Ranking Documents Semantically Using Ontological Relationships

Boanerges Aleman-Meza[+], I. Budak Arpinar[*], Mustafa V. Nural[*] and Amit P. Sheth[δ]

[+] Rice University, Houston, TX, 77005, USA, ba8@rice.edu

[*] Computer Science, University of Georgia, Athens, GA 30602, {budak, nural}@cs.uga.edu

[δ] Kno.e.sis Center, Wright State University, Dayton, OH 45435, USA amit.sheth@wright.edu

***Abstract*—Although arguable success of today's keyword based search engines in certain information retrieval tasks, ranking search results in a meaningful way remains an open problem. In this work, the goal is to use of semantic relationships for ranking documents without relying on the existence of any specific structure in a document or links between documents. Instead, real-world entities are identified and the relevance of documents is determined using relationships that are known to exist between the entities in a populated ontology. We introduce a measure of relevance that is based on traversal and the semantics of relationships that link entities in an ontology. We expect that the semantic relationship-based ranking approach will be either an alternative or a complement to widely deployed document search for finding highly relevant documents that traditional syntactic and statistical techniques cannot find.**

I. Introduction

Research in search techniques was a critical component of the first generation of the Web, and has gone from academe to mainstream. A second generation "Semantic Web" is being built by adding semantic annotations that software can understand and from which humans can benefit. Discovering complex relationships on the Semantic Web and ranking search results based on these relationships will enable this vision and transform the hunt for documents into a more efficient analysis enabled by semantic technology. In today's Web search technologies, the link structure of the Web plays a critical role. In this work, our goal is to *use semantic relationships for ranking documents without relying on the existence of any specific structure in a document or links between documents.* In our work, real-world entities are identified and the relevance of documents is determined using relationships that are known to exist between the entities in a populated ontology, that is, by "connecting-the-dots." The implementation of the methods described here builds upon an existing architecture for processing unstructured information that solves some of the scalability aspects for text processing, indexing and basic keyword/entity document retrieval.

The contributions of this work are in demonstrating the role and benefits of using relationships for ranking documents when a user types a traditional keyword query. Our research contributions that make this possible are as follows:

- A flexible semantic discovery and ranking component takes user-defined criteria for identification of the most interesting semantic associations between entities in ontology.
- Semantic analytics techniques substantiate feasibility of the discovery of relevant associations between entities in an ontology of large scale such as that resulting from integrating a collaboration network with a social network (i.e., for a total of over 3 million entities). In particular, one technique is introduced to measure relevance of the nearest or neighboring entities to a particular entity from a populated ontology [2].
- The relevance of documents is determined based on the underlying concept of exploiting semantic relationships among entities in the context of a populated ontology.

Search of documents is an area that keeps on evolving. Document retrieval techniques are developed considering the possibilities offered by the nature of documents. For example, the techniques for retrieval of Web documents exploit the link structure among them [2]. Similarly, search techniques for Weblogs or blogs tend to make extensive use of the date/time of postings as criteria in the search techniques. This work proposes a method intended for ranking documents that do not have to contain links to other documents nor be constrained to any particular structure. While any well-formed ontology can be used, we expect the ontology to also contain a rich set of named entities and their relationships. Our architectural design allows using any well-formed existing ontology. However, we expect the following critical elements to be present in the populated ontology:

- The ontology must contain rich set of named entities.
- Semantic relationships between named entities should be available since they are the basis to the context of how one entity relates to others.
- The ontology used for retrieval and ranking of documents has to be related to the document collection of interest.

Note that the methods presented in this work exploit semantics of named entities and relationships whereas some other approaches exploit the semantics of nouns, verbs, etc. for incorporating semantics in search, for example, Cognition[1].

II. Related Work

The term *semantic search* is commonly used when semantics are used for improving search results. Existing semantic search approaches include entity-based search

[1] http://www.cognition.com/

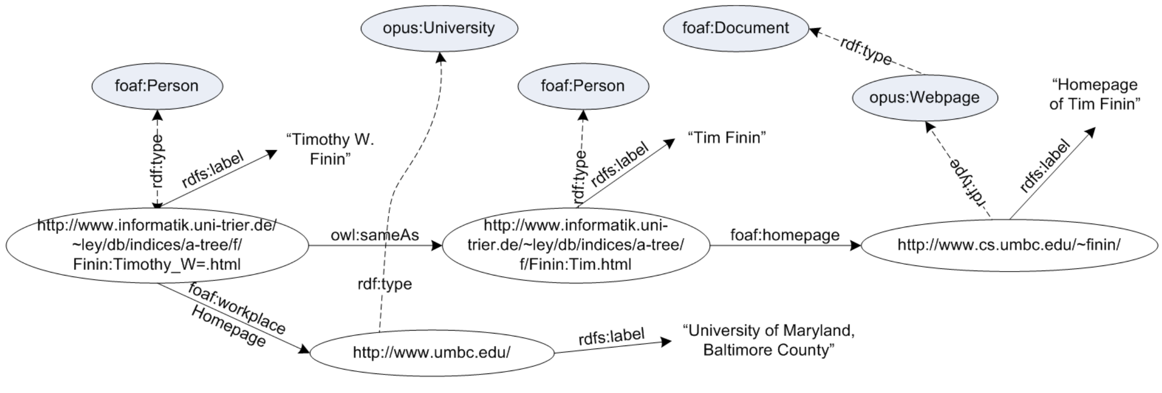


Figure 1: Example Relationships in SwetoDblp Ontology

[7][11]. The method described in this work also fits in the category of entity-based search.

A key difference with many link analysis algorithms is that our approach does not require that the documents be interlinked, as it is the case for Web documents. Methods such as PageRank [13] rely upon hyperlinks to assign a score on the basis the number references that a page receives, thus more popular pages have a higher rank.

Existing work that uses relationships for finding or ranking documents has yet to exploit the full potential of semantic relationships. For example, thread-activation techniques have been applied for searching related documents [6]. The main difference from our work is that their approach puts emphasis on literal values of entities as part of the search process. In our approach, only the 'name' of literals is used during the semantic annotation step (as well as synonyms). The main reason for which we do not use other literals of entities is that there might be a large variety of information in literals of entities that is not relevant for search purposes. For example, the text of an abstract of a publication is important metadata yet it might be more common to find the *title* of the publication than the *abstract* in other documents.

Techniques of discovery of semantic associations have been used for finding patents [14]. Their approach makes use of relationships to determine *important* entities. For example, a patent that has many *citation* relationships from other patents would be more important than a patent having many *inventor* relationships. Therefore, it is possible to determine importance of entities within the ontology. Their search approach can then retrieve patents based on keywords and show the important patents first. The disadvantage is that a patent by new inventors might not be in the top results even though the patent might be quite relevant to a query. This is because the aggregated effect of important entities makes it difficult for 'new' entities to gain high ranking.

Ontology concepts and relations have been used for finding research papers by incorporating link analysis techniques to determine popular entities within a populated ontology [16]. Their approach also uses relationships to determine important entities. For example, the authors of publications highly cited are more important than other authors. They show that the approach works correctly by comparing whether conference venues deemed important by the algorithm in fact are so.

III. RESEARCH BACKGROUND

A. *Large Populated Ontologies*

The development of Semantic Web applications typically involve processing of data represented using or supported by ontologies. A populated ontology is one that contains not only the schema or definition of the concepts and relationship names but also a large number of entities that constitute the instance population of the ontology.

In some domains, there are available ontologies that were built with significant human effort. However, it has been demonstrated that large ontologies can be built with tools for extraction and annotation of metadata. DBpedia demonstrates a large-scale automated ontology creation from wiki content [12]; see [1] for a survey of Web data extraction tools.

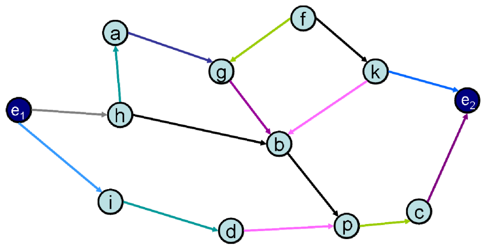
SwetoDblp is a large ontology that we created in the LSDIS Lab with a shallow schema yet a large number of real world instance data. It was built from an XML file from DBLP[2] whereby instead of a one-to-one mapping from XML to RDF, the creation of the ontology with emphasis on the addition of relationships and the semantics of URIs. Figure 1 shows a fragment from the SwetoDblp ontology. SwetoDblp is used as the underlying ontology for our experimental evaluation of the ranking scheme. SwetoDblp [3] is publicly available for download together with additional datasets that are used for its creation [4].

B. *Discovery, Analysis and Ranking of Relationships*

Relationships play an important role in the continuing evolution of the Web and it has been argued that people will use web search not only for documents, but also for information about semantic relationships. A key notion to process relationships between entities is the concept of *semantic associations*, which are the different sequences of relationships that interconnect two entities; semantic associations are based on intuitive notions such as connectivity and semantic similarity [9]. Each semantic association can be viewed as a simple path consisting of one or more relationships. Figure 2 illustrates a small graph of arbitrary entities and the results of a query for semantic associations

[2] http://dblp.uni-trier.de/

[3] http://knoesis.wright.edu/library/ontologies/swetodblp/



e1 → h → a → g ← f → k → e2
e1 → h → a → g ← f → k → b → p → c → e2
e1 → h → a → g ← b → p → c → e2
e1 → h → b ← g ← f → k → e2
e1 → h → b ← k → e2
e1 → h → b → p → c → e2
e1 → i → d → p → c → e2
e1 → i → d → p → b ← h → a → g ← f → k → e2
e1 → i → d → p → b ← g ← f → k → e2
e1 → i → d → p → b ← k → e2

Figure 2: Example Semantic Associations from a Small Graph

taking two of them as input (i.e., how are e_1 and e_2 are associated?).

Research in the area of ranking semantic relations includes [15][6][5], where the notion of "semantic ranking" is presented to rank query results returned within Semantic Web portals. The techniques reinterpret query results as "query knowledge-bases", whose similarity to the original knowledge-base provides the basis for ranking. The actual similarity between a query result and the original knowledge-base is derived from the number of similar super classes of the result and the original knowledge-base. In our approach, the relevancy of results usually depends on a context defined by users. Furthermore, the other ranking approaches are applied to Semantic Web query results and data (e.g., RDF triples) as opposed to Web documents in our approach.

C. *Semantic Annotation*

Semantic annotation is the process of identifying items of interest in unstructured text. In general, annotations that could be identified include words, nouns, named entities (e.g., person names, cities, and countries), dates, currency values, etc. We implemented a semantic annotation component that identifies named entities that exist in the ontology and keeps track of their position and offset in the text, their type (i.e., concept in an ontology), and their identifier (in this case the URI). Hence, the semantic annotation component takes as input a populated ontology, a list of concepts that is used to select the named-entities that are to be spotted in text, and a list of the names of attributes that are used as the 'name' of the entities to be spotted. In Semantic Web terminology, these are called literal properties; examples include rdfs:label and foaf:name (for their respective *rdfs* and *foaf* namespaces). The indexing of these semantically annotated documents produced by the semantic annotation process should also be addressed. In fact, the experiences developing such applications lead to investigate integrated architectures for processing unstructured data, as explained in the next section.

IV. RANKING DOCUMENTS USING RELATIONSHIPS

A. *Unstructured Information Management*

There are various architectures available for implementing new techniques in or related to search technology. In this work, we selected UIMA[4] (Unstructured Information Management Architecture) because it provides capabilities to build custom annotators, which can be used for indexing and retrieval based on whether the annotations appear in a document. UIMA provides a robust framework for text analysis tools, indexing and retrieval. It also provides an asynchronous scale-out framework for scalability.

B. *Overview*

Relevance of documents is based on the intuition of determining how the input query relates to the entities spotted in a document whereby such entities are connected in different ways in the ontology. That is, *a collection of documents can be viewed through the lenses of a large populated ontology containing named-entities*. The challenge is to incorporate human judgment into an algorithm to determine relevance of semantic relationships using ontology. The overall schematic includes a populated ontology, a collection of documents and semantic annotation thereof, indexing and retrieval, and ranking with respect to the user query. Collection of documents from various sources are annotated and indexed with semantic annotations together with the original documents using UIMA. The ontology (SwetoDblp in this case) is the source of domain knowledge involving entities and relationships. Relevance-based search engine incorporates the ontology, semantically annotated documents and the ranking parameters from the domain expert to generate ranking scores for the documents. The relevance measure makes use of subjective knowledge by a domain expert as described in the next section. One key element is that relationship sequences are assigned weights by referring to the schema of the ontology and this is done only once; regular users do not have to be concerned with this setup.

C. *Relevance Measure using Relationships*

In terms of entity-based search, the aim is to retrieve results that match the user input, which might directly specify the entity of interest. However, when hundreds or thousands of results are retrieved, ranking is necessary. *The relevance measure described here determines how relevant an entity is with respect to other entities that appear in the same document.* Let us refer to the entity that did match the user query as *match-entity*. The intuition behind determining relevance using relationships is that entities mentioned in a document are related directly or indirectly. The data contained in the ontology plays a key role because it contains relationships between entities. In our earlier work, we determined relevant documents with respect to a set of concepts [3]. The score of a document was the summation of the weights of paths from entities spotted in a document to the concepts. However, there is typically more than one path connecting two entities. In addition, there are connections between entities that do not necessarily imply relevance, regardless of their path length. It

[4] http://uima.apache.org

is then necessary to consider the type of each segment in a path connecting the match-entity to other entities in a document. In fact, the same two entities might lead to different relevance score because of the directionality of the path.

For example, consider the heated debate on link between autism and some vaccines [10]. The American Academy of Pediatrics and other major health organizations agree that there is probably no relationship between autism and vaccines. But at the same time many parents remain unconvinced. Suppose that a parent likes to make a more informed decision before her daughter is vaccinated with MMR (measles, mumps, and rubella) using the proposed method for ranking documents based on semantic relationships. Furthermore, assume that she is unaware of any side affects or medical debate about MMR vaccine-autism link. But, her intention is to get more information about this vaccine. So her search keywords involve *MMR vaccine*. The most important and relevant information for a mother is arguably risks and benefits associated with the vaccine. Suppose that three documents mention the *MMR vaccine.* However, first document also mentions *autism*, second document mentions *measles disease* whereas third document mentions *Merck*, which is the manufacturer of the vaccine. Since the input keywords from the user is *MMR vaccine*, then the entity *MMR vaccine* in the ontology would be the match-entity for the annotations at three documents. Suppose the ontology includes relationships *causes* connecting *vaccine* and *disorder*, *causes* connecting as *chemical substance* and *disorder*, *contains* connecting *vaccine* and *chemical substance*, *prevent* connecting *vaccine* and *disorder, and* finally *manufactures* connecting *manufacturer* and *vaccine* (Figure 3). Then, there are sequences of relationships in the ontology connecting *MMR vaccine* to *autism*, *MMR vaccine* to *measles,* and *MMR vaccine* to *Merck*. Three documents are related to the query, but arguably documents that mention *autism* or *measles* (or both) are more 'closely' related to the query because from the mother's perspective entities involved in *causes* and *prevents* relationships are more relevant (or important). It is easy for humans to assess such relationship, but a computer algorithm requires specific steps to assess the value added by each of the multiple relationships connecting entities (from match-entity to other entities in document).

There are various factors to consider in the relevance of relationships connecting two entities. It is possible to find the set of neighboring important entities of a match-entity. Then, the score of a document can be determined depending on how many of its annotations belong to such set.It is possible to analyze each relationship (i.e., edge) and expand it into a path of larger length according to the relevance of the path (or lack thereof). In the example of the match-entity *MMR vaccine*, it makes sense to consider the entity *autism* as 'important', which it is connected to *MMR vaccine* by a *causes* relationship (note that recent studies prove otherwise, yet again strength or correctness of this relationship is subject to ongoing debate; this simplified representation is for the sake of an intuitive example). On the other hand, if the entity *Merck* is the match-entity, then it might not make as much sense to consider each drug manufactured by *Merck* as important because there are too many. A domain expert needs to specify this type of "match-entity → relationship → entity" sequences. This might seem a daunting task at first but the schema part of the ontology is used to specify such sequences by referring to the classes of entities (i.e., concepts) instead of each entity at a time. In the previous example, sequences considered important would be "*Vaccine → causes → Disorder*" and "*Vaccine → prevents → Disorder.*"

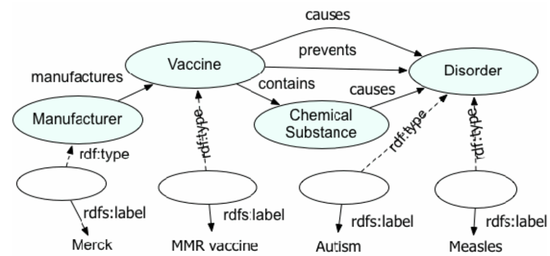


Figure 3: Semantic Relations

The previous examples illustrated paths of length one. However, paths of longer length might also reveal important information on how MMR *vaccine* and *autism* are connected: "*Vaccine → contains →* Chemical Substance → causes → *Disorder.*"

An additional factor in the sequences that determine important entities is that the degree of such importance can vary. In our initial experiments, we used values between zero and one yet a simpler approach is to use three levels: *low*, *medium*, and *high*. For example, the sequence "*Vaccine → causes → Disorder*" could be given a high-importance where as the sequence "*Manufacturer → manufactures → Vaccine*" could be given a low-importance. The relevance measure takes as input the match-entity, the other entities with respect to which the relevance is determined, and a list of sequences with their corresponding importance levels. The relevance measure then proceeds as follows:

i. Initialize total score to zero
ii. Each sequence is considered independently, for which:
   a. Each possible undirected path starting from the match-entity is evaluated with respect to sequence to determine a set of neighboring entities that are important with respect to the match-entity.
   b. The resulting set, possibly empty, of the neighboring entities, is added to either of these sets: *lowSet*, *mediumSet* and *highSet*.
iii. Take each entity in the "other entities set"
   a. If it is in *lowSet*, then add the corresponding low-score to the total score
   b. If it is in *mediumSet*, then add the corresponding medium-score to the total score
   c. If it *is in highSet*, then add the corresponding high-score to the total score

Finally, the total score contains the relevance of the match-entity with respect to other entities based on whether and to which degree they are related to the match-entity. A domain expert assigns the "low/medium/high" scores, as mentioned earlier. In our experience, these facilitate the scoring of a document whereby even small differences in scores has an impact on the ranked results.

D. *Ranking of Documents Using Relevance Measure*

The retrieval and ranking process is as follows. The input from user consists of one or more query terms, as mentioned earlier. For an input query from user, two queries are created and then resolved by UIMA (through its indexing mechanism). The first query retrieves documents that match the user query as part of an existing annotation (i.e., an annotation-query). The second query retrieves documents that match the user input as a traditional keyword-based search. These keyword results include a score that is computed by UIMA. We include keyword matches (with their default score) in the results presented to user yet our ranking method does not re-rank these results. In fact, the documents that match both a keyword-query and an annotated query are removed from the keyword-matches to avoid showing duplicate results to the user. The intention is to have a "fall-back" mechanism into keyword-search when the user input does not match any of the existing annotations.

The core of our ranking method takes place when the entity-matches from an annotation-query are re-ranked. The model to compute the score of a document requires information from three pieces. The first is the entity from the ontology that did match the annotation query. For example, the entity *IBM Corporation* is the match for an input query *IBM* that matched an annotation in a document. Synonyms included in the ontology are used by the annotation step automatically. Second, annotations of other entities spotted in the document are used to compute the relevance of the document. Third, the ontology information is used as well. Hence, the score of a document d is a function of the entity e that does match the user input, the set A of other annotations in the document, and the ontology O, namely, *score* $d = r(e, A, O)$. Thus, the score of a document is different if the input query does match a different annotation in the document, or if the ontology undergoes modifications. If the ontology is modified to have more (or fewer) named entities, then the set A might be different and affect the score of a document. If the ontology is modified to have more (or fewer) connections among its entities, then the relevance measure might produce a different score for a document. It is reasonable to assume that the ontology is not going to change frequently, at least not on per-query basis. Hence, the set A containing other annotations in the document will not change either. Then, the only other variable in computing the score of a document is that of the entity whose annotation in the document did match the user input. In the simplest case, only one entity from the ontology is a match. The score of the document is then determined directly by the relevance measure. In this case, two groups of results would be shown to the user. One with the resulting documents ranked according to the relevance measure. The other with the keyword results for the query, if any.

E. *Remarks on Usage of Ontology*

Other methods have used the ontology itself to assign different importance values to entities in the ontology [14][16]. We explored this possibility yet it is possible that newer elements in the ontology could not be assigned a satisfying importance value unless they are referenced more frequently in the ontology, that is, by means of other entities linking to them. In contemporary Web search techniques, it might be beneficial that methods provide the most popular entity. However, we believe that in other document collections it is more important to find the relevant documents, which might not be linked from other documents sufficiently to be retrieved top in the list of ordering of results from link-analysis methods.

V. EXPERIMENTAL EVALUATION

In the experimentation, we used the SwetoDblp Ontology [4], which is based from data from the DBLP bibliography as mentioned earlier. The document collection used in the evaluations was chosen directly from the metadata in DBLP publications that links to the electronic edition (i.e. ee links) of the publications.

In the evaluation setup, we randomly chose family name of authors and then queried the system with the family name as input keyword. The search-results are organized according to each entity-name match. Hence, we verified whether the documents found for each named entity do match the known documents (through the ee link). We crawled documents that are linked from DBLP and performed semantic annotation with the ontology. The known links from publications of authors is then used to verify whether the results of a query do match with retrieved documents. The Figure 5 illustrates the measure of precision for the top 5, 10, 15 and 20 results for over 150 random queries. The average value for precision in the top five and top 10 results was 77% and for the top 15 results it was 73%. In Figure 4 it can be seen that a large majority of the results were near or above the 80% line. Next, we evaluated how recall compares with precision when the top 10 results are considered. The Figure 5 is a scattered-plot illustrating this where the queries are the same as those in previous figure. Precision vs. recall illustrates that a good number of the results are at or over the 80% precision yet for a small number of results both precision and recall are rather low. After inspecting manually the queries that lead to such low values we found that few of them were family-names that are common given-names such as Philip, Anthony, and Christian.

An important aspect of this study is whether or not the use of relationships brings benefits for finding relevant documents. Nevertheless, high values of precision counts as an evidence of bringing benefits.

VI. CONCLUSIONS AND FUTURE WORK

Just as the link structure of the Web is a critical component in today's Web search technologies, complex relationships will be an important component in emerging Web search technologies. This paper addresses the problem of how to exploit semantic relationships of named-entities to improve relevance in search and ranking of documents. The use of relationships to rank documents is promising. This can prove advantageous in search scenarios where it cannot be expected that the documents be interlinked. Moreover, there is potential benefit of combining this method with those based on link analysis. We also found that the scoring method is robust for the cases when there are multiple entity-matches for a query.

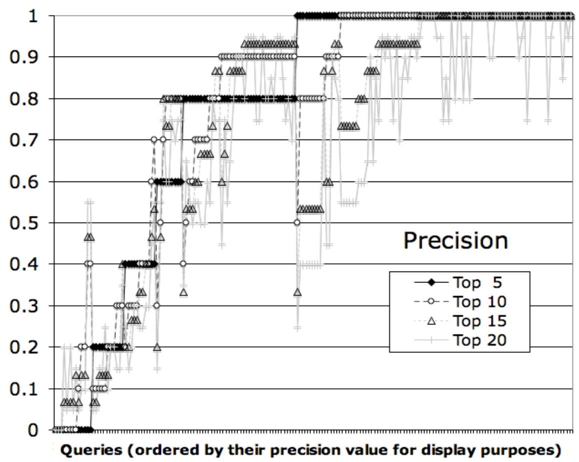


Figure 4: Precision for top 5, 10, 15, and 20 results

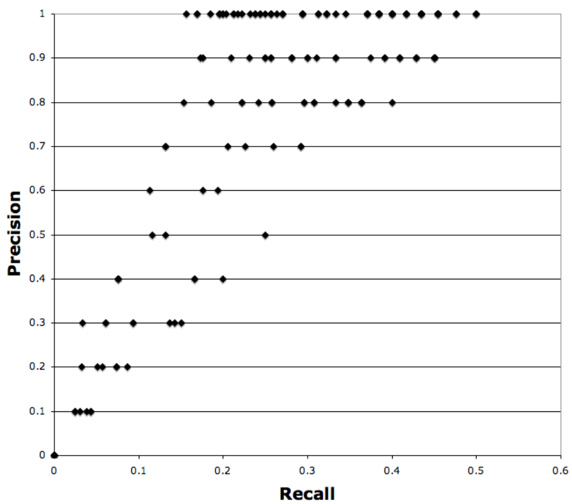


Figure 5: Precision vs. Recall for top 10 results

There are a few weaknesses on the applicability of the methods proposed in this paper. An ontology that is far from complete in its domain (i.e. low-quality) could negatively affect semantic annotation and retrieval steps. It is also important to note that the dependence on a semantic annotation process could limit the applicability of this method to documents containing *unnamed* entities. For example, entities of type *event* rarely are given a name (exceptions include the "9/11" events). Their applicability though, could be significant, for example, in search of events in *news*. In addition, a domain expert manually assigns the importance levels of relationships in the domain ontology. Although end-users are unaware of this process, a better approach may require automating this process, perhaps by capturing user-interests and feeding them into the ranking engine to produce customized rankings for a particular user.

Of particular interest are comparisons of how the presented research fits and/or complements with techniques based on link analysis. We anticipate three cases. In the first, documents are simply contained in text-corpora without any links between them. The second case is that of documents in a corporate intranet where although the documents contain links between them, it might not be sufficient for achieving the full value of link analysis methods. The third case involves documents at large on the Web. It could be possible that a link-analysis method retrieves documents based on user input and the top documents are later processed by our techniques.

VII. ACKNOWLEDGEMENTS

This research was supported by NSF grant No. IIS-0325464 awarded to the University of Georgia titled "Semdis: Discovering Complex Relationships in the Semantic Web".

REFERENCES

[1] A. H. F. Laender, B. A. Ribeiro-Neto, A. S. da Silva, and J. S. Teixeira. A Brief Survey of Web Data Extraction Tools. SIGMOD Record, 31(2):84-93, 2002.

[2] B. Aleman-Meza, Ranking Documents based on Relevance of Semantic Relationships. PhD Dissertation, University of Georgia, 2007.

[3] B. Aleman-Meza, A. P. Sheth, P. Burns, D. Paliniswami, M. Eavenson, and I. B. Arpinar. Semantic Analytics in Intelligence: Applying Semantic Association Discovery to Determine Relevance of Heterogeneous Documents. In Advanced Topics in Database Research, Volume 5, (Keng Siau, Ed.), Idea Group Publishing, pages 401-419, 2006.

[4] B. Aleman-Meza, F. Hakimpour, I. B. Arpinar, and A. P. Sheth. SwetoDblp Ontology of Computer Science Publications. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 2007.

[5] C. Hurtado, A. Poulovassilis, and P. Wood. Ranking Approximate Answers to Semantic Web Queries, In Proc. 6th European Semantic Web Conference, Springer, 2009.

[6] C. Rocha, D. Schwabe, and M. P. Aragao. A Hybrid Approach for Searching in the Semantic Web. In Proc. 13th International World Wide Web Conference, New York, NY, pages 374-383, 2004.

[7] H. Chen, K. J. Lynch, K. Basu, and T. D. Ng. Generating, Integrating and Activating Thesauri for Concept-Based Document Retrieval. IEEE Intelligent Systems, 8(2):25-35, 1993.

[8] J. Hassell, B. Aleman-Meza, and I. B. Arpinar. Ontology-Driven Automatic Entity Disambiguation in Unstructured Text. In Proc. 5th International Semantic Web Conference, Athens, Georgia, pages 44-57, 2006.

[9] K. Anyanwu, A. Maduku, and A. P. Sheth, "Semrank: ranking complexrelationship search results on the semantic web," in In 14th International World Wide Web Conference. ACM Press, 2005, pp. 117–127.

[10] M. F. Downs. Autism-Vaccine Link: Evidence Doesn't Dispel Doubts. WebMD, 2008.

[11] R. V. Guha, R. McCool, and R. Fikes. Contexts for the Semantic Web. In Proc. 3rd International Semantic Web Conference, Hiroshima, Japan, pages 32-46, 2004.

[12] S. Auer, and J. Lehmann. What have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content. In Proc. 4th European Semantic Web Conference, Innsbruck, Austria, pages 503-517, 2007.

[13] S. Brin, and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Proc. 7th International World Wide Web Conference, 1998.

[14] S. Mukherjea, and B. Bamba. BioPatentMiner: An Information Retrieval System for BioMedical Patents. In Proc. 30th International Conference on Very Large Data Bases, Toronto, Canada, 2004.

[15] T. Franz, A. Schultz, S. Sizov, and S. Staab. TripleRank: Ranking Semantic Web Data By Tensor Decomposition, In Proc. 8th International Semantic Web Conference (ISWC2009), 2009.

[16] Z. Nie, Y. Zhang, J. R. Wen, and W. Y. Ma. Object-level Ranking: Bringing Order to Web Objects. In Proc. 14th International World Wide Web Conference, Chiba, Japan, pages 567-574, 2005.