Towards a formal model for sharing and reusing ranking computations

Antonio J. Roa-Valverde
Semantic Technology Institute
University of Innsbruck
antonio.roa@sti2.at

Andreas Thalhammer
Semantic Technology Institute
University of Innsbruck
andreas.thalhammer@sti2.at

Ioan Toma
Semantic Technology Institute
University of Innsbruck
ioan.toma@sti2.at

Miguel-Angel Sicilia
University of Alcala
msicilia@uah.es

ABSTRACT

In this paper we present our efforts towards the design of vRank. The "Vocabulary for Ranking" allows to formally materialize ranking data algorithms. We justify the need for such a vocabulary and we show potential applications in the context of Linked Data consumption.

1. INTRODUCTION

A great part of the research in the field of semantic search focuses on proposing new ranking algorithms, while many of them are either adaptations to the well-known PageRank algorithm [2] or extensions of information retrieval techniques like TF-IDF [8].

Until now, users of ranking algorithms (mostly through search engines) do not have the possibility of deciding how they want the data to be ranked for them. Most of the services offered by data providers are not flexible and impose their relevance models to data consumers. This design policy presents an important inconvenience with regard to data consumption: what is relevant for a user might not be relevant for another one and vice versa. In this way, there is a need for a mechanism that allows data consumers to decide how they want the data to be ranked, as they know better than anyone else what their preferences for consuming the requested data are.

A possible way of implementing this mechanism is described through this paper. We propose vRank, the "Vocabulary for Ranking", that addresses the modeling of ranking approaches to provide an efficient and flexible vehicle for data consumption. We focus on the scope of the Web of Data, however we remark that vRank's practical applications are not restricted to the Linked Data domain.

The rest of this paper is structured as follows. In section 2 we justify the aim of vRank. Section 3 presents the different parts of the vocabulary. Section 4 describes the potential applications where vRank could be used. In section 5 we describe existent approaches which are similar to vRank. Finally, in section 6 we conclude this paper and enumerate future working lines. In the attached appendix we show an example of how to use vRank.

DBRank'12, in conjunction with VLDB, August 27, 2012, Istanbul, Turkey.

2. PURPOSE OF VRANK

The purpose of vRank is to provide data consumers with a standardized, formal, unambiguous, reusable and extensible way of representing ranking computations. The way data is consumed depends strongly on what is relevant for data consumers. When data providers offer a ranking service, obviously they cannot contemplate all possible relevance models of consumers. Therefore, the need for the functionality that vRank tries to implement is apparent. The following requirements have guided the design of vRank:

1. The need for unifying the way ranking algorithms are developed in order to promote reusability and evaluation.
2. The need for a common and accepted model to homogenize the exploitation of ranking services.
3. The need for isolating data from any kind of assumption regarding to publication and consumption (data providers and consumers may not share the same interests).

Offering ranking computations as part of the data can facilitate its consumption in several ways:

- Different relevance models computed by diverse ranking strategies can coexist within the same dataset. Consumers can adapt data requests to their relevance expectations.
- Data ranking becomes open and shareable. Consumers can reuse a specific way of ranking a dataset. If existent ranking approaches do not suit consumers' needs, they can extend the dataset with their own method.
- Consumers can reuse ranking scores in order to evaluate and compare different strategies over a given dataset.
- Consumers (and not data providers) can have control about how they want to consume data, giving more preference to what is more relevant.

In the following subsections we develop these aspects in more detail.

2.1 Ranking crystallization

Ranking algorithms rely on data structures that are used to compute the final scores of data items. Traditionally, these data structures are kept internal and inaccessible for the users. By using a service, data consumers can submit their queries and retrieve a list of results ordered according to the implemented relevance model. This kind of behavior defines ranking algorithms as a black box, which makes it very difficult, if not impossible, to reuse and share computations over existent data.

The relevance models computed by ranking algorithms need to be materialized in a way that can be offered publicly and and can be queried by data consumers. The publication can be done "easily" in RDF with a vocabulary that models the ranking domain. This is what vRank has been defined for.

SPARQL[1] is considered as the standard language for querying the Web of Data, however it does not support any kind of ranking apart from ORDER BY clauses. By adopting vRank it is not necessary to extend SPARQL with ranking support as the ranking can be made explicit within the dataset. Consumers do not need to learn a different query language or any kind of extension. They still can use ORDER BY clauses and just adapt their queries to use the according vRank triples (see appendix B).

2.2 Ranking evaluation

The development of ranking algorithms in the scope of Linked Data maintains a certain grade of parallelism with the evolution of the Semantic Web. While the first ranking algorithms were designed with the aim of ranking ontology documents [4], in a similar way to how traditional approaches rank HTML documents, the arrival of the Web of Data has changed the focus of ranking strategies towards information modeled as entities and their relationships [3]. In just a period of ten years the research on Semantic Web topics has delivered diverse kinds of works addressing the topic of ranking information following different approaches ranging from statistical analysis to link analysis.

Due to the different policies used in ranking is very difficult to establish a technical comparison to analyze the accuracy and precision of each algorithm in reference to others. While there are different benchmarks and data sets to measure and compare the ranking strategies in the area of information retrieval [1, 6, 9], the same is still missing when referring to ranking on the Web of Data. Authors in [7] justify the need for a benchmark[2] applied to search and ranking on the Web of Data and establish the first steps towards a methodology for resource retrieval evaluation.

One of the main contributions of vRank is that it helps to homogenize the way ranking services are exploited, so that third parties can compare and evaluate them.

2.3 Evolution of data

In an open environment like the Web, data is always going under modifications and revisions. When data is updated, the ranking scores associated to the data items have to be updated as well. A consumer may be interested in analyzing the ranking scores over the time in order to predict future changes that might affect her consumption patterns.

[1] http://www.w3.org/TR/rdf-sparql-query/

[2] The Semantic Search workshop series and the Semantic Search challenge can be considered as an ongoing effort towards this direction.

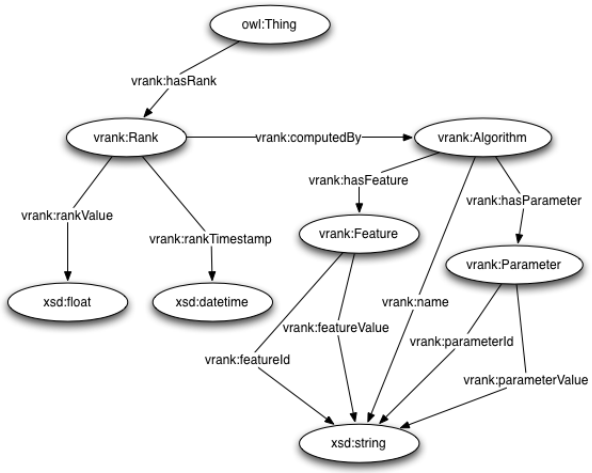

Figure 1: vRank overview

The mechanism implemented by vRank opens new possibilities for addressing the problem of measuring changes within data.

2.4 Multirelevancy

Consumers can make use of the available ranking scores to combine and compose their own ranking functions. This approach is addressed under what is known in the literature as ranking aggregation [5]. Following the same pattern, the newly obtained scores can be materialized and shared by using vRank.

3. VOCABULARY DESCRIPTION

The Vocabulary for Ranking (vRank) aims to model ranking information within datasets. We have tried to keep a simplistic design and therefore, we have reused existent vocabularies wherever possible. A full specification of vRank is available under the namespace *http://purl.org/voc/vrank#*. Figure 1 shows an overview of vRank. In the following, we describe the core components of the vocabulary.

3.1 Algorithm

In vRank an *Algorithm* is an entity that models metadata about a ranking implementation. The main purpose of this entity is to provide provenance information about the ranking scores. By knowing which settings have produced certain ranking scores, a data consumer can decide which ranking approach should be applied to the requested data.

In order to characterize certain algorithm, vRank allows the use of features and parameters.

3.1.1 Feature

A *Feature* complements the description of an algorithm in terms of its functionality. Features should be specified by the authors of the ranking approach with the aim of facilitating its understanding to data consumers.

As already mentioned, ranking algorithms are characterized by a diverse functionality, which in many cases is combined under the same implementation. This functionality could be used to classify the different algorithms. However, at the time of writing, we could not identify the existence of

a published taxonomy about ranking methodologies. In order to facilitate the adoption of vRank we consider necessary the existence of an initial set of features.

We are currently analyzing the state of the art in ranking in order to find a proposal. Some examples of features include "query dependency" (static ranking vs dynamic ranking) and "granularity" (document, predicate, entity, etc),

3.1.2 Parameter

A *Parameter* adds a finer level of description than a *Feature*. The main target of a *Parameter* is to capture the specific configuration of the algorithm that leads to the obtained ranking scores. An example of *Parameter* is the damping factor used by PageRank.

3.2 Rank

Rank is an entity that formalizes the ranking scores associated to a data item. Anything that can be model in RDF can have an associated *Rank*. The flexibility of the model resides on relating different instances of *Rank* with a particular data item.

A *Rank* by itself is meaningless. Therefore, *Rank*s are related to *Algorithm*s and to concrete executions (defined by specifying different *Parameter*s). In order to capture different executions with certain settings we have added a timestamp to the *Rank* entity.

3.3 Reuse of other vocabularies

In the following we list the components we have reused from other vocabularies.

- dublin-core: properties to model metadata such as creator, dataset's title and descriptions.
- *void:Dataset* should be use to describe the dataset containing the ranking instances.
- *foaf:homepage* of the ranking dataset's homepage.

4. APPLICATIONS

vRank has not been conceived to replace the current full text search model that has been established as the de facto mechanism for retrieving information on the Web. We are aware that most of nowadays data consumers are not experts and in many cases they have never used a query language like SQL or SPARQL before. vRank has been thought to be adopted by middleware solutions that implement ranking services and want to incorporate a flexible data consumption strategy in the process. We envision vRank to be used in data stores and search engines that make use of internal scoring mechanisms for presenting data to the user.

5. RELATED WORK

To the best of our knowledge, at the time of writing there are no similar approaches that try to solve the problem of modeling ranking scores in the same way as vRank does. The closest effort to vRank we have found has been developed by Ontotext[3] within the OWLIM RDF store[4]. OWLIM implements an internal ranking mechanism named RDF Rank[5] that extends the behavior of PageRank. RDF Rank scores are made available via the *#hasRDFRank* predicate, which is handled internally by OWLIM.

The biggest difference remains in the purpose of the model. In OWLIM the aforementioned predicate has been designed to be manipulated only internally and not to be shared publicly. Despite OWLIM offers the possibility of exporting RDF Rank values to an external file, their main purpose is to be used as initial activation values for the algorithm. In addition OWLIM does not contemplate the use of diverse ranking algorithms and therefore there is no implementation of this feature in the internal vocabulary.

[3] http://www.ontotext.com/

[4] http://owlim.ontotext.com/display/OWLIMv50/OWLIM+Primer

[5] http://owlim.ontotext.com/display/OWLIMv50/OWLIM-SE+RDF+Rank

6. CONCLUSIONS AND FUTURE WORK

In this paper we have described our efforts towards the design of the Vocabulary for Ranking (vRank). As an ongoing work, vRank is still in an initial stage and needs further contribution regarding to model refinement and evaluation in order to achieve a successful adoption by the community.

Regarding to model refinement, vRank can be extended to deal with query dependent algorithms. So far, vRank only contemplates algorithms that do not depend on user queries to compute the ranking scores. We are currently evaluating the tradeoff of including query information within the data model. Initial hypotheses indicate an overload of the model and a loss of generality that could affect the reusability of the ranking.

To evaluate the usefulness of vRank, we plan to model the results of running a pool of ranking configurations in a concrete domain like life sciences. The reason for choosing this domain is the amount of data available and the active community making use of it.

7. REFERENCES

[1] J. Artiles, S. Sekine, and J. Gonzalo. Web people search: results of the first evaluation and plan for the second. In *WWW*, 2008.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7), 1998.

[3] R. Delbru, N. Toupikov, M. Catasta, G. Tummarello, and S. Decker. Hierarchical link analysis for ranking web data. *ESWC. LNCS*, 6089, 2010.

[4] L. Ding, T. Finin, A. Joshi, R. Pan, S. R. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. *CIKM*, 2004.

[5] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. *WWW*, 2001.

[6] J. Kamps, S. Geva, A. Trotman, A. Woodley, and M. Koolen. Overview of the inex 2008 ad hoc track. advances in focused retrieval. In *International workshop of the Initiative for the Evaluation of XML Retrieval*, 2008.

[7] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object ranking in the web of data. *World Wide Web Conference*, 2010.

[8] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5), 1988.

[9] I. Soboroff, A. de Vries, and N. Craswell. Overview of the trec 2006 enterprise track, 2006.

APPENDIX

A. DATASET EXAMPLE

The following shows an example of how to use vRank to rank existing datasets. The example is itself a dataset containing three entities with two rankings each. Each of the rankings associated with an entity have been computed by two different algorithms, namely TripleRank and Ding.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix void: <http://rdfs.org/ns/void#> .
@prefix vrank: <http://purl.org/voc/vrank#> .
@prefix : <http://example.com/data#> .


<http://example.com/data#> .
    a void:DatasetDescription ;
    dc:title "An example of usage for
         the Vocabulary or Ranking"@en ;
    dc:creator "Antonio J. Roa-Valverde" ;
    foaf:primaryTopic :Data .

:Data
    rdf:type void:Dataset ;
    foaf:homepage <http://example.com/data> ;
    dc:title "Ranking Data Example" ;
    dc:description "Ranking data example computed over
         the Semantic Web Dog Food dataset."@en .

<http://data.semanticweb.org/conference/eswc/2012/
research-track-committee-member>
    vrank:hasRank :r1 ;
    vrank:hasRank :r4 .

<http://data.semanticweb.org/conference/eswc/2011/
programme-committee-member>
    vrank:hasRank :r2 ;
    vrank:hasRank :r5.

<http://xmlns.com/foaf/0.1/Person>
    vrank:hasRank :r3 ;
    vrank:hasRank :r6 .

:a1
    a vrank:Algorithm ;
    vrank:hasName  "TripleRank";
    vrank:hasFeature :f1 ;
    vrank:hasParameter :p1;
    dc:creator "Thomas Franz" ;
    dc:creator "Antje Schultz" ;
    dc:creator "Sergej Sizov" ;
    dc:creator "Steffen Staab" .

:a2
    a vrank:Algorithm ;
    vrank:hasName "Ding" ;
    vrank:hasFeature :f2 ;
    vrank:hasParameter :p2;
    dc:creator "Nickolai Toupikov" ;
    dc:creator "Juergen Umbrich" ;
    dc:creator "Renaud Delbru" ;
    dc:creator "Michael Hausenblas";
    dc:creator "Giovanni Tummarello" .

:f1
    a vrank:Feature ;
    vrank:featureId "Query dependency" ;
    vrank:featureValue "Static ranking" .

:f2
    a vrank:Feature ;
    vrank:featureId "Granularity" ;
    vrank:featureValue "Entity" .

:p1
    a vrank:Parameter ;
    vrank:parameterId "iterations" ;
    vrank:paramValue "500" .

:p1
    a vrank:Parameter ;
    vrank:parameterId "damping factor" ;
    vrank:paramValue "0.15" .

:r1
    a vrank:Rank;
    vrank:hasRankTimestamp "2012-05-17T20:10:00"^^xsd:datetime;
    vrank:rankValue 0.66;
    vrank:computedBy :a1 .

:r2
    a vrank:Rank;
    vrank:hasRankTimestamp "2012-05-17T20:10:00"^^xsd:datetime;
    vrank:rankValue 0.75;
    vrank:computedBy :a1 .

:r3
    a vrank:Rank;
    vrank:hasRankTimestamp "2012-05-17T20:10:00"^^xsd:datetime;
    vrank:rankValue 0.21;
    vrank:computedBy :a1 .

:r4
    a vrank:Rank;
    vrank:hasRankTimestamp "2012-06-01T20:16:00"^^xsd:datetime;
    vrank:rankValue 0.91;
    vrank:computedBy :a2 .

:r5
    a vrank:Rank;
    vrank:hasRankTimestamp "2012-06-01T20:16:00"^^xsd:datetime;
    vrank:rankValue 0.44;
    vrank:computedBy :a2 .

:r6
    a vrank:Rank;
    vrank:hasRankTimestamp "2012-06-01T20:16:00"^^xsd:datetime;
    vrank:rankValue 0.31;
    vrank:computedBy :a2 .
```

B. QUERY EXAMPLE

B.1 Q1

Select all the items in the dataset having a rank score computed by TripleRank and sort them in descendent order.

```
PREFIX vrank: <http://purl.org/voc/vrank#>
PREFIX ex: <http://example.com/data#>
SELECT * WHERE {
  ?x vrank:hasRank ?rank .
  ?rank crank:computedBy <http://example.com/data#a1>
  ?rank vrank:rankValue ?score .
}
ORDER BY DESC(?score)
```

B.2 Q2

Select all the scores associated to foaf:Person.

```
PREFIX vrank: <http://purl.org/voc/vrank#>
PREFIX ex: <http://example.com/data#>
SELECT ?rank ?algorithm ?score WHERE {
  <http://xmlns.com/foaf/0.1/Person> vrank:hasRank ?rank .
  ?rank vrank:computedBy ?algorithm .
  ?rank vrank:rankValue ?score .
}
```

C. ACKNOWLEDGMENTS

We are grateful for many helpful suggestions from Dieter Fensel and the anonymous referees.