xhRank: Ranking Entities on the Semantic Web

Xin He[1] and Mark Baker[1]

[1] School of Systems Engineering, University of Reading, Whiteknights,
Reading, Berkshire, RG6 6AY, UK
{x.he, mark.baker}@reading.ac.uk

Abstract. In general, ranking entities (resources) on the Semantic Web (SW) is subject to importance, relevance, and query length. Few existing SW search systems cover all of these aspects. Moreover, many existing efforts simply reuse the technologies from conventional Information Retrieval (IR), which are not designed for SW data. This paper proposes a ranking mechanism, which includes all three categories of rankings and are tailored to SW data.

Keywords: Semantic Web, ranking, RDF, entity, resource, ontology.

1 Introduction

Semantic Web (SW) querying in general involves matchmaking, graph exploration, and ranking, which form a process pipeline. Existing approaches to ranking SW entities (resources) can be categorised into three types, based on importance, relevance, and query length respectively. Importance-based rankings [2, 4, 5, 6] rank the importance of SW resources, such as classes, instance resources and properties. Relevance-based rankings [2, 4, 5, 6] match keywords to SW resources. These approaches are purely based on word occurrence, and do not taken into account word order and dispersion in literal phrases. Query length-based rankings [6] rank resource by following the idea that shorter queries tend to capture stronger connections between key phrases. However, we rarely see ranking schemes used in existing SW search engines that cover all of these aspects. In addition, although Information Retrieval (IR) and web algorithms, such as *PageRank* and *TF-IDF* have been adapted for application in some SW search engines, we argue that they can still be further improved to be better suited for SW data.

Therefore, by analysing the limitations presented in existing research efforts and considering the specific way that SW data is stored, this paper proposes an approach, namely *xhRank*, to ranking SW resources. This includes relevance, importance, and query-length based rankings, all of which are particularly designed for SW data.

2 The xhRank Approach

In SW resource searching, there are in general three situations, in which a user input may match an instance resource that the user intends to find (*Target Resource*):

- (1) Only the target resource is matched. The user-input keywords uniquely match

with the literals that directly describe the target resource. In this case, the user intends to find a resource by providing its most direct annotations.

- (2) The target resource and its forward neighbouring resources are matched. The user-input keywords match not only the literals that directly describe the target resource, but the literals that describe its forward neighbours. These neighbours represent the attributes of the target resource. In this case, the user intends to find a resource by providing its most direct annotations as well as information about some attributes of the resource that is known to the user.
- (3) Only forward neighbouring resources of the target resource (but not the target resource itself) are matched. The user-input keywords match the literals describing the forward neighbours of the target resource, but not the literals describing the target resource itself. In this case, the user intends to find a resource by providing information about some attributes of the resource that is known to the user.

In xhRank, all these situations are covered in the overall ranking, which is a summation of the relevance, importance, and query-length rankings, as presented below.

2.1 Relevance-based Ranking

Phrase-level Ranking. xhRank employs an alternative phrase ranking approach to the word occurrence-based approach used by most existing SW search systems. In addition to syntactical similarity, our approach takes into account term order and dispersion. The degree of similarity of a phrase (*Key Phrase*) to another phrase (*Target Phrase*) is determined by a phrase, called *Related Key Phrase*, extracted from the key phrase, in which each word corresponds to a word in the target phrase and in which the term order is compliant with the target phrase. For example, given the key phrase “Audrey Hepburn Hollywood Actress” and the target phrase “Audrey Hepburn was a Belgian-born, Dutch-raised actress of British and Dutch ancestry”, the related key phrase is “Audrey Hepburn Actress”. It should be noted that there may be more than one such related key phrase exists for a key phrase - target phase pair.

In the context of SW query, a key phrase refers to a phrase extracted from the user input, whilst a target phrase refers to the value of a literal. Instead of returning an overall score as the result, the resulting related key phrases (*Phrase Similarity Result*) are returned, with each word in the related key phrases represented by its position in the key phrase, in conjunction with a rating value for that word. Each word in the related key phrase is rated according to the (1) Syntactical similarity *S*: the similarity score between the keyword and the corresponding word in the target phrase; (2) Importance of the keywords *I*: specified by the user; (3) Normalisation ratio *N*: used to normalise the related key phrase by the length of the literal. The higher the ratio of words in the key phrase to words in the target phrase, the more valuable these words are; and (4) Discontinuous weighting *D*: The more times the words in the related key phrase are divided by the non-related words, the less valuable these related words are.

Graph-level Ranking. The graph mentioned here is the resulting graph from a graph exploration process. The node where the graph exploration initiated is called *Central Node*, which is by design related to the user input, and the graph itself is called *Context Graph*. Graph-level ranking is to compute the relevance of the central node to

the user input, which is subject to all resources within the context graph whose literals are related to the user input. Each of such resources is called a *Related Node*.

The relevance of a graph to a user input is calculated based on how well the user-input key phrases are covered by the literals related to the user input. By assembling the phrase similarity results (each of which is obtained by the phrase-level ranking against a key phrase - related literal pair), all possible coverage against a key phrase is obtained. The relevance score is thus computed subjects to the best coverage result.

2.2 Importance-based Ranking

Resource (Node) Ranking. xhRank employs a variation on *ReConRank* [2] to rank the importance of resources. ReConRank (employed in *SWSE* [3]) is altered from the well-known *PageRank*/*HITS* algorithms. xhRank further improves on it by executing the ranking based on a complete graph (at global scale) and prior to query time.

Property (Edge) Ranking. In xhRank, the importance of SW property resources in RDF graphs (as edges) is dependent on the cost of that property. This is a prerequisite of the query length-based ranking, and is uniquely applied to the properties describing instance resources. The cost of a property *P* in the unit-graph [1] of a resource *A* is determined by the popularity of *P* among all instance resources of class *C*, where *A* is an instance of *C*. Thus, each property is ranked against a class.

2.3 Query Length-based Ranking

In xhRank, the query length-based ranking is used to evaluate a node (target node) within a graph (context graph) against a user input. The target node is evaluated based on the semantic distance between the target node and each of the nodes (related node) within the context graph that is related to the user input.

2.4 Overall Ranking

Overall ranking extends the graph-level (relevance) ranking by complementing it with importance and query-length based rankings. The input to the ranking process is a list of explored graphs generated by the graph exploration process (a process prior to ranking). Each explored graph has a related node as its root. Thus, overall ranking is performed against each of these explored graphs (as the context graph) and against a node within the graph (as the target node). In the three situations discussed above, in situation (1) and (2), the target node is just the root node of the explored graph, which is also a related node. However, in situation (3), the target node is not a related node, but the “super-node” (backward neighbour) of all related nodes within the context graph. Thus, for each explored graph, in addition to the root node, the *Top Node* is also selected as a target node. A top node of an explored graph is the node, from which all related nodes can be navigated to by means of only following forward links.

In addition, there are a few more points to note:

– Although explored graphs are strictly hierarchical, there can still be more than one top node in an explored graph. In this case, only the top node with the closest overall distance to the related nodes is selected.
– Top node strategy is applied only when there is more than one related node in the explore graph, which would otherwise fall into situation (1).
– Non-root related nodes in an explored graph are not selected as target nodes.

Therefore, in order to incorporate query-length based ranking into the graph-level (relevance-based) ranking, when performing graph-level ranking, prior to the related key phrases being assembled, the rating value for each keyword position is multiplied by the reciprocal for the cost of the path from the target node to the related node that is described by that literal. In order to introduce the importance-based ranking to the graph-level (relevance-based) ranking, the importance of each resource node and the cost of each property is applied to the graph-level ranking. Hence, the overall ranking of a target node against a user input is obtained. Consequently, the overall ranking value of all target nodes are ordered, and the best *K* results are returned to the user.

It should be noted that graph explorations are performed based on the SW data, which includes all semantic relations that have been deduced from the corresponding ontologies prior to query time. Therefore, by interpreting the three situations (by means of following the semantic links) all semantics of the SW data are discovered.

3 Conclusions

In this paper, a ranking approach, namely xhRank, is proposed, which is tailored to the nature of SW data, in particular, the three possible situations in SW resource searching. The phrase-level (relevance-based) ranking provides a means to compute the similarity between two phrases by considering term relevance, position, and dispersion, which is believed more accurate than pure word occurrence-based approaches. The introduction of the importance and query length-based rankings to the graph-level (relevance-based) ranking further improves the ranking accuracy.

References

1. He, X. et al: A Graph-based Approach to Indexing Semantic Web Data. In: Proc. 9th ISWC, Poster and Demo Session (2010)
2. Hogan, A. et al: ReConRank: A Scalable Ranking Method for Semantic Web Data with Scalable Ranking Method for Semantic Web Data with Context. In: Proc. 2nd SSWS (2006)
3. Hogan, A. et al: Towards a Scalable Search and Query Engine for the Web. In: Proc. 16th WWW, Poster Session, pp. 1301--1302 (2007)
4. Zhang, L. et al: Semplore: An IR Approach to Scalable Hybrid Query of Semantic Web Data. In: Proc. 6th ISWC+ASWC2007, pp. 652--665. LNCS, vol. 4825, pp. 652--665, (2008)
5. Cheng, G. et al: Searching and Browsing Entities on the Semantic Web. In: Proc. 17th WWW, Poster Session, pp. 1101--1102, (2008)
6. Wang, H. et al: Q2Semantic: A Lightweight Keyword Interface to Semantic Search. In: Proc. 5th ESWC. LNCS, vol. 5021, pp. 584--598, (2008)