Thereafter, we measure the rank distances and provide the results. We conclude by giving some insights about the achieved results.

The evaluation was designed to answer the following research questions:

RQ.1 How different are the rankings performed across different countries?

RQ.2 Is there any similarity between the ranking performed by different users?

RQ.3 Which of the three: classes, properties or entities ranking functions performs better?

RQ.4 Is there any similarity between the rankings performed by a particular ranking function in any particular type of resource (classes, properties, entities)?

Ranking Functions. We evaluate the ranking functions on three levels: (1) classes, (2) properties and (3) entities. The

total number of evaluated ranking functions for classes, properties and entities were three, four and twelve respectively. The evaluated classes functions were: the number of instances of the classes—the number of resource instantiations that are sub-type of the class—in the dataset in (1) descending (Instances↑) and (2) ascending order (Instances↓), and; (3) the best (Best) class rank (R^c). The evaluated properties functions were: the number of instances of the property in the dataset in (1) descending (Instances↑) and (2) ascending order (Instances↓); (3) the RELIN [3] property rank, and; (4) the best (Best) property rank (R^p). The evaluated entities functions were (1) the incoming and (2) outgoing links of a resource in the dataset; the (3) incoming (PAGE-IN) and (4) outgoing (PAGE-OUT) links of the resource's Wikipedia page; the (5) DBtrends rank presented in Section 4; the (6) DBpedia page-rank (DB-RANK) [23]; the (7) number of external incoming links to the resource's Wikipedia page (E-PAGE-IN); the (8) Page Authority measured by SEO (SEO-PA)[3]; the (9) Wikipedia Page-Rank (PAGE-RANK); the (10) MIXED-RANK a combination of DBtrends and PAGE-RANK ranking functions; the (11) social shared links (SHARED-LINKS); (12) the distance achieved by the best entity rank (r^e) combination applied to the entity rank samples (R^e), and; (13) the best (Best) entity rank (R^e). The E-PAGE-IN, SEO-PA, PAGE-RANK, SHARED-LINKS were measured by SEO review tool[4]. The best rank (Best) was evaluated based on the average position of the resource (class, property, entity) in the users profiles. In this evaluation we use the average one month Google Trends index to build the DBtrends rank. We also provide statistics such as median, average maximum, average distance as well as the standard deviation.

5.1 Results

The tables 1, 2 and 3 display the general results achieved by the evaluation of entities R^e, properties R^p and classes R^c ranks respectively. The tables contain (1) the average distance $\overline{D_R}(R, r)$ between the rank sample data R and different ranks r, (2) the standard deviation $\sigma_{D_F}(R)$, (3) the median $\widetilde{D_F}(R)$, (4) the average rank size $\overline{R}$ as well as (5) the average maximum distance of the samples per country $\overline{D_{R_{max}}}(R)$. The average confidence of each country in performing the tasks is displayed in Table 1. Considering R a rank set and r a rank, formally R, $D_F(R)$, $D_R(R, r)$ and $D_{R_{max}}(R)$ are defined as follows:

$$\begin{aligned} R =& \bigcup r \\ D_F(R) =& \{d_{D_F} \mid \forall r_\beta, r_\gamma \in R,\ d_{D_F} = D(r_\beta, r_\gamma)\} \\ D_R(R, r) =& \{d_{D_R} \mid \forall r_\beta \in R, d_{D_R} = D(r_\beta, r)\} \\ D_{R_{max}}(R) =& \{d_{D_{R_{max}}} \mid \forall r_\beta, r_\gamma \in R, d_{D_{R_{max}}} = D_{max}(r_\beta, r_\gamma)\} \end{aligned} \tag{7}$$

Table 1 displays the results of twelve different ranking functions applied to the entity rank sample data (R^e): (1) the incoming and (2) outgoing links of a resource in the dataset; the (3) incoming (PAGE-IN) and (4) outgoing (PAGE-OUT) links of the resource's Wikipedia page; the (5) DBtrends

[3] `https://moz.com/learn/seo/page-authority`
[4] `http://www.seoreviewtools.com/`

Country(R^e)		USA	India	AVG
	$\sigma_{D_F}(R^e)$	26.00	47.00	36.50
	$\overline{D_{R_{max}}}(R^e)$	203.00	311.68	257.34
	$\widetilde{D_F}(R^e)$	96.00	144.00	120.00
	$\overline{\lvert F(r^e)\rvert}$	20.00	16.33	18.16
	$Confidence(\%)$	0.55	0.90	0.72
$\overline{D_R}(R^e, r^e)$	DB-IN	104.02	143.87	123.95
	DB-OUT	104.95	127.41	116.18
	PAGE-IN	99.04	136.78	117.91
	PAGE-OUT	110.73	129.66	120.20
	DBtrends	97.33	139.02	118.17
	DB-RANK	109.80	144.50	127.15
	E-PAGE-IN	94.20	129.18	111.69
	SEO-PA	97.90	134.18	116.04
	PAGE-RANK	89.63	129.58	109.60
	SHARED-LINKS	101.97	126.75	112.36
	MIXED-RANK	86.86	129.58	108.22
	Best	83.06	126.38	104.99

Table 1: Average rank similarity for entities. The table presents the following statistics: (1) the average distance $\overline{D_R}(R^e, r^e)$ between the entity rank sample data r^e and different entity ranks R^e, (2) the standard deviation $\sigma_{D_F}(R^e)$, (3) the median $\widetilde{D_F}(R^e)$, (4) confidence, (5) the average rank size $\overline{\lvert F(r^e)\rvert}$ as well as (6) the average maximum distance of the samples per country $\overline{D_{R_{max}}}(R^e)$. The ranks includes: (1) the in and (2) out degree of a resource in the dataset; the (3) in (PAGE-IN) and (4) out degree (PAGE-OUT) of the resource's Wikipedia page; the (5) DBtrends rank presented in Section 4; the (6) DBpedia page-rank (DB-RANK); the (7) number of external links pointing to the resource's Wikipedia web page (E-PAGE-IN); the (8) Page Authority measured by SEO (SEO-PA); the (9) Wikipedia Page-Rank (PAGE-RANK); the (10) MIXED-RANK a combination of DBtrends and PAGE-RANK ranking functions (see Figure 2); the (11) social shared links (SHARES-LINKS); and the (12) Best entity rank (r^e) .

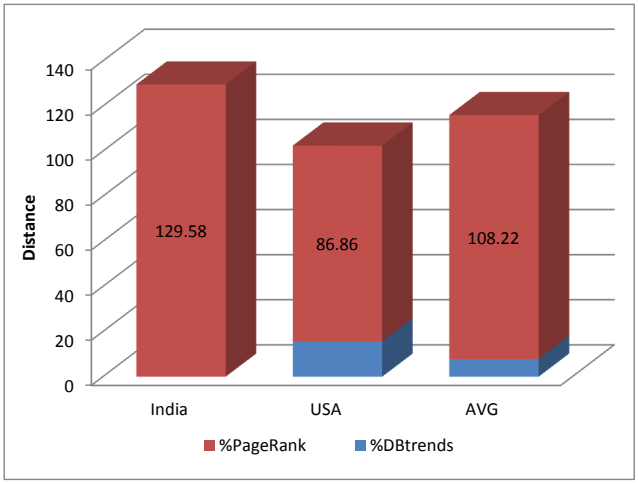

Figure 2: MIXED-RANK, a combination of the best performed entity rank (PAGE-RANK) and DBtrends.

rank presented in Section 4; the (6) DBpedia page-rank (DBRANK) [23]; the (7) number of external incoming links to the resource's Wikipedia page (E-PAGE-IN); the (8) Page Authority measured by SEO (SEO-PA)[5]; the (9) Wikipedia Page-Rank (PAGE-RANK); the (10) MIXED-RANK a combination of DBtrends and PAGE-RANK ranking functions; the (11) social shared links (SHARED-LINKS), and; (12) the distance achieved by the best entity rank (r^e) combination applied to the entity rank samples (R^e). The E-PAGE-IN, SEO-PA, PAGE-RANK, SHARED-LINKS were measured by SEO review tools[6].

The MIXED-RANK was measured by performing a refinement operator between DBtrends and PAGE-RANK. The refinement operator checked for all existing combinations between the two ranks. Figure 2 shows the best combination in percentage of DBtrends and PAGE-RANK for Indians (resp. 0% and 100%) and Americans (resp. 13.14% and 86.86%);

Table 3 shows the results of three different class ranks applied to the class sample data (R^c): the number of instances of the classes—the number of resource instantiations that are sub-type of the class—in the dataset in (1) descending (Instances↑) and (2) ascending order (Instances↓); as well as (3) the best (Best) class rank (r^c).

Table 2 shows the results of five different property ranks applied to the property sample data (R^p): the number of instances of the property in the dataset in (1) descending (Instances↑) and (2) ascending order (Instances↓); (3) the RELIN [3] property rank; as well as (4) the best (Best) property rank (r^p).

5.2 Discussion

The results in Table 1 show that, on average, Indians were ∼40% more confident than Americans in performing the

[5]https://moz.com/learn/seo/page-authority
[6]http://www.seoreviewtools.com/

Country(R^p)		USA	India	Comb.
	$\sigma_{D_F}(R^p)$	36.67	45.72	40.08
	$\overline{D_{R_{max}}}(R^p)$	170.00	163.97	167.13
	$\widetilde{D_F}(R^p)$	102.83	113.75	106.68
	$\overline{R^p}$	15.55	14.63	15.08
$\overline{D_R}(R^p, r^p)$	Instances↑	58.82	122.03	90.42
	Instances↓	60.39	127.59	93.99
	Relin	51.08	91.31	71.19
	RandomRank	44.00	83.20	63.60
	Best	44.57	76.78	60.68

Table 2: Measuring different property ranks. The table presents the following statistics: (1) the average distance $\overline{D_R}(R^p, r^p)$ between the property rank sample data S^p and different entity ranks R^p, (2) the standard deviation ($\sigma_{D_F}(R^p)$), (3) the median $\widetilde{D_F}(R^p)$; (4) the average maximum distance of the samples per country $\overline{D_{R_{max}}}s(R^p)$; and, (5) the average number of property of rank per country $\overline{R^p}$. The ranks in the table are: (1) the number of results returned by Google (G-Results); the number of instances of the property in the dataset in (2) descending order Instances↑ and (3) ascending Instances↓; (4) the RELIN property rank; as well as (5) the best (Best) property rank (r^p) .

Country(S^c)		USA	India	Comb.
	$\sigma_{D_F}(R^c)$	2.06	3.49	2.62
	$\overline{D_{R_{max}}}(R^c)$	11.35	15.88	12.96
	$\widetilde{D_F}(R^c)$	3.85	4.81	4.21
	$\overline{R^c}$	3.93	3.76	3.84
$\overline{D_R}(R^c, r^c)$	Instances↑	8.96	11.88	8.96
	Instances↓	4.04	8.54	6.29
	Best	4.14	8.39	6.27

Table 3: Measuring different class ranks. The presents the following statistics: (1) the average distance $\overline{D}(R^c, r^c)$ between the property rank sample data R^c and different entity ranks R^c, (2) the standard deviation $\sigma_{D_F}(R^c)$, (3) the median $\widetilde{D_F}(R^c)$; (4) the average maximum distance of the samples per country $\overline{D_{R_{max}}}(R^c)$; and, (5) the average number of classes of rank per country ($\overline{R^c}$). The ranks presented in the table are: the number of instances of the class in the dataset in (1) descending Instances↑ and (2) ascending Instances↓ order; as well as (3) the best (Best) class rank (r^c) .

ranking tasks (**RQ.1**). However, the internal agreement for entities was much higher for Americans (**RQ.2**). The median $\widetilde{D_F}(R^e)$ and average max distances $\overline{D_{r_{MAX}}}(R^e)$ among the entity ranks of Americans where respectively ∼44% and ∼35% higher than the Indians. The same pattern did not apply to the property ranks where the differences were not tangible. The Indians achieved an internal entity rank agreement ∼2% higher than Americans when comparing the average maximal distance $\overline{D_{R_{max}}}(R^e)$. The results also show that Americans found all the 20 entities relevant wheres Indians found 16 of them ($\overline{r^e}$).

Regarding the measured entity ranks, the MIXED-RANK achieved the best result and is followed closely by PAGE-RANK, SHARED-LINKS, E-PAGE-IN and SEO-PA (**RQ.3**). The use of pure query logs extracted from Goolge Trends (DBtrends) obtained the ninth position. The MIXED-RANK achieved an entity rank only ∼3% higher than the ideal rank (BEST). The idea behind MIXED-RANK is to use a mixture of global long-term users interest (PAGE-RANK) with a short-term interest (DBtrends). Long-term interest is used because in the Web, pages with high interest usually became authorities and thus, have more incoming links. However, the time to access and link the content may take time. As the search for information occurs before it be found and linked, it is possible to measure short-term interest by using query logs. DBtrends represent short-term interests because it is based on query logs.

The graph depicted in Figure 2 shows that the query log helped reduce the rank distance of Americans but had no influence in the result achieved by PAGE-RANK with Indians. Although globally the combination of both PAGE-RANK and DBtrends achieved the lowest distance, the result is not consistent enough to reach any further conclusion.

Moreover, the entity ranks using measures from external sources (e.g. PAGE-RANK, MIXED-RANK and SHARES-LINK) achieved better result than the ranks generated by the graph structure (e.g. DB-RANK, PAGE-IN and PAGE-OUT) (**RQ.4**) . We believe that this result is due to two facts: (1) first, the structure of a RDF knowledge graph does not necessary say something about the importance of a entity, although it can be used for properties and classes (Table 4) with relative success; (2) the importance of an entity is better estimated by its use in the real world than by the knowledge graph.

DBtrends rank uses statistics of query logs of Google available via Google Trends. However, indexes extracted from other sources such as PAGE-RANK, SHARED-LINKS and SEO-PA in Table 1 can also be used as a seed in the proposed ranking function.

An interesting observation (**RQ.2**) is that dbr:Animal is chosen as top first for 13 Americans, in contrast to merely four of the Indians. This also explains the occurrence of dbr:Lepidoptera as the most important entity for some users. However, this finding is not observed when comparing average results. For instance, dbr:Plant appears in first position for Indians and dbr:Animal at ninth, where for Americans dbr:Animal appears at second and dbr:Plant at fifth. Moreover, the top first results of the Americans are less sparse than for the Indians. The top first entity of the Americans is devised among eight entities against 13 of the Indians. Furthermore, (**RQ.3**) the results shown in Table 2 demonstrate that RandomRank is the best property ranking function, being only ∼ 5% higher than the best method (BEST). Moreover, RandomRank is followed closely by RELIN, a state-of-the-art method for property ranking.

Class	Distance	#Rank
dbo:PopulatedPlace	29.09	1
dbo:Settlement	28.72	2
dbo:Place	26.63	3
owl:Thing	24.54	4

Table 4: Average best rank for dbr:New_York classes. The table above displays the average class distance and rank for dbr:New_York classes.

The (**RQ.2**/**RQ.3**) results in Table 3 show that sorting of the classes by the number of instances in ascending order (Instances↓) is a very effective method for ranking classes. In the case of the American users, it even performs better than the average best rank (BEST). Another curious observation is that, by doing so, the hierarchy is reproduced from the less to the more generic class. Table 4 shows the best rank for dbr:New_York, where the hierarchy can be seen clearly. However, it is important to observe that the results differ from the results achieved by the entity ranking task. Different from the class rank, the results for entity rank show abstract concepts with lower distance, especially with North Americans—e.g. Plant (Rank 5) and Animal (Rank 2).

Finally, (**RQ.2**) the average rank similarity among the different users ranks (internal agreement) for entities, properties and classes are respectively ∼63%, ∼37% and ∼67%.

6. CONCLUSION, LIMITATIONS & FUTURE WORK

In this paper, we presented a function for ranking RDF data based on query logs. We presented an extension of the Speman's Footrule similarity ranking function to measure heterogeneous ranks and demonstrate that the proposed extension is more accurate than the previously introduced formula. We compared over fifteen ranking functions applied to RDF data (classes, properties and entities) of 60 users across two different countries. We showed that the proposed function can generate better ranks when combined with existing ranking functions. However, the result does not allow us to establish any conclusion of the different impact of the MIXED-RANK among countries. The evaluated results show that the use of ranks from external data sources is more efficient when ranking entities. The same result is not observed for classes and properties as they achieve better results when using an internal dataset rank. For instance, a simple approach of sorting classes by the number of instances can provide a very good class rank, while the best rank for entities is the PAGE-RANK value from the entity's Wikipedia web page. For future work, we plan to (1) further investigate the use of information coming from external sources and their combination as well as (2) extend the evaluation to other countries and ranking functions.

7. ACKNOWLEDGEMENTS

This work was supported by a grant from the EU H2020 Framework Programme provided for the projects Big Data Europe (GA no. 644564), HOBBIT (GA no. 688227), CNPq under the program Ciências Sem Fronteiras and by Intituto de Pesquisa e Desenvolvimento Albert Schirmer (CNPJ 14.120.192/0001-84).

8. REFERENCES

[1] M. Alsarem, P.-E. Portier, S. Calabretto, and H. Kosch. Ranking entities in the age of two webs, an application to semantic snippets. In *The Semantic Web. Latest Advances and New Domains*, volume 9088 of *Lecture Notes in Computer Science*, pages 541–555. Springer International Publishing, 2015.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International Conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[3] G. Cheng, T. Tran, and Y. Qu. RELIN: Relatedness and Informativeness-based Centrality for Entity Summarization. In *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I*, ISWC'11, pages 114–129, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] G. Cheng, D. Xu, and Y. Qu. Summarizing entity descriptions for effective and efficient human-centered entity linking. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 184–194. International World Wide Web Conferences Steering Committee, 2015.

[5] S. F. De Araújo and D. Schwabe. Explorator: a tool for exploring RDF data through direct manipulation. In *Linked data on the web WWW2009 workshop (LDOW2009)*, 2009.

[6] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. *The Semantic Web – ISWC 2005: 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005. Proceedings*, chapter Finding and Ranking Knowledge on the Semantic Web, pages 156–170. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[7] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, Dec. 2004.

[8] T. Franz, A. Schultz, S. Sizov, and S. Staab. TripleRank: Ranking Semantic Web Data by Tensor Decomposition. In *Proceedings of the 8th International Semantic Web Conference*, ISWC '09, pages 213–228, Berlin, Heidelberg, 2009. Springer-Verlag.

[9] A. Hogan, A. Harth, and S. Decker. ReConRank: A Scalable Ranking Method for Semantic Web Data with Context. In *In 2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.

[10] K. Järvelin. Cumulated gain-based evaluation of ir techniques. volume 20, page 2002, 2002.

[11] M. Kendall. *Rank correlation methods*. Griffin, London, 1948.

[12] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, Sept. 1999.

[13] R. Kumar and S. Vassilvitskii. Generalized distances between rankings. In *WWW*, WWW '10, pages 571–580. ACM, 2010.

[14] S. Lawrence. Context in web search. *IEEE Data Eng. Bull.*, 23(3):25–32, 2000.

[15] E. Marx, A. Zaveri, M. Mohammed, S. Rautenberg, J. Lehmann, A.-C. N. Ngomo, and G. Cheng. DBtrends : Publishing and Benchmarking RDF Ranking Functions. In *2nd International Workshop on Summarizing and Presenting Entities and Ontologies, Co-located with the 13th Extended Semantic Web Conference*, SUMPRE'16, 2016.

[16] N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 25–34, New York, NY, USA, 2011. ACM.

[17] A.-C. N. Ngomo and S. Auer. LIMES - a time-efficient approach for large-scale link discovery on the web of data. *integration*, 15:3, 2011.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[19] T. Penin, H. Wang, T. Tran, and Y. Yu. *The Semantic Web: 3rd Asian Semantic Web Conference, ASWC 2008, Bangkok, Thailand, December 8-11, 2008. Proceedings.*, chapter Snippet Generation for Semantic Web Search Engines, pages 493–507. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[20] A. Roa-Valverde and M.-A. Sicilia. A survey of approaches for ranking on the web of data. *Information Retrieval*, 17(4):295–325, 2014.

[21] C. Rocha, D. Schwabe, and M. P. Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 374–383, New York, NY, USA, 2004. ACM.

[22] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:88–103, 1904.

[23] A. Thalhammer and A. Rettinger. Browsing DBpedia entities with summaries. In *The Semantic Web: ESWC 2014 Satellite Events*, pages 511–515. Springer, 2014.

[24] V. N. Vapnik and V. Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[25] X. Zhang, G. Cheng, and Y. Qu. Ontology summarization based on rdf sentence graph. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 707–716, New York, NY, USA, 2007. ACM.

[26] Z. Zhuang and S. Cucerzan. Re-ranking search results using query logs. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 860–861, New York, NY, USA, 2006. ACM.