xhRank: Ranking Entities for Semantic Web Searching

Xin He
School of Systems Engineering
University of Reading
Reading, United Kingdom
x.he@reading.ac.uk

Mark Baker
School of Systems Engineering
University of Reading
Reading, United Kingdom
mark.baker@computer.org

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

Keywords-semantic web; ranking; RDF resource; semantic search; query

I. INTRODUCTION

Semantic Web (SW) querying, in generally, involves match making, 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 [1, 2, 3, 4] rank the importance of SW resources, e.g. classes, instance resources and properties. Relevance-based rankings [1, 2, 3, 4] match keywords to SW resources. These approaches are purely based on word occurrence, and do not take into account word order and dispersion in literal phrases. Query length-based rankings [4] 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 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 a ranking approach, namely xhRank [5]. This is a part of a SW search engine that we have developed, and is used for ranking SW resources. All relevance, importance, and query-length based rankings are included in our approach. Our experiments demonstrate that this approach is effective and that the ranking results are compliant with human perceptions.

The rest of the paper is organised as follows. We start in Section 2 with an overview of the three situations that may occur in SW searching. Section 3 introduces the xhRank approach to ranking RDF resources on the SW. This includes all relevance, importance, and query length based rankings. The evaluation of our approach is provided in Section 4. We then discuss related work in Section 5 and conclude in Section 6.

II. THE SCENARIOS IN SW SEARCHING

In SW resource searching, there are in generally 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 also 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.

III. THE XHRANK APPROACH

In xhRank, all these situations mentioned in Section 2 are covered in the overall ranking, which is a summation of the relevance-based, importance-based, and query length-based rankings, as presented below.

A. Relevance-based Ranking

Relevance-based ranking includes Term-level, Phrase-level, and Graph-level rankings, as detailed below:

1) Term-level Ranking

In xhRank, the similarity between two terms are computed based on the Levenshtein Distance or Edit Distance algorithm, which is by default supported by the Fuzzy Search functionality of Apache's Lucene [6]. According to the algorithm, the similarity between two terms (two strings) is computed depending on the minimum number of operations, e.g., an insertion, deletion, or substitution of a single character, needed to transfer one term into another.

2) 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. Figure 1 illustrates a comparison example between word-occurrence and xhRank based rankings.

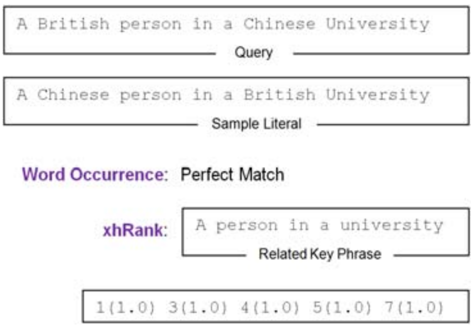

Figure 1. A comparison between word-occurrence based and xhRank based rankings

In this case, intuitively, xhRank's phrase level ranking is more reasonable than simply counting the word occurrence. Based on word occurrence, the key phrase and target phrase in Figure 1 are perfectly matched (all the seven keywords are related). However, based on human perception, we know that the query will return the wrong person in the wrong University. However, what has actually been matched is "a person in a University". In xhRank, the system finds that only five terms are related.

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.

It should be noted that somewhat complicated algorithms are required to enable such rankings. Thus, in many cases, this technique requires more computational resources than word-occurrence based rankings. The complexity of the computation is highly dependent on the length of the target phrase. Therefore, this approach favours relatively short target phrases. It would be very costly to implement this approach on a web search, in which target phrases refer to web documents. However, in the SW paradigm, target phrases refer to literals, which are normally very short in length (in most cases less than five words). Therefore, this approach is particularly suitable for searching the SW.

3) Graph-level Ranking

This computes the degree of relevance of a graph against a user input. The graph mentioned here is the resulting graph from a graph exploration process. The node where the graph exploration initiated is called the Central Node, which is by design related to the user input, and the graph itself is called a Context Graph. Graph-level ranking is used 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 these resources is called a Related Node. For example, in Figure 2, graph A is the context graph of target node R_2. L_7, L_8 and L_9 are literals related to the user input. R_3 and R_4 are therefore related nodes.

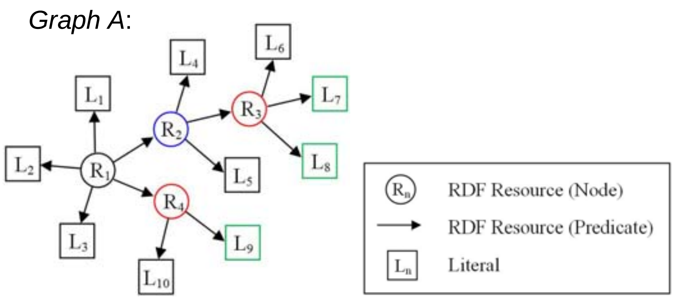

Figure 2. An example of the Context Graph of a Target Graph

The relevance of a graph to a user input is subject to the literals that are related to the user input. As related literals only describe related nodes, in other words, the relevance of a graph against a user input is subject to all related nodes within the graph. Apart from the central node, which is always a related node, these related nodes may also appear as neighbouring nodes within the context graph.

The relevance of a graph to a user input is calculated based on how well the user-input key phrases are covered by the related literal phrases within the graph. It leverages the results of phrase-level ranking, known as Phrase Similarity Result, which is a group of related key phrase lists. Each list consists of a number of elements, each of which is a keyword position and relevance score combination. Thus, against each key phrase, if there is more than one node

 ISBN: 978-1-61208-175-5

related, there may be more than one possibility of coverage as the result. By assembling these related key phrase lists for the related literal phrases, all possible coverage against a key phrase is obtained. The relevance score against a key phrase is thus computed subjects to the best coverage result. For example, Figure 3 illustrates how two related key phrase lists are assembled.

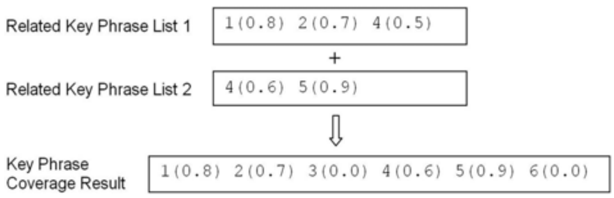

Figure 3. An example of assembling two related key phrase lists

The phrase similarity results (for all related literals) are then assembled. Figure 4 illustrates how phrase similarity results are assembled.

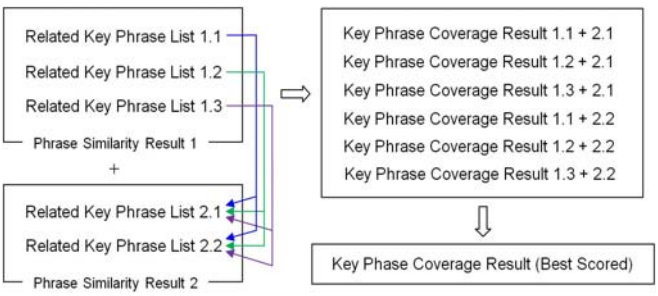

Figure 4. An example of assembling two phrase similarity results

A score against each key phrase coverage result is then calculated based on the average score of each position. The highest score among all key phrase coverage results is selected as the relevance score of the graph to that key phrase. Hence, the overall relevance score for the whole user input (including all key phrases) is calculated as the average relevance score for each key phrase.

B. Importance-based Ranking

This includes ranking the importance of SW class and instance resources (as nodes) and SW property resources (as edges) in RDF graphs.

1) Resource (Node) Ranking

The quality of resource importance rankings (based on linkage structure) depends heavily on how well the graphs and the contained RDF resources are interlinked. The ideal situation is that all resources and graphs are semantically interlinked with all related resources and graphs on a global scale, thereby forming a comprehensive graph for ranking. However, as our experiments are conducted against individual RDF datasets, resources are only linked within datasets. This will dramatically influence the ranking results. Therefore, importance ranking for SW resources is not implemented in our current experiments.

However, we still consider a variation on ReConRank [1] (the ranking approach as used in SWSE [7]) has the potential to offer an effective approach for ranking the importance of SW resources. ReConRank is a PageRank-like approach, which interconnects both resources and documents into one graph using semantic links and ranks resources based on that graph. The limitations of ReConRank are: First, the computation of the linkage-structure ranking is subject to incomplete graphs (the nodes that are related to the user input), which affects the query accuracy; Second, the ranking is performed at query time, thus affecting query speed. Therefore, by executing ReConRank-like ranking based on a complete graph (at global scale) and prior to query time, the ranking of resources' importance can be efficiently executed.

2) Property (Edge) Ranking

The importance of each property is ranked dependent on the cost of that property. This is a prerequisite of query length-based ranking, and is only applied to the properties that describe instance resources. In xhRank, the cost of a property *P* in the unit-graph 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. The cost of *P* against *C* is calculated using equation (1), in which |*property*| is the number of *P* found among the instances of *C*, and *N* is the total number of instances of *C*. This is similar to the approach employed in Q2Semantic [4]. It applies to all properties including those connected with blank nodes in both directions. The lower the cost of a property, the more important the property is.

$$Cost_{p-c} = 2 - \log_2\left(\frac{|property|}{N} + 1\right) \quad (1)$$

C. Query Length-based Ranking

In xhRank, in general, query length-based rankings are used to evaluate a node (Central Node) within a graph (Context Graph) against a user input. Thus, the target node is evaluated based on the semantic distance between the target node and each of the nodes within the context graph that is related to the user input (Related Node). (See Section III A 3.)

By assuming each edge in the context graph has the same importance, the ranking score of a target node is computed as the average length of each path between the target node and a related node.

xhRank also provides an option to weight backward links lower than forward links, by altering the value of a factor called BackwardLinkRate (BLR), which is a positive number in the interval (0, 1). Hence, by considering both the importance of edges and the *BLR* factor, a target node is evaluated using equation (2), in which p_i is a path between the target node and a related node, *e* is an edge in p_i, and *n* is the quantity of such paths.

 ISBN: 978-1-61208-175-5

$$E_{t-n} = \frac{\sum_{i \in (1,n)} \left(\sum_{e \in p_i} Cost_e \times BLR \right)}{n} \quad (2)$$

D. 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 following only forward links.

In addition, there are a few 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) ranking, when performing the graph-level ranking, prior to the related key phrase lists being assembled, the relevance score for each keyword position is multiplied by the reciprocal of the cost of the path from the target node to the candidate resource 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.

IV. EVALUATION

We have developed a keyword-based semantic search system to demonstrate and evaluate our ranking approach. As there is currently no standard benchmark for evaluating searching against the SW, we select real world RDF datasets for our experiments. Our selection criteria are, we select RDF datasets that (1) are well known; (2) are in use; (3) are of different size; and (4) have different usage and purposes.

Based on these criteria, the datasets selected for our experiment are given below.

- myExperiment [8]
- the Lehigh University Benchmark (LUBM) (50) [9]
- DBLP (RKB Explore) [10]

(Although LUBM is a benchmark dataset, it effectively represents complicated RDF structures, and is valuable for evaluating the searching accuracy on relation based resource queries.)

We evaluate our ranking approach in terms of the system effectiveness (the accuracy of searching).

The ultimate result of the proposed semantic framework in this research will be the ranking of the available resources, indicating which is the best match, which is the next best and so on. Therefore, the objective of the effectiveness evaluation experiments is to show that the resultant matchmaking and rankings computed by the system agree reasonably well with human perception for the same situation.

A detailed study about existing effectiveness evaluation approaches has been conducted in [11], in which two basic conclusions have been drawn:

(1) There are no agreed, best practice evaluation methods that can be used to evaluate semantic matching solutions.

(2) The precision and recall metrics used in conventional IR domain cannot be directly applied to measure effectiveness of systems that return a fuzzy value for the relevance. They are only applicable to systems that return a Boolean relevance.

Therefore, we have adopted the Generalised Measures of Precision and Recall employed in [11] to evaluate their system effectiveness.

Our experiments have been carried out against the selected datasets. In line with the typical situations discussed in Section 2, we have selected six query examples, two examples for each situation, to demonstrate how the system effectively retrieves results in different scenarios. These results have been compared with human perceptions.

Participants have been selected for the human participant studies. Our selection criteria are shown below: We select human participants (1) in different age range (from 25 to 50); (2) of both male and female gender; (3) who have excellent English reading skill; (4) with different backgrounds (eastern and western); (5) with different expertise (IT including people from the Semantic Web community, Mechanical engineering, Business, Finance, Accounting, Food industry etc.)

The aim is to minimise biasing results by selecting a cross-section of participants.

For each query case, the user input (the keywords) is provided, followed by an explanation of what exactly the user intends to find through the query.

Top five-scored results of each query are selected for the participants to rank. These results are given in random order. The original order computed by our system is hidden to the participants. Each result is shown by a diagram illustrating the semantic relations between the matched resource and its neighbours. For the sake of simplicity, each resource (a node) is represented using the literal values (including the label values of the corresponding datatype properties) that describe the resource. Each object property (an edge connecting two resources) is represented using its label values. There is also an explanation of the diagram followed in the next page, which help the participants to capture the semantic meanings of the result.

It should be noted that each result selected for the human participant studies are scored differently from others. Where results have the same score, we randomly select one from them for the study. This is because, as our ranking system is very sensitive, query results with the same score usually have the same semantic relation structure, and have exactly the same matches to the keywords. There is little value in the participants ranking these results in order to investigate the effectiveness of our system. However, studying results with differing scores generated by our system makes it relatively straightforward to discover how accurately our system ranks the query results with different similarities to user requirements. In practise, the top-k results will be returned to the user.

The query cases are given in Table 1.

The comparisons of the system rankings and average human rankings of the query results for each query case are stipulated in Table 2.

TABLE II. COMPARISONS OF SYSTEM AND HUMAN RANKINGS FOR QUERY RESULTS

Query Results (Ranked by System)	Average Human Ranking
1	1.36
2	3.09
3	3.00
4	3.45
5	4.09

(a) Query 1

Query Results (Ranked by System)	Average Human Ranking
1	1.27
2	3.73
3	3.18
4	3.27
5	3.55

(b) Query 2

Query Results (Ranked by System)	Average Human Ranking
1	1.27
2	3.09
3	2.45
4	3.73
5	4.45

(c) Query 3

Query Results (Ranked by System)	Average Human Ranking
1	1.64
2	1.45
3	3.27
4	4.09
5	4.54

(d) Query 4

Query Results (Ranked by System)	Average Human Ranking
1	1.18
2	2.81
3	3.54
4	3.64
5	3.81

(e) Query 5

Query Results (Ranked by System)	Average Human Ranking
1	2.00
2	2.90
3	3.36
4	2.45
5	4.27

(f) Query 6

TABLE I. QUERY CASES

Query	Scen	Keywords	User intends to find
Q1	1	matchmaking rank, semantic web, volume1	A publication. The title includes keywords “matchmaking”, “rank”, and “semantic web”. It is published in “Volume 1” (of a Journal, for example).
Q2	1	Constraint Normal Logic Programming, Functorial Framework	A publication, which includes keyword phrases “Constraint Normal Logic Programming”, and “Functorial Framework”.
Q3	2	Applications of Membrane Computing, Gabriel Ciobanu 2006	A publication. The title includes key phrase “Applications of Membrane Computing”. It is related to a person called “Gabriel Ciobanu”. The publication year is “2006”.
Q4	3	Yanchun Zhang Jinli Cao 2003	A publication between two people, called “Yanchun Zhang” and “Jinli Cao” respectively. This is published in 2003.
Q5	2	AssociateProfessor9 GraduateCourse30 Publication6	A person called “AssociateProfessor9”, who is related to a graduate course, called “GraduateCourse30”, and a publication entitled “Publication6”.
Q6	3	Department20 University3 Course47 GraduateCourse44	A person in a department, called “Department20” at a University, called “University3”. This person is related to an (undergraduate) course, called “Course47”, as well as a graduate course, called “GraduateCourse44”.

The resulting generalised measures for the precision and recall against each query case are stipulated in Table 3.

TABLE III. THE PRECISION, RECALL, AND F-MEASURE FOR THE QUERY CASES

Query Case	Situation	Precision	Recall	F-measure
1	1	0.855	0.854	0.855
2	1	0.782	0.782	0.782
3	2	0.864	0.863	0.864
4	3	0.900	0.899	0.899
5	2	0.847	0.845	0.846
6	3	0.774	0.773	0.773

It should be noted that there are a number of issues that affect the participants' rankings in our experiments, as presented below.

(1) The participant's level of understanding of the Semantic Web structure. During our human participant studies, we have found that enabling ordinary users to gain an understanding Semantic Web concepts and operations presents a significant challenge. Most people are used to conventional means of gathering information, in which all retrieved data of a search result is presented in a single node (e.g., a web page). Many of the participants find it difficult to comprehend why we return a single node as a matched result, rather than the full picture shown to them. Further, in some scenarios, they may have trouble understanding why a resource is regarded as a matched resource, even if the text describing the node contains none of the keywords, whilst in other cases, resources that contain matched texts are not selected, for example the result1 of Query scenario4. This causes some confusion for users. We have tried to explain the Semantic Web as a large knowledge base. However, it seems that this explanation is still not very helpful for some participants. As the Human-Computer Interaction (HCI) implications of the Semantic Web are not the focus of this research, we have accepted that the experience for users may not be completely intuitive. Nonetheless, we have gained a deeper understanding of how significant the HCI is, and how important good interfaces are in helping ordinary people to become consumers of the Semantic Web, and in enabling them possibly to contribute to it.

(2) Familiarity with the context of the subject matter. In all six query scenarios, we require participants to rank the results according to semantic meanings rather than syntactical similarities or word occurrences. This requires the participants to understand the meanings of the textual information to a certain extent. For example, in query scenario1 and scenario2, the user intends to find a publication in the Computer Science (CS) or Artificial Intelligence (AI) domain. Some participants are not familiar with scientific phraseology, and have problems interpreting the exact meaning of the titles of publications.

(3) Human common sense does not apply. Most datasets used in the Semantic Web community are still isolated with limited inter-connections, and are mainly used for research purposes. Therefore, common sense judgements are not normally applicable to this data. For example, xhRank rankings are in part based on popularities of resources and properties. In query scenario 4, the result1 and result2 have exactly the same similarity based on relevance and query-length. The system ranks result1 over result2 because of the importance of the resources. Result1 belongs to class "Book Section Reference", whereas result2 belongs to class "Article Reference". In the DBLP dataset, there are 780,998 instances of Book Section References and 495,071 instances of Article References. Thus, an instance of class "Book Section Reference" has higher importance than an instance of class "Article Reference". However, this information is hidden to the participants, and they are unable to use common sense to interpret the rationale behind the rankings. In real-world searches, when a user searches for keyword "No.7" in amazon.com for example, it is expected that the system will rank "Chanel No.7" perfume higher than "Wilton No.7 Flower Nail", as the former product is more popular than the latter, although they have the same syntactical similarity to the keyword.

Although the above issues have encountered in our experiments, the overall results ranked by our system are still optimised. According to the human participant studies, the system is able to effectively locate the best matched result, which is most important for the users. The rest of the order of the results produced by the system is reasonably compliant with human perception.

It should be noted that this evaluation is limited by the number of people who participate in the exercise, and the amount of time they were able to devote to each study. Although the human participants are carefully selected, there will unavoidably be some bias arising from the subjective view of the participants. In addition, going through six studies takes an average of over two hours to complete. It is unavoidable that participants will tend to focus less by the time they get to the last few studies. In ideal circumstances, the system should be put on line to enable public access to the system. The evaluation should then be conducted by statistical analysis of the time each search result (the link) is clicked. This will ensure that the system effectiveness is more accurately evaluated.

V. RELATED WORK

As presented in Section 1, xhRank is employed in our SW search engine, which searches SW resources. There are numerous well known SW search systems, such as Semplore [2], Falcons [3], Q2Semantic [4], SWSE [6], Swoogle [12], Watson[13], SemSearch [14], and Sindice [15]. The majority of these systems are currently the most widely used search systems for the SW, in particular the Open Linked Data [16]. Swoogle, Sindice, and Watson are mainly used as document-oriented SW search engines, whereas Falcons, Semplore, Q2Semantic, SemSearch and, SWSE specialist in entity-oriented SW searches, which are more related to our work.

In general, ranking schemes employed in existing SW search systems can be categorised into three types, based on importance, relevance, and query length respectively. Most of these ranking schemes cover one or two categories. Importance-based ranking can be further categorised into

Linkage-structure based (a variation of Google's PageRank) and popularity based approach. Swoogle uses linkage-based approach to rank the importance of SW document, but not SW resources. SWSE is SW resource-oriented. However, the linkage-based approach is based on incomplete graph structure and is executed at query time, which affects the query performance and accuracy. The popularity based approaches are used by Falcons and Semplore to rank SW resource and used by Q2Semantic to rank properties. Relevance-based rankings are used by many systems, such as Falcons, SWSE, Q2Semantic, Semplore, SemSearch, and Sindice to match keywords with SW documents or resources. These approaches are purely based on word occurrence, and do not taken into account word order and dispersion within the literals. Query-length-based approaches are used by Q2Semantic to match resource. However the ranking is based on clustered (incomplete) graphs.

Compared to the ranking mechanisms implemented in existing SW search systems, xhRank covers all these three categories of ranking types; its ranking algorithm is based on complete RDF graph structures; and it supports an alternative to the conventional word occurrence approach. Experiments we have conducted show that the ranking effectiveness is very good and the ranking results are compliant with human perceptions.

VI. CONCLUSION AND FUTURE WORK

In this paper, a ranking approach, namely xhRank, is proposed, which is tailored to the nature of the 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. The introduction of the importance and query length-based rankings to the graph-level (relevance-based) ranking further improves the ranking accuracy.

Our future research will begin with running our system against the Open Linked Data and Billion Triple Challenge [17], which contains the largest scale and very well interlinked SW datasets. Moreover, as explained in Section IV, an improved user interface will be developed for ordinary users to understand the query results in a more straightforward manner. Our system will be put on line to enable public access, and the evaluation will then be conducted by statistical analysis of the time each search result (the link) is clicked. These will ensure that the system effectiveness is more accurately evaluated.

ACKNOWLEDGMENT

This research is sponsored by the Research Endowment Trust Fund (RETF) and School of Systems Engineering of University of Reading.

REFERENCES

[1] A. Hogan, A. Harth, and S. Decker: ReConRank: A Scalable Ranking Method for Semantic Web Data with Context. In: Proc. 2nd SSWS, 2006.

[2] L. Zhang, Q. Liu, J. Zhang, H. Wang, Y. Pan, and Y. Yu: Semplore: An IR Approach to Scalable Hybrid Query of Semantic Web Data. In: Proc. 6th ISWC+ASWC, pp. 652-665. LNCS, vol. 4825, 2008.

[3] G. Cheng, W. Ge, and Y. Qu: Searching and Browsing Entities on the Semantic Web. In: Proc. 17th WWW, Poster Session, pp. 1101-1102, 2008.

[4] H Wang, K Zhang, Q Liu, T Tran, and Y Yu: Q2Semantic: A Lightweight Keyword Interface to Semantic Search. In: Proc. 5th ESWC. LNCS, vol. 5021, pp. 584-598, 2008.

[5] X. He and M. Baker: xhRank: Ranking Entities on the Semantic Web. In: 9th ISWC, Posters & Demo Sesstion, CEUR-WS, vol.658, pp. 41-44, 2010.

[6] Apache Lucene, URL: http://lucene.apache.org/, 20.09. 2011.

[7] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker: Searching and Browsing Linked Data with SWSE: the Semantic Web Search Engine. In :Journal of Web Semantics, 2011.

[8] myExperiment, http://www.myexperiment.org/, 20.09. 2011.

[9] Y. Guo, Z. Pan, and J. Heflin: LUBM: A Benchmark for OWL Knowledge Base Systems. In: J. of Web Semantics, vol. 3, no. 2-3, pp. 158-182, 2005.

[10] DBLP (RKB Explore), http://dblpp.rkbexplorer.com/, 20.09. 2011.

[11] A. Bandara: Semantic Description and Matching of Services for Pervasive Environments. PhD Thesis, University of Southampton, 2008.

[12] L. Ding, T. Finin, A. Joshi, Y. Peng, R. Cost, J. Sachs, R. Pan, P. Reddivari, and V. Doshi: Swoogle: A Search and Metadata Engine for the Semantic Web. Proc. 13th ACM Conf. on Information and Knowledge Management, pp. 652-659, 2004.

[13] M. d'Aquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, and E. Motta: WATSON: A Gateway for the Semantic Web. Proc. 4th ESWC, Poster Session, 2007.

[14] Y. Lei, V. Uren, and E. Motta, "SemSearch: A Search Engine for the Semantic Web", In: Proc. EKAW, pp. 238-245, 2006.

[15] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello: Sindice.com: A Document-Oriented Lookup Index for Open Linked Data. In: Journal of Metadata, Semantics and Ontologies, vol.3, no.1, pp. 37-52, 2008.

[16] Open Linked Data, URL: http://linkeddata.org/, 20.09. 2011.

[17] Billion Triple Challenge 2011 Dataset, http://challenge.semanticweb.org/, 20.09. 2011.