RANKING DOCUMENTS BASED ON RELEVANCE OF SEMANTIC RELATIONSHIPS

by

BOANERGES ALEMAN MEZA

(Under the Direction of Ismailcem Budak Arpinar)

ABSTRACT

In today's web search technologies, the link structure of the web plays a critical role. In this work, the 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. Instead, named/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." We introduce a measure of relevance that is based on traversal and the semantics of relationships that link entities in an ontology. 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 thesis are in demonstrating the role and benefits of using relationships for ranking documents when a user types a traditional keyword query. The research components that make this possible are as follows. First, a flexible semantic discovery and ranking component takes user-defined criteria for identification of the most interesting semantic associations between entities in an ontology. Second, 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. Last, the relevance of documents is determined based on the underlying concept of exploiting semantic relationships among entities in the context of a populated ontology. Our research involves new capabilities in combining the relevance measure techniques along with using or adapting earlier capabilities of semantic metadata extraction, semantic annotation, practical domain-specific ontology creation, fast main-memory query processing of semantic associations, and document-indexing capabilities that include keyword and annotation-based document retrieval. 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.

INDEX WORDS: Semantic Analytics, Knowledge Discovery, Semantic Associations, Semantic Web, Ontology, Ranking, Web Search, Annotation, Relevance Measures, Social Networks, OWL, RDF

RANKING DOCUMENTS BASED ON RELEVANCE OF SEMANTIC RELATIONSHIPS

by

BOANERGES ALEMAN MEZA

Master of Applied Mathematical Sciences, University of Georgia, 2001

B.E. in Computer Engineering, Instituto Tecnologico de Chihuahua II, Mexico, 1998

A Dissertation Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2007

© 2007

Boanerges Aleman Meza

All Rights Reserved

RANKING DOCUMENTS BASED ON RELEVANCE OF SEMANTIC RELATIONSHIPS

by

BOANERGES ALEMAN MEZA

Major Professor:	Ismailcem Budak Arpinar
Committee:	Amit P. Sheth Charles B. Cross John A. Miller

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2007

DEDICATION

I would like to dedicate this thesis to my parents, Concepcion Meza Montes and Boanerges Aleman Barrera, without whose support this work would not have been possible.

ACKNOWLEDGEMENTS

Firstly, I would like to thank all parties who sponsored me for this degree. Thanks to CONACYT ("Consejo Nacional de Ciencia y Tecnología") - National Council for Science and Technology of Mexico, and the Graduate School at the University of Georgia for the Dissertation Completion Assistantship.

I would like to thank my advisor I. Budak Arpinar for his advice, guidance and support. In particular, I thank him for giving me the opportunity for mentoring several students pursuing their Master's degree. I would also like to thank members of my doctoral committee, Professors Krys Kochut, Charles Cross, John A. Miller, and Amit P. Sheth. I would like to thank Professor Sheth for his continued support, guidance and invaluable insights that led me to think of him as unofficial co-advisor.

I have been lucky to be part of a group consisting of many active students. The opportunity to work with a variety of enthusiastic young researchers has been invaluable. I appreciate the opportunities for research/collaboration as well as discussions/interactions with a variety of people, including Cartic Ramakrishnan, Chris Halaschek, Christopher Thomas, Delroy Cameron, Farshad Hakimpour, Kunal Verma, Matthew Eavenson, Meena Ngarajan, Sheron Decker, and other members/alumni of the LSDIS Lab. In addition, I would like to thank Hector de los Santos Posadas, Teresa and Mike Shirley, and Weiwei Zhong for their friendship and support. Finally, thanks to God.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Existing Web search technologies have attained success by using link analysis techniques to determine popular (and therefore arguably important) Web documents. Other techniques make use of other information to determine relevant documents such as click-through data and explicit feedback. It has been noted that enterprise corpora lack the highly hyperlinked structure of documents that is required by link analysis techniques [24]. The method proposed in this thesis does not make use of or depend on the existence of hyperlinks or any specific structure within documents. The approach taken uses the semantics of relationships between named-entities for ranking documents. Hence, a populated ontology containing named-entities is utilized. Available datasets of this type are increasingly available online. For example, the National Library of Medicine's MeSH (Medical Subject Heading) vocabulary is used for annotation of scientific literature. Efforts in industry [88] as well as those by scientific communities (e.g., Open Biological Ontologies, http://obo.sourceforge.net, which lists well over fifty ontologies) have demonstrated capabilities for building large populated ontologies. Additionally, metadata extraction and annotation in web pages has been addressed earlier and proven scalable [31][32][44]. We expect two critical elements to be present in populated ontologies. First, the ontology must contain named entities. Some populated ontologies have few or no named entities. For example, many events on the terrorism domain are not given a name and they are referred to using general terms such as car bombing (together with a date). Named entities are needed for entity spotting in documents. Second, the ontology needs to have a good number of relationships interconnecting its instance population. Semantic relationships (also known as typed or named relationships) are the basis to the context of how one entity relates to others. For example, a list of cities and countries has much more value when there are relationships connecting each city to the country where it is located. The ontology used for retrieval and ranking of documents has

to be related to the document collection of interest. In some cases this might be a limitation due to the lack of an ontology, but the number of ontologies available is increasing as mentioned before (see also [34]).

A number of techniques that rely upon ontologies for better search and/or ranking require the user to model/formulate complex queries involving concepts and relations. We base our approach on the premise that users will not be asked to formulate complicated queries. A query is entered in exactly the same way as in existing search engines, but it will match in different ways according to the spotted named entities in documents. For example, the keyword *georgia* is a match for both *University of Georgia* and *Georgia Institute of Technology*. The use of named-entities allows us to present results to user depending on the named-entities that match the query. The intention is to provide the user with access to search results that are grouped by entity-match. If a query does match one or more entities, then the results are presented grouped by each entity match, whereas, the results that only match keywords (but not known entities) are simply listed as keyword matches. The results for an entity match are ranked independently of the results of others. In the previous example of input keyword *georgia*, there would be two ranked (large) lists of documents for named entities *University of Georgia* and *Georgia Institute of Technology*. This type of search is typically referred to as entity-based search.

Nevertheless, entity-based search methods require a variety of capabilities such as spotting of named-entities, index, and retrieval. Additionally, scalability is always an important requirement for this type of applications. Hence, it is convenient to build upon existing architectures designed for processing unstructured information. We chose to use UIMA (Unstructured Information Management Architecture) [37] because it provides a robust framework for text analysis tools, indexing and retrieval. For the purpose of validating our approach, we implemented a UIMA Analysis Engine that processes a document to detect named-entities from a populated ontology. The output consists of an annotated document. The document collection processing capabilities of UIMA take the document collection to create indexes of keywords and of the spotted named entities in the documents.

Semantic annotation and indexing take place as pre-processing steps. For querying, UIMA includes capability of retrieval of documents that match either a keyword query in the traditional way, or a keyword as part of a particular annotation. It is at this point where our method takes the results from UIMA and computes a score for the documents based on the input keyword(s). The novelty of our approach is in using semantic relationships between entities to determine relevance of documents. In particular, the relevance measure first takes the annotations within a document that match the keyword input from user. In the example introduced earlier, it would find that the annotation for entity *University of Georgia* does match the input *georgia* from user.

Figure 1: Documents Containing Named Entities and their Relationships

Second, the ontology is queried to determine the relevant entities to the entity that did match the annotation. Figure 1 illustrates that documents are not just linked with implicit or unnamed links but such links are made within a context and the documents contain named entities that have relationships to other entities if an ontology properly captures such information. The keyword input from user is interpreted with respect to the ontology that captures the domain of interest.

The documents are expected to be in or related to such domain so that named entities match correctly to the annotations in documents. It is at this point where the semantics of relationships play a significant role. For example, a 'university' has a stronger relationship to the city and state where it is located than to a neighboring state. The determination of the relevance effect of relationships from the ontology takes place only once and it is specific to the ontology being used. A domain expert performs this task, but the manual effort is not significant, because it involves referring to concepts and relationships of the ontology schema, which tends to be relatively small in ontologies that have large number of instance data [86]. The relevance measure then considers how the other entities in a document relate to the entity that did match the annotation. Other aspects for determining relevance score of a document include resolving ambiguities for cases when more than one entity from the ontology have same name. In fact, no entity disambiguation takes place in the preprocessing for semantic annotation of documents. Nevertheless, the intuition behind our previous work on entity disambiguation [49] is quite similar but applied differently in this work. In summary, the relevance score of a document exploits both the information that the ontology provides with respect to relationships among entities and the spotted entities in documents.

1.1 Contributions

The contributions of this thesis demonstrate the benefits of using relationships for ranking documents where the input is a keyword query. The necessary components to make this possible include new techniques as well as use and adaptation of earlier techniques for analysis of the semantics of relationships, their discovery, and semantic annotations. The contributions of this thesis are as follows.

(1) A flexible semantic discovery and ranking approach that takes user-defined criteria for identification of the most interesting semantic associations between entities in an ontology.

(2) Semantic analytics techniques that substantiate feasibility of the discovery and analysis 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.

(3) An ontological approach for determining the relevance of documents based on the underlying concept of exploiting semantic relationships among entities [8]. This is achieved by combining the relevance measure techniques mentioned before with existing capabilities of semantic metadata extraction, semantic annotation, practical domain-specific ontology creation, fast main-memory query processing of semantic associations, and document-indexing capabilities that include keyword and annotation-based document retrieval. There are two key elements of this contribution. First, a novel method determines relevance of entities using semantic relationships exploiting metadata from a populated ontology. Second, an implementation that uses a large, real-world ontology to demonstrate effective use of semantic relationships for ranking documents. We describe how we implemented a complete search system and present evaluations using measures of precision and recall over a document collection that contains names of people and affiliations in the domain of Computer Science Research.

1.2 Context and Scope

The Information Retrieval research community has addressed the problem of finding relevant-documents but there are additional challenges and possibilities when Semantic Web techniques are considered [87]. 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. Similarly, search techniques for Weblogs or *blogs* tend to make extensive use of the date/time of postings as criteria in the search techniques. The methods proposed in this thesis are intended for ranking documents that do not have to contain links to other documents nor be constrained to any specific structure. In addition, the methods will perform better when *named* entities are mentioned in the documents, whereby such named-entities exist in the ontology being used by the system. The architecture is designed to be able to use arbitrary ontologies yet these should be populated ontologies. That is, the ontology should contain a large number of named entities interlinked to other entities because the method relies on relationships between entities to determine relevance. Some methods rely on pre-processing that assigns a rank to each document. Our method retrieves documents relevant to a query and then ranks them. Other approaches

exploit the semantics of nouns, verbs, etc. for incorporating semantics in search, for example, by using WordNet [70]. The methods presented in this thesis exploit semantics of named entities instead.

The challenges in research dealing with ranking documents include traditional components in information retrieval systems. Due to the large number of documents on the Web, it is necessary to process many documents, which need to be processed and indexed. Other components include fast retrieval of the documents relevant to a query and their ranking. In the work presented in this thesis, it is also necessary to perform a process of semantic annotation for spotting appearances of named-entities from the ontology in the document collection. The capabilities for indexing and retrieval of documents containing such *annotations* bring additional complexity. The type of challenges involved in techniques that process large ontologies include processing of data that is organized in a graph form as opposed to traditional database tables. The techniques presented in this thesis make extensive use of graph traversal to determine how entities in an ontology are connected. This is often needed to determine relevant entities according to the paths connecting them. The challenge involved is that ontologies containing over a million entities are no longer the exception [91]. Lastly, other challenges exist in evaluation of the approach. It is typically difficult to devise methods to evaluate many queries in an automated manner. This is due to the difficulty of knowing in advance which documents are relevant to a query. In fact, this is a more challenging problem when the search method can differentiate between results that match different named-entities for the same input from user. It would be necessary to know in advance the subset of documents that are relevant for each different named-entity matching the query. In summary, the challenges involved are in terms of traditional document retrieval as well as processing of ontology data and its usage for annotation, indexing and retrieval of documents, measuring relevance among entities in the ontology, and measure relevance using entities and their relationships for ranking of documents.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter first describes necessary components that are not the main contributions of the thesis yet are important components of the proposed method for relationship-based ranking of documents. These components are a populated ontology, semantic annotation of document collection to identify the named entities from the ontology, indexing and retrieval based on keyword input from user. Second, related previous work is described.

2.1 Semantic Web

The Semantic Web [20] is a vision that describes a possible form that the Web will take as it evolves. Such vision relies upon added semantics to content that in the first version of the Web was intended solely for human consumption. This can be viewed from the perspective that a human could easily interpret a variety of web pages and glean understanding thereof. Computers, on the other hand, can only achieve limited understanding unless more explicit data is available. It is expected that the mechanisms to describe data in Semantic Web terms will facilitate applications to exploit data in more ways and lead to automation of tasks. The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.

One of the basic means to explicitly state or add meaning to data is the Resource Description Framework, which provides a framework to capture the meaning of an entity (or resource) by specifying how it relates to other entities (or classes of resources). Thus, this is a step beyond metadata, in particular, semantic metadata, which can be described as content enriched with semantic annotations using classes and relationships from an ontology [89]. Semantic technologies are gaining wider use in Web applications [92][62][71].

2.2 Large Populated Ontologies

The development of Semantic Web applications typically involves processing of data represented using or supported by ontologies. An ontology is a specification of a conceptualization [39] yet the value of ontologies is in the agreement they are intended to provide (for humans, and/or machines). In the Semantic Web, an ontology can be viewed as a vocabulary used to describe a world model. A populated ontology is one that contains not only the schema or definition of the classes/concepts and relationship names but also a large number of entities that constitute the instance population of the ontology. That is, not just the schema of the ontology is of particular interest, but also the population (instances, assertions or description base) of the ontology. A highly populated ontology (ontology with instances or assertions) is critical for assessing effectiveness, and scalability of core semantic techniques such as semantic disambiguation, reasoning, and discovery techniques. Ontology population has been identified as a key enabler of practical semantic applications in industry; for example, Semagix reports that its typical commercially developed ontologies have over one million objects [91]. Another important factor related to the population of the ontology is that it should be possible to capture instances that are highly connected (i.e., the knowledge base should be deep with many explicit relationships among the instances). This will allow for a more detailed analysis of current and future semantic tools and applications, especially those that exploit the way in which instances are related.

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 [47][48][95][103][105]; see [59] for a survey of Web data extraction tools. Industry efforts have demonstrated capabilities for building large populated ontologies [88], which are sometimes called shallow ontologies. *Shallow* ontologies contain large amounts of data and the concepts and relations are unlikely to change, whereas *deep* ontologies contain smaller (or not any) amounts of data but the actual concepts and relations require extensive efforts on their building and maintenance [86].

An ontology intended for search of documents calls for focusing on a specific domain where populated ontologies are available or can be easily built. Ontologies used in our approach need to contain

named-entities that relate to other entities in the ontology (i.e., resource-to-resource triples). The named-entities from the ontology are expected to appear in the document collection. This can be a limitation in certain domains for which ontologies are yet to be created. However, techniques and developments continue for metadata extraction of semantics. For example, a recent work opens possibilities of ontology creation from wiki content [18]. In domains such as life sciences (http://obo.sourceforge.net) and health-care many comprehensive, open, and large ontologies have been developed. For example, UniProt (http://www.pir.uniprot.org) and Glyco/Propreo [83] are ontologies with well over one million entities (see also http:///bioontology.org/). In domains such as financial services/regulatory compliance [94] and intelligence/defense, a number of non-public ontologies have been developed. Other large ontologies such as TAP [41] and Lehigh Benchmark (http://swat.cse.lehigh.edu/projects/lubm/) have also proven useful for developments and evaluations in Semantic Web research. Lehigh Benchmark is a suitable dataset for performance evaluation but it is a synthetic dataset.

SWETO Ontology. We now review our earlier work for building a test-bed ontology, called SWETO (Semantic Web Technology Evaluation Ontology) [4]. SWETO has demonstrated that large populated ontologies can be built from data extracted from a variety of Web sources. We have found that the richness and diversity of relationships within an ontology is a crucial aspect. SWETO captures real world knowledge with over 40 classes populated with a growing set of relevant facts, currently at about one million instances. The schema was created in a bottom-up fashion where the data sources dictate the classes and relationships. The ontology was created using Semagix Freedom, a commercial product which evolved from the LSDIS lab's past research in semantic interoperability and the SCORE technology [88]. The Freedom toolkit allows for the creation of an ontology, in which a user can define classes and the relationships that it is involved in using a graphical environment.

We selected as data sources highly reliable Web sites that provide instances in a semi-structured format, unstructured data with structures easy to parse (e.g., html pages with tables), or dynamic sites with database back-ends. In addition, the Freedom toolkit has useful capabilities for focused crawling by exploiting the structure of Web pages and directories. We carefully considered the types and quantity of

relationships available in each data source by preferring those sources in which instances were interconnected. We considered sources whose instances would have rich metadata. For example, for a 'Person' instance, the data source also provides attributes such as gender, address, place of birth, etc. Last, public and open sources were preferred, such as government Web sites, academic sources, etc. because of our desire to make SWETO openly available. Figure 2 illustrates the fact that a variety of heterogeneous data sources are extracted for the instance population of the ontology.

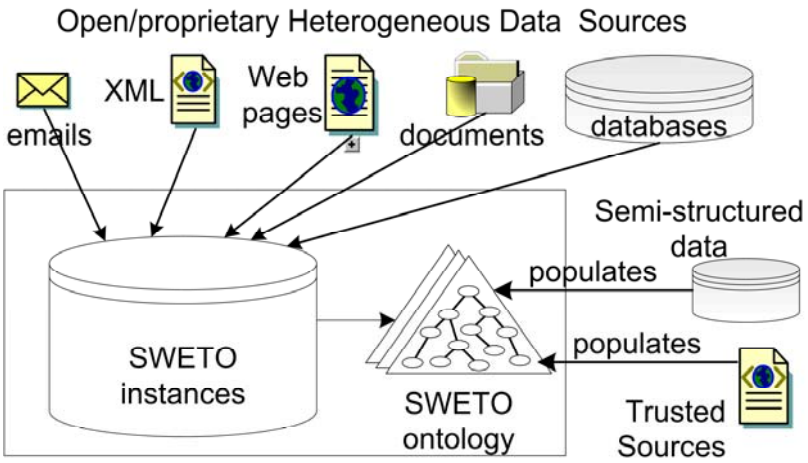

Figure 2: Overview of Data Sources for Instance Population of SWETO Ontology

All knowledge (or facts that populate the ontology) was extracted using Semagix Freedom. Essentially, extractors were created within the Freedom environment, in which regular expressions are written to extract text from standard html, semi-structured (XML), and database-driven Web pages. As the Web pages are 'scraped' and analyzed (e.g., for name spotting) by the Freedom extractors, the extracted instances are stored in the appropriate classes in the ontology. Additionally, provenance information, including source, time and date of extraction, etc., is maintained for all extracted data. We later utilize Freedom's API for exporting both the ontology and its instances into one of the semantic web representation languages (e.g., RDF). For keeping the knowledge base up to date, the extractors can be scheduled to rerun at user specified time and date intervals. Automatic data extraction and insertion into a knowledge base also raise issues related to the highly researched area of entity disambiguation [55][68][80][82]. In SWETO, we focused greatly on this aspect of ontology population.

Using Freedom, instances can be disambiguated using syntactic matches and similarities (aliases), customizable ranking rules, and relationship similarities among instances. Freedom is thus able to automatically disambiguate instances as they are extracted [44] but if it detects ambiguity among new instances and those within the knowledge base, yet it is unable to disambiguate them within a preset degree of certainty, the instances are flagged for manual disambiguation.

The instance population of SWETO includes over 800,000 instances and over 1,500,000 explicit relationships among them. SWETO has been a frequently used dataset in research involving populated ontologies [3][4][5][17][33][53][78][97][101].

SwetoDblp Ontology of Computer Science Publications. SwetoDblp [10] builds upon our previous experience on creating and using SWETO. It integrates additional relationships and entities from other data sources. It is a large populated ontology with a shallow schema yet a large number of real world instance data. It was built from an XML file from DBLP (http://dblp.uni-trier.de/) whereby instead of a one-to-one mapping from XML to RDF, the creation of the ontology emphasizes the addition of relationships and the value of URIs. The hierarchical structure of XML documents implies relationships from parent to children elements. However, such relationships depend upon human interpretation. The creation of SwetoDblp is done through a SAX-parsing process that performs various domain-specific transformations on a large XML document to produce RDF. The schema-vocabulary part of the ontology is a subset of an ontology used by the back-end system of the LSDIS Lab's publications library. This schema adopts major concepts and relationships from other vocabularies and extends them where needed. In addition, we used standard practices to indicate equivalence of classes and relations to six other vocabularies such as the AKTors publication ontology [85] (using *equivalentClass* and *equivalentProperty* of the OWL vocabulary).

We followed specific guidelines to provide the general framework under which various domain specific mappings were implemented for the creation of SwetoDblp. First, in the original XML document, the names of persons appear as plain literal values such as <author>Li Ding</author> but each of these is represented as an RDF resource in SwetoDblp having its own URI. Our goal was to create URIs so that

they can be reused by other datasets based on the assumption that the URI of choice will likely be the URL pointing to the author's DBLP entry on the Web. However, other methods to create URIs do allow for content-negotiation depending on whether a request on the Web indicates that a Webpage is needed or that XML/RDF content is needed. The form in which URIs are set in DBPedia is one example of such content-negotiation [18]. Second, we made an effort to reuse existing semantic web vocabularies whenever possible. For example, if the homepage of an author is available in the original XML document, then such relationship is kept in the resulting RDF by using *foaf:homepage* (of the FOAF vocabulary). In addition, the 'homepage' is represented as an RDF resource (with the URL as its URI); this domain-specific mapping automatically assigns a label to the homepage resource with the prefix "Homepage of ." In very few cases, the data from DBLP indicates that a person can be referred to by more than one name. Examples include "Tim Finin" and "Timothy W. Finin." In SwetoDblp, such names are explicitly represented with a *owl:sameAs* relationship (which is the only relationship from the OWL vocabulary that is used in SwetoDblp instance data). Lastly, few other data sources used in the creation of SwetoDblp. Two of them are Universities and Organizations datasets that are used to determine and then explicitly add an affiliation relationship to a person either from the homepage of the person, or from 'note' elements appearing in the DBLP XML document. Similarly, a dataset about Publishers is used to create a relationship from literal values such as <publisher>McGraw-Hill</publisher> to an RDF publisher entity with an URI that points to the actual website of the publishing company. The Publishers dataset was created manually with the most commonly appearing names of publishers in the original XML document from DBLP, but more publisher entities were added to cover all publishers that appear in DBLP data. We could not locate the website of a small number of (arguably local or out of business) publishers. We assigned them an arbitrary URI using the "example.org" domain name as prefix. In addition, another dataset is of information about 'Series' such as Lecture Notes in Computer Science and CEUR Workshops. This small dataset of around 100 series entities was created manually to facilitate the creation of 'in series' relationships based on a lookup operation on literal values such as <series>Dagstuhl Seminar Proceedings</series>. Over 5,700 relationships were added from publication to series in

SwetoDblp. These datasets are all represented in RDF to allow for easy inclusion of synonyms. A lookup operation on the respective datasets is in most cases the key to establish relationships that enrich SwetoDblp. Figure 3 illustrates an example where a person entity has homepage from which the affiliation information is extracted. It also shows the case of two entities connected through *sameAs* relationships.

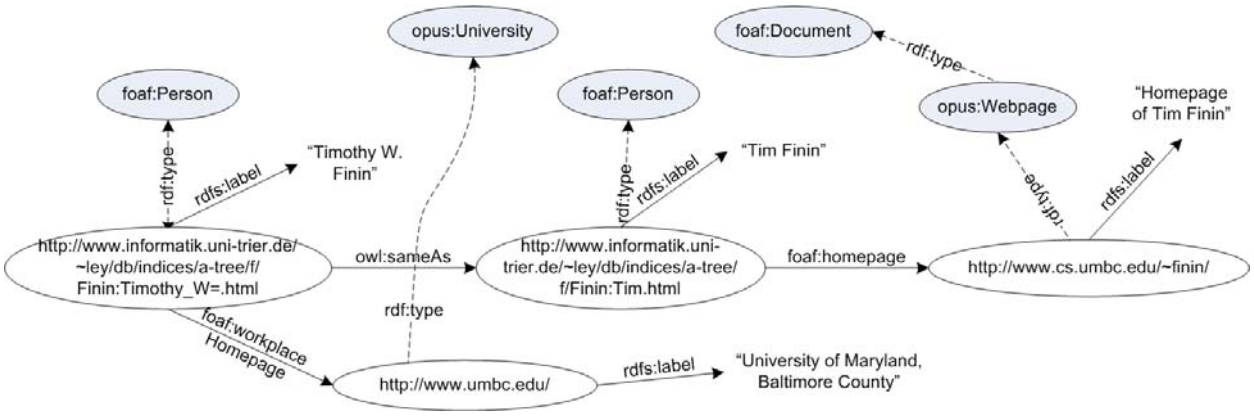

Figure 3: Example of Relationships in SwetoDblp Entities

SwetoDblp is publicly available for download together with additional datasets that were used for its creation (http://lsdis.cs.uga.edu/projects/semdis/swetodblp/) and other details are also available [10]. The additional datasets facilitated the integration and addition of many relationships and entities in SwetoDblp. Hence, incorporating other data sources enriches the resulting ontology. Although SwetoDblp is a relatively recent effort, it has been frequently used (or recognized) dataset in research involving populated ontologies [11][15][21][25][57][49][65].

2.3 Discovery, Analysis and Ranking of Relationships

A key element present in Semantic Web is that of relationships, which are a first-class object in RDF. Relationships provide the context (or meaning) of entities, depending on how they are interpreted and/or understood [104]. The value relies on the fact that they are *named* relationships. That is, they refer to a 'type' defined in an ontology. Relationships will 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 [90]. The SemDIS project at the LSDIS Lab

(http://lsdis.cs.uga.edu/projects/semdis/) builds upon the value of relationships on the Semantic Web. 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 [13]. Each semantic association can be viewed as a simple path consisting of one or more relationships, or, pairs of paths in the case of semantic similarity. Figure 4 illustrates a small graph of entities and the results of a query for semantic associations taking two of them as input.

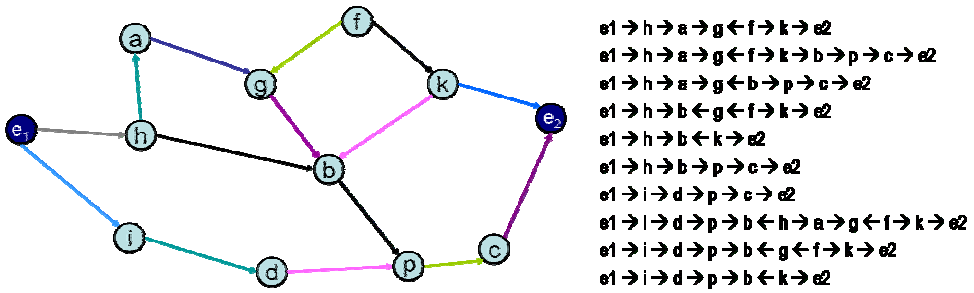

Figure 4: Example Semantic Associations from a small graph

Most useful semantic associations involve some intermediate entities and associations. Relationships that span several entities may be very important in domains such as national security, because this may enable analysts to see the connections between disparate people, places and events. In fact, applications that utilized the concept of semantic associations include search of biological terms in patent databases [72], provenance and trust of data sources [33], and national security [30][93]. The applicability of semantic associations in my research comes from the need to analyze relationships.

The type of operations needed to discover semantic associations involve graph-based traversals. It has been noted that graph-based algorithms help analysts of information to understand relationships between the various entities participating in events, activities, and so on [28]. The underlying technical challenge is also related to the common *connecting-the-dots* applications that are found in a broad variety of fields, including regulatory compliance, intelligence and national security [51] and drug discovery [61]. Additionally, techniques that use semantic associations have been applied for Peer-to-Peer (P2P) discovery of data and knowledge aggregation [7][76]. For example, a P2P approach was proposed to

make the discovery of knowledge more dynamic, flexible, and scalable [7]. Since different peers may have knowledge of related entities and relationships, they can be interconnected in order to provide a solution for a scientific problem and/or to discover new knowledge by means of composing knowledge of the otherwise isolated peers.

Ranking of semantic associations has been addressed by our colleagues taking the approach of letting the user choose among discovery mode and conventional mode of discovery/ranking of relationships [14]. They considered rare vs. common appearances of relationships in a populated ontology.

Research in the area of ranking semantic relations also includes [66][96], where the notion of "semantic ranking" is presented to rank queries returned within semantic Web portals. Their technique reinterprets 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.

Ontocopi is an application that identifies communities of practice by analyzing ontologies of different domains [2]. Ontocopi discovers and clusters related instances by following paths not explicit between them. Their work differs from ours in the dataset size. We aim at large scale algorithms that take advantage of the large metadata extracted from data sources.

The problem of finding relevant information has been approached with social networks [106]. Agents search data, based on referral-graphs that get updated according to answers received as well as the discovered connections to other agents that they are referred to. Their approach to efficient search in the network differs with our approach mainly because we try to get multiple paths connecting entities of interest whereas their approach aims at locating relevant information.

2.4 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. The result of semantically annotating a document is a set of explicit assertions indicating named-entities within them. Such assertions can be embedded within the document or be placed in a separate document. We implemented a semantic annotation component that identifies named entities from the ontology and keeps track of their position and offset in the text, their type (i.e., class/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 classes 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). An early prototype of this semantic annotation component was used for annotation and browsing in most of the content of the website of the 2006 International Semantic Web Conference (http://iswc2006.semanticweb.org/).

In earlier work, semantic annotation was critical for an application addressing how semantics can help in the Document-Access problem of Insider Threat [6]. The semantic annotation process was performed using the Semagix Freedom toolkit. Freedom is based on technology developed at and licensed from the LSDIS Lab [88]. The Semantic Enhancement Engine [44] of Freedom was used for automatic semantic annotation of a small (i.e., 1K) collection of documents. The indexing of these documents was done separate from the Freedom toolkit and included keeping track of the named-entities spotted by the semantic annotation process. In fact, the experiences developing such applications lead to investigate other options in respect to architectures/frameworks for processing unstructured data, as explained in the next section.

2.5 Unstructured Information Management

There are various architectures available for implementing new techniques in or related to search technology. We considered a few of the non-commercial solutions that have capabilities for indexing and retrieval: Lucene (http://lucene.apache.org), the KIM platform [77], and UIMA, which was open-sourced by IBM. We selected UIMA (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. For the purposes of indexing, one or more annotators can be run across a document collection and the results of all or specific annotators can be indexed. Throughout this paper we mention UIMA's features that are applicable in this work; extensive details on such framework are available elsewhere [37]. UIMA provides a robust framework for text analysis tools, indexing and retrieval and in terms of scalability, it has demonstrated success processing 11 million abstracts [102]. The semantic annotation capability in our approach was extended from our earlier work [6][8] and implemented for the UIMA framework.

A significant challenge with the Web today is the lack of explicit semantic information about data and objects being presented in web pages. Techniques such as Microformats (http://microformats.org) or RDFa (http://www.w3.org/TR/xhtml-rdfa-primer/) are gaining popularity but it might still take some time for wider adoption. In the future, large-scale adoption of this type of semantic annotation could facilitate the processing of documents for semantic-based search methods. A possible yet different way of annotation could be by using off-the-shelf toolkits for term identification. For example, Yahoo! Term Extraction (developer.yahoo.com) identifies phrases and terms from a given input text.

2.6 Semantic Search

The term *semantic search* is commonly used when semantics are used for improving search results. Existing semantic search approaches include concept-based search in documents [26][38] and entity-based search [42]. Most techniques rely on some form of preprocessing or indexing of documents that summarize, extract or glean semantics [68][84][79]. Guha explains various forms of semantic search [40]. The method described in this thesis fits in the category of entity-based search. A key difference with many link-analysis algorithms is that my approach does not require that the documents be interlinked, as it is the case for Web documents. Methods such as PageRank [23] 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 and/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 [81]. 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 [72]. 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 extending/incorporating link analysis techniques to determine popular entities within a populated ontology [74]. 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. The drawback of this method is also that non-important entities might not appear high in the results.

Taalee's Audio/Video Semantic Search Engine called MediaAnywhere was perhaps the first semantic search technology and commercial offering [100]. The following Enterprise Semantic Application development platform called Freedom [88] provided a comprehensive toolkit for crawling and extraction of metadata that can be used to build an ontology. Attribute-based search is then possible on the extracted pages, whereby results can be shown to user depending on the type of the entities

extracted from the web. Freedom also supports semantic annotation yet for the purposes of the research described in this thesis, it was desirable to use a non-commercial platform to build upon.

Semantic Search in UIMA. Let us use a simple example to explain existing semantic search capability in UIMA. We built on top of this capability for implementation of the ranking method described in this thesis. Keyword-based queries are supported in the traditional way in UIMA, but it also includes a semantic-search capability that can receive queries specifying that a keyword should match some annotation in particular. Suppose that a UIMA annotator contains a list of governors of U.S. states and creates a 'governor-usa' annotation every time it finds the appearance of governor's name in a document. Let us refer to these queries as annotation-queries. Such queries follow an XML-like syntax where the tag is the name of the annotation of interest. For example, to find documents containing governors with 'arnold', the UIMA annotation-query would be: <governor-usa>arnold</governor-usa>. The challenge is of course, that of building annotators for many entities of interest. The details of how we extended UIMA search component to deal with ambiguous entities and rank the retrieved results according to the proposed relevance measure are explained in a subsequent chapter.

CHAPTER 3

FLEXIBLE APPROACH FOR RANKING COMPLEX RELATIONSHIPS

Ranking documents by considering their relevance to a query requires analysis of relationships. In this chapter, we describe our earlier research on respect to ranking semantic associations where a user-defined context is used to determine relevance [3]. A prototype demo of the ranking technique [43] was source of good feedback, which led us to revise in more detail the approach for ranking semantic associations (canned demo available online, http://lsdis.cs.uga.edu/Projects/SAI/ranking_demo/). Extended work on ranking of semantic associations included evaluations by human subjects as well as a revised ranking formula [5] with corresponding updated demo online that uses an ontology of about 35K entities (at http://lsdis.cs.uga.edu/projects/semdis/rankingAH/).

The main goal is to ease the process of analyzing metadata that was aggregated from different sources and enable users to uncover previously unknown and potentially interesting relations, namely, semantic associations [13]. A query to find relationships connecting two entities typically results in many paths. Because of the expected high number of paths, it is likely that many of them would be regarded as irrelevant with respect to the user's domain of interest. Thus, the semantic associations need to be filtered according to their perceived relevance. Also, a customizable criterion needs to be imposed upon the paths representing semantic associations to focus only on relevant associations. Additionally, the user should be presented with a ranked list of resulting paths to enable a more efficient analysis.

To determine the relevance of semantic associations it is necessary to capture the context within which they are going to be interpreted and used (or the domains of the user interest). For example, consider a sub-graph of an RDF graph representing two soccer players who belong to the same team and who also started a new restaurant together. If the user is just interested in the sports domain the semantic associations involving restaurant related information could be regarded as irrelevant (or ranked lower). This can be accomplished by enabling a user to browse the ontology and mark a region (sub-graph) of

nodes and/or properties of interest. If the discovery process finds some associations passing through these regions then they are considered relevant, while other associations are ranked lower or discarded.

3.1 Ranking Criteria

The ranking process needs to take into consideration a number of criteria which can distinguish among associations which are perceived as more and less meaningful, more and less distant, more and less trusted, etc. The ranking score assigned to a particular semantic association is defined as a function of these parameters. Furthermore different weights can be given to different parameters according to users' preferences (e.g., trust could be given more weight than others).

Context Definition. We define a region of interest as a subset of classes (entities) and properties of a schema. We have considered class level and property level. Within the Class level, an "Organization" class may be considered relevant together with subclasses "PoliticalOrganization", "Financial-Organization" and "TerroristOrganization", but a class "Account" that is parent of the class "CorporateAccount" may not be of importance. At a Property level, we can specify restrictions as indication of which classes the property can be applied to ("domain" in RDFS) as well as which classes a property points to ("range" in RDFS). An example is a property "involvedIn" with a domain "Organization" and range "Event" (that is, Organization → involvedIn → Event). A user can define several ontological regions with different weights to specify the association types s/he is interested in. Hence, we define a *context* as a set of user defined *regions* of interest.

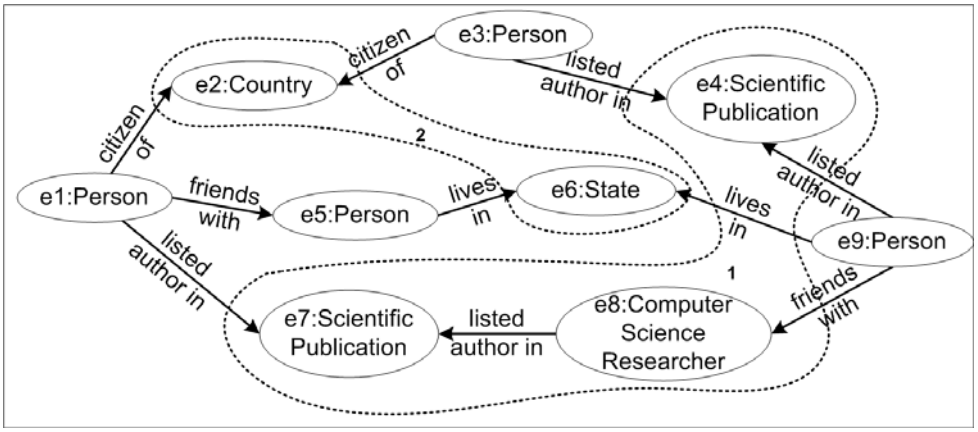

Figure 5: Context Example

To illustrate our approach, consider three sample associations between two entities as depicted at Figure 5, where a user has specified a contextual *region 1* containing classes 'Scientific Publication' and 'Computer Science Researcher'. Additionally, assume the user specified *region 2* containing classes 'Country' and 'State'. The resulting regions, 1 and 2, refer to the computer science research and geographic domains, respectively. For the associations at Figure 5, (with say, weights 0.8 and 0.2 for regions 1 and 2, respectively), the bottom-most association would have the highest rank because all of its entities and relationships are in the region with highest weight. The second ranked association would be the association at the top of the figure because it has an entity in *region 1*, but (unlike the association in the middle) also has an entity in *region 2*.

Before formally presenting the ranking criteria, we introduce notation used throughout the paper. Let A represent a Semantic Association, that is, a path sequence consisting of nodes (entities) and edges (relationships) that connects the two entities. Let *length*(A) be the number of entities and relationships of A. Let R_i represent the region i, that is, the set of classes and relationships that capture a domain of interest. Given that both entities and relationships contribute to ranking, let c be a component of A (either an entity or a relationship). For example, c_1 and $c_{length(A)}$ correspond to the entities used in a query where A is one of the Semantic Associations results of the query. We define the following sets for convenience, using the notation $c \in R_i$ to represent whether the type (rdf:type) of c belongs to region R_i:

$$X_i = \{c \mid c \in R_i \wedge c \in A\} \quad (1), \qquad Z = \{c \mid (\forall i \mid 1 \le i \le n) c \notin R_i \wedge c \in A\}$$

where n is the number of regions in the query context. Thus, X_i is the set of components of A in the i^{th} region and Z is the set of components of A not in any contextual region. We now define the *Context* weight of a given association A, C_A, such that

$$C_A = \frac{1}{length(A)} \left(\sum_{i=1}^{n} (W_{R_i} \times |X_i|)\right) \times \left(1 - \frac{|Z|}{length(A)}\right) ,$$

where n is the number of regions, W_{Ri} is the weight for the i^{th} region.

Subsumption. Classes in an ontology that are lower in the hierarchy can be considered to be more specialized instances of those further up in the hierarchy. That is, they convey more detailed information and have more specific meaning. For example, an entity of type "Professor" conveys more meaning than an entity of type "Person". Hence, the intuition is to assign higher relevance based on *subsumption*. For example, in Figure 5, entity 'e8' will be given higher relevance than entity 'e5'.

We now define the *component subsumption weight* (*csw*) of the i^{th} component, c_i, in an association A such that

$$csw_i = \frac{H_{c_i}}{H_{depth}},$$

where H_{ci} is the position of component c_i in hierarchy H (the topmost class has a value of 1) and H_{depth} is the total height of the class/relationships hierarchy of the current branch. We now define the overall *Subsumption* weight of an association A such that

$$S_A = \prod_{i=1}^{length(A)} csw_i$$

Trust. Various entities and their relationships in a Semantic Association originate from different sources. Some of these sources may be more trusted than others (e.g., Reuters could be regarded as a more trusted source on international news than some other news organization). Thus, trust values need to be assigned to the meta-data extracted depending on its source. For the dataset we used, trust values were empirically assigned. When computing Trust weights of a Semantic Association, we follow this intuition: the strength of an association is only as strong as its weakest link. This approach has been commonly used in various security models and scenarios [16]. Let T_{ci} represent the assigned trust value (depending on its data source) of a component c_i. We define the *Trust weight* of an overall association A as:

$$T_A = \min(t_{c_i}).$$

Rarity. Given the size of current Semantic Web datasets, many relationships and entities of the same type exist. We believe that in some queries, rarely occurring entities and relationships can be considered more interesting. This is similar to the ideas presented in [63], where infrequently occurring relationships (e.g., rare events) are considered to be more interesting than commonly occurring ones. In some queries however, the opposite may be true. For example, in the context of money laundering, often individuals engage in common case transactions to avoid detection. In this case, common looking (not rare) transactions are used to launder funds so that the financial movements will go overlooked [12]. Thus the user should determine, depending upon the query, his/her *Rarity* weight preference.

We define the *Rarity* rank of an association A, in terms of the rarity of the components within A. First, let *K* represent the knowledge base in the ontology (all entities and relationships of the instance population of the ontology). Now, we define the *component rarity* of the i^{th} component, c_i, in *A* as rar_i such that

$$rar_i = \frac{|M| - |N|}{|M|}, \text{ where}$$

$M = \{res \mid res \in K\}$ (all entities and relationships in *K*), and

$$N = \{res_j \mid res_j \in K \wedge typeOf(res_j) = typeOf(c_i)\},$$

with the restriction that in the case res_j and c_i are both relationships (i.e., of type rdf:Property), the subject and object of c_i and res_j must have a same type in the ontology. Thus rar_i captures the frequency of occurrence of component c_i, with respect to the entire ontology. Then, the overall *Rarity* weight, R_A, of an association *A*, as a function of its the components, is

$$R_A = \frac{1}{length(A)} \times \sum_{i=1}^{length(A)} rar_i \quad \textbf{(a)}; \qquad R_A = 1 - \frac{1}{length(A)} \times \sum_{i=1}^{length(A)} rar_i \quad \textbf{(b)},$$

where *length*(*A*) is the number of components in *A*. If a user wants to favor rare associations, (a) is used; in contrast, if a user wants to favor more common associations (b) is used. Thus, R_A is essentially the average *Rarity* of all components in *A* (or commonality if rare associations are not favored).

Popularity. When investigating the entities in an association, it is apparent that some entities have more incoming and outgoing relationships than others. Somewhat similar to Kleinberg's ranking algorithm [56], as well as the PageRank [23] algorithm used by Google, our approach takes into consideration the number incoming and outgoing relationships of entities. In some queries, associations with entities that have a high *Popularity* may be more relevant. These entities can be thought of as hotspots. For example, authors with many publications would have high popularity. In certain queries, associations that pass through these hotspots could be considered very relevant. Yet, in other queries, one may want to rank very popular entities lower. For example, entities of type 'Country' may have an extremely high number of incoming and outgoing relationships yet they might not add much relevance to a particular query. We define the Popularity of an association in terms of the popularity of its entities, namely, the entity popularity p_i, of the i^{th} entity e_i, in association A as:

$$p_i = \frac{|\ pop_{e_i}\ |}{\max_{1 \le j \le n}(|\ pop_{e_j}\ |)} \text{ where } typeOf(e_i) = typeOf(e_j)$$

where n is the total number of entities in the populated ontology. Thus, pop_{ei} is the set of incoming and outgoing relationships of e_i and $\max_{1 \le j \le n}(|\ pop_{e_j}\ |)$ represents the size of the largest such set among all entities in the ontology of the same class as e_i. Thus p_i captures the *Popularity* of e_i, with respect to the all other entities of its same type. The overall *Popularity* weight P of an association A is defined such that

$$P_A = \frac{1}{n} \times \sum_{i=1}^{n} p_i \ \textbf{(a)}; \qquad P_A = 1 - \frac{1}{n} \times \sum_{i=1}^{n} p_i \ \textbf{(b)},$$

where n is the number of entities (nodes) in A and p_i is the entity popularity of the i^{th} entity in A. If a user wants to favor popular associations, (a) is used; in contrast, if a user wants to favor less popular associations then (b) is used. Thus, P_A is essentially the average *Popularity* or *non-Popularity* of all entities in A.

Association Length. In some queries, a user may be interested in more direct associations (i.e., shorter associations). Yet in other cases a user may wish to find indirect or longer associations. Long paths may be more significant in the domain where there may be deliberate attempts to hide relationships. For example, potential terrorist cells remain distant and avoid direct contact with one another in order to defer possible detection [58], also, money laundering involves deliberate innocuous looking transactions that may change several hands. Hence, the user can determine which *Association Length* influence, if any, should be used.

We define the *Association Length* weight L, of an association A. If a user wants to favor shorter associations (a) is used, otherwise (b) is used.

$$L_A = \frac{1}{length(A)} \textbf{ (a)}; \qquad L_A = 1 - \frac{1}{length(A)} \textbf{ (b)}.$$

Overall Ranking Criterion. The overall association Rank using the before mentioned criteria, is defined as

$$W_A = k_1 \times C_A + k_2 \times S_A + k_3 \times T_A + k_4 \times R_A + k_5 \times P_A + k_6 \times L_A$$

where k_i $(1 \leq i \leq 6)$ add up to 1.0 and is intended to allow fine-tuning of the ranking criteria (e.g., popularity can be given more weight than association length). This provides a flexible, query dependant ranking approach to assess the overall relevance of associations.

3.2 Evaluation of Ranking of Semantic Associations

The prototype application consists of the components illustrated in Figure 6. We modified the TouchGraph (touchgraph.com) applet for visual interaction with a graph to define a query context. Prior to a query, a user can define contextual regions of the visualized ontology, with their associated weights. Unranked associations are passed from the query processor to the ranking module. The associations are then ranked according to the ranking criteria defined by the user.

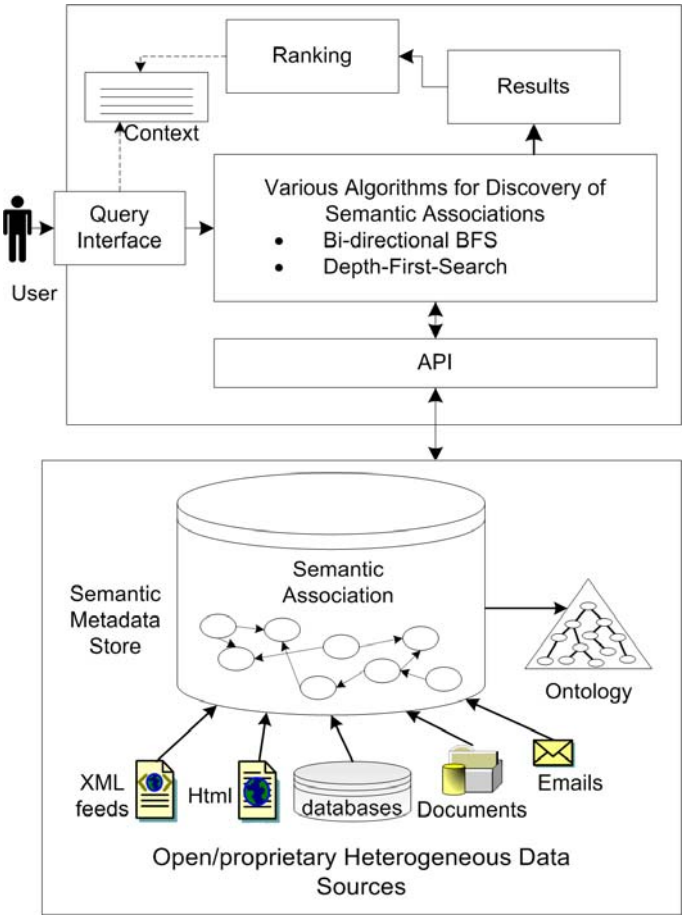

Figure 6: System Architecture (Ranking Components)

The Web-based user interface allows the user to specify entities on which Semantic Association queries are performed. Optionally, the user can customize the ranking criteria by assigning weights to each individual ranking criterion. The version of SWETO ontology [4] used for the evaluation contains a majority of instances including cities, countries, airports, events (such as terrorist events), companies, banks, persons, researchers, organizations, and scientific publications, among others.

Due to the various ways to interpret Semantic Associations, we evaluated the results with respect to those obtained by a panel of five human subjects, graduate students in computer science and not familiar with the research presented here. The human subjects were given (randomized) query results from different Semantic Association queries (each consisting of approximately 50 results where the longest associations were of length 12). Together with the results, all subjects were provided with the

ranking criteria for each query (i.e., context, whether to favor short/long, rare/common associations, etc.). The human subjects were also provided with the type(s) of the entities and relationships in the associations, thus allowing them to judge whether an association was relevant to the provided context. They then ranked the associations based on this modeled interest and emphasized criterion. Given that the human subjects assigned different ranks to the same association, their average rank was used as a reference (target match).

There is a number of ways that the criteria can be customized (e.g., favor long and rare vs. short and popular associations), for which we evaluated five combinations. This is a small set, yet arguably it is a representative sample of these combinations. In each of the test queries, we have emphasized (highly weighted) two of the criteria. The following list presents the ranking criteria and broader impact of each query.

Table 1: Queries in the evaluation and scenario/applicability

Query	Query Details	Scenario / Applicability
1	Between two entities of type '*Person*', with context of collegiate departments ('*University*', '*Academic Department*', etc.); favors rare components	Illustrates how the ranking approach can capture a user's interest in rare associations within a specific domain
2	Between two entities of type '*Person*'. Favors short associations in the context of computer science research	Demonstrates the ability to capture the user interest in finding more direct connections (i.e., collaboration in a research project/area)
3	Between a '*Person*' and a '*University*', where common (not rare) associations are highly weighted and in the context of mathematics (departments and professors)	Shows the flexibility to highlight common relationships. This may be relevant, when trying to model the way a person relates to others in a similarly as the common public
4	Between a '*Person*' and a '*Financial Organization*'; long associations and the financial domain context are favored	Generally relevant for semantic analytics applications, such as those involving money laundering detection [58]
5	Between two '*Persons*'; unpopular entities and the context of geographic locations are favored	Demonstrates the system's capability to filter non relevant results which pass through highly connected entities, such as countries

Figure 7 illustrates the number of Semantic Associations in the intersection of the top *k* system and human-ranked results. This shows the general relationship between the system and human-ranked associations. Note that the plot titled 'Ideal Rank' demonstrates the ideal relationship, in which the intersection equals *k* (e.g., all of the top five system-ranked associations are included within the top five human-ranked associations).

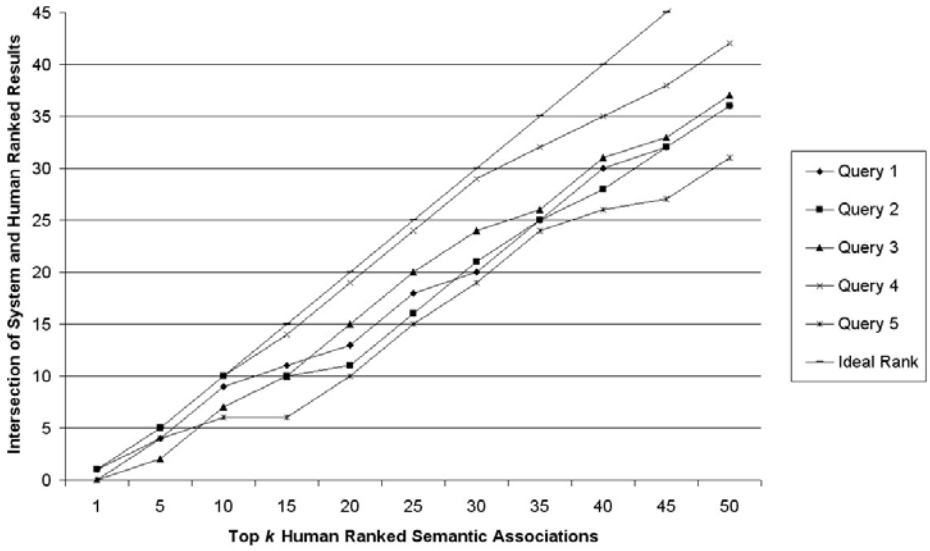

Figure 7: Intersection of (top *k*) Human and System Rankings

An interesting aspect in the evaluation was to gain insight on the disagreement between human-ranked results. Figure 8 shows the agreement of human subject, their average, and ranking by the system. The x-axis represents semantic associations that are ranked first, second, etc. according to average rank scores of human subjects. The x-axis does not contain actual rank scores, but their corresponding ordering. On the other hand, the y-axis represents rank scores given by the system and human subjects. It is evident that there are varying levels of disagreement in human subjects ranking. The system's ranking falls within the range of ranking disagreement of human subjects (the Spearman's Footrule distance measure of the system rankings with respect to average users' rankings of 0.23).

The minimum average distance of the system assigned ranks from that of the human subject's for a query (considered in relative order) was 0.55, while the maximum never exceeded 4. The results are promising, given that out of the top ten human-ranked results, the system averaged 8.4 matches.

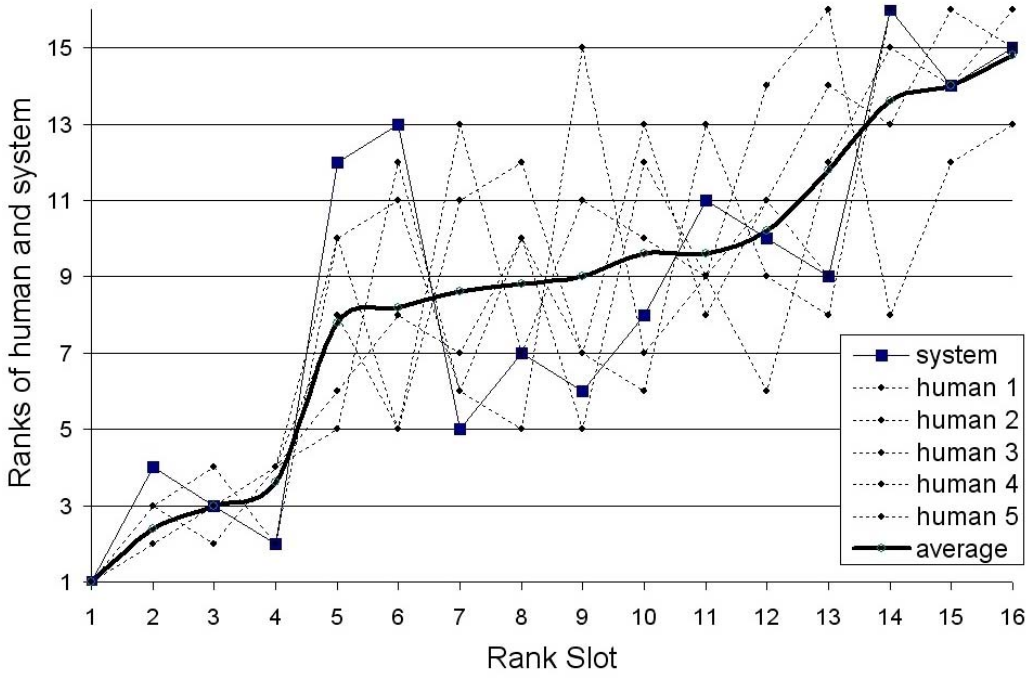

Figure 8: Human subject's agreement and ranking by the system

3.3 Observations

In three out of the five queries, the top human-ranked association directly matched the system assigned rank. Additionally, the top human-ranked association fell within the top five system-ranked associations in all five queries. There exists disagreement in the ranking of human subjects themselves. The ranking by the system falls within the range of ranking disagreement of human subjects. Hence, these results demonstrate the potential of the ranking algorithm and suggest that the approach is flexible enough to capture a user's preference and relevantly rank these complex relationships.

The most significant ranking criteria contributing to results as expected from user was that of *context*. This can be better illustrated by an example consisting of a semantic association query between entities *George W. Bush* and *Jeb Bush* where the context is defined using the concept *Actor* from the ontology. In the results of the query, various semantic associations include the now politician *Arnold Schwarzenegger*, who exists in the ontology under both concepts *Politician* and *Actor*. Figure 9 is a screenshot of results where, as expected, top ranked associations between *George W. Bush* and *Jeb Bush* do in fact include *Schwarzenegger* because he fits the *Actor* context.

Figure 9: Sample Ranking Results for Context

Direct applicability of these ranking techniques includes ranking semantic associations from a dataset of events and venues containing geo-spatial features [45][46].

CHAPTER 4

ANALYSIS OF RELEVANT ASSOCIATIONS BETWEEN ENTITIES

The methods presented in the previous chapter dealt with finding semantic associations containing up to five relationships (i.e., a path of maximum 5 edges). In this chapter, I describe how discovery of longer associations can be improved by incorporating an *analysis* step that could discard or shorten certain paths. In an earlier application of semantic associations for National Security, we found that it was common to obtain results of hundreds of semantic associations connecting two entities in an ontology [93]. In fact, processing large ontologies demands fast data access and facilities for the type of graph-traversal operations involved in discovery and analysis of semantic associations. The implementation of techniques in this thesis that use ontologies of over 1 million entities relies upon BRAHMS, which is a main-memory RDF-database [53] developed at the LSDIS Lab.

4.1 Semantic Analytics: The case of Conflict of Interest Detection

We use the case of Conflict of Interest to demonstrate how analysis of associations takes place by considering strength of relationships connecting entities. This method takes into account a variety of different relationships among entities to measure relevance (such as same-affiliation and co-editorship relationships). In my research, an application for the detection of Conflict of Interest uses semantic analytics on social networks [9] to demonstrated (and explain) the challenges of bringing together a semantic & semi-structured social network (FOAF) with a social network extracted from the collaborative network in DBLP.

Conflict of Interest. Conflict of Interest (COI) is a situation where bias can exist or be perceived based on the relationships or connections of the participants involved either explicitly or implicitly. The connections between participants could come from various origins such as family ties, business or friendship ties and confidential information. Detecting COI is required to ensure “fair-play” in many decision-making situations such as contract allocation, IPO (Initial Public Offerings) or company

acquisitions, corporate law and peer-review of scientific research papers or proposals. Detection of COI is also critical where ethical and legal ramifications could be quite damaging to individuals or organizations.

The detection of COI usually involves analysis of social network data, which is hard to obtain due to privacy concerns. The case of academic research does not involve much of a privacy concern because researchers are open to expose their identity in published research (listing collaborators) and in their participation on the research community, e.g., as reviewers or organizers of conferences. Social and collaborative information is widely published via various media such as magazines, journals and the Web.

Social Networks: Graphs of Person Entities. In particular, the advance of Web technologies has facilitated the access to social information not only homepages of persons and hyperlinks but also via many social networking sites. Social networking websites attract more and more people to contribute and share information. For example, the LinkedIn social network comprises a large number of people from information technology areas and it could be used to detect COI in situations such as IPO or company acquisitions. MySpace, Friendster, Orkut and Hi5 contain data that could substantiate COI in situations of friendship or personal ties. The list keeps growing. Facebook was targeted to college students but it has expanded to include high-school students and now it is open to anyone. Club Nexus is an online community serving over 2000 Stanford undergraduate and graduate students [1]. The creation of Yahoo! 360° and the acquisition of Dodgeball by Google are relatively recent examples where the importance of social network applications is evident not only considering the millions of users that some of them have but also due to the (even hundreds of) millions of dollars they are worth. Hence, it is not surprising that social network websites do not openly share their data. Other reasons for not doing so include privacy concerns. In some sites, the true identity of users is available only to their connections in the same network (e.g., Facebook, LinkedIn). Other sites such as LiveJournal publish the social connections of users openly yet the true identity of users is (in most cases) hidden behind a nickname.

Although social network websites can provide data to detect COI, they are isolated even when their users might overlap a lot. That is, many people have accounts in more than one site. It was estimated that 15% of the users overlapped in two social networks [64]. Moreover, much of the social information is

still hosted in the distributed homepage-hyperlink style. Therefore, our case of demonstrating COI detection faces a big challenge: integration of different social networks. The Friend-of-a-Friend (FOAF) vocabulary can be used to publish information about persons, their relationships to workplaces and projects, and their social relations. The aggregation of such FOAF documents by means of the "knows" relationship of the FOAF vocabulary results in a social network. In our previous work [9], we integrated a network of FOAF documents with a second network, the DBLP bibliography (dblp.uni-trier.de/), which provides collaboration network data by virtue of the explicit co-author relationships among authors. Although there were significant challenges for the integration of the two networks, such *integration* is not part of the contributions presented in this thesis. Instead, the focus is to describe how relationships were analyzed, as this was a completely different method as presented in the previous chapter. Interested readers on the disambiguation method are referred to [9].

4.2 Analysis of Relationships between Entities in a Social Network

From the perspective of detection of COI, each of the relationships among the people in a social network needs to be analyzed. However, by adhering to a strict definition of COI, there is only one situation in which there exists a conflict of interest: the existence of a strong and direct relationship. The aim is that human involvement can be drastically reduced but will still be relevant in other cases, such as when the quality of data is not perfect, the domain is not perfectly modeled and when there is no complete data. The subjective nature of the problem of COI detection is a good example where Semantic Web techniques cannot be expected to be fully automatic in providing the correct solution. For these reasons, we use the notion of potential COI as it applies to cases where evidence exists to justify an estimated level of "low," "medium," or "high" degree of possible COI, as illustrated in Table 2.

Table 2: Levels of Conflict of Interest (between persons in a social network)

Type	Level	Remarks
Definite COI	Highest	Sufficient evidence exists to require participant to abstain (i.e., recuse)
Potential COI	High	Evidence justifies additional verifications; participant is suggested to recuse
	Medium	Little evidence of potential COI
	Low	Shallow evidence of potential COI, which in most cases can be ignored

4.3 Measuring Strength of Relationships

We implemented two techniques. In the first, a preprocessing step quantified the strength of relationships between people. Weights were represented by means of reified statements. It has been noted elsewhere that the dataset size can drastically increase due to the verbosity of the XML serialization of RDF to represent reified statements [69]. This would have an even larger impact on large datasets. In the second technique, we take a different approach that consists of computing strength of relationships at execution time. There are various types of relationships being considered for detection of COI. The basic ones are FOAF *knows* and DBLP *co-author*.

The strength of relationships is captured by weights between 0 and 1, where 1 refers to maximum strength. The relationship *foaf:knows* is used to explicitly list the person that are known to someone. These assertions can be weighted depending upon the provenance, quality and/or reputation of their sources. On the other hand, the assertion of the *foaf:knows* relationship is usually subjective and imperfect. For example, *foaf:knows* from *A* to *B* can be indicative of potential positive bias from *A* to *B* yet it does not necessarily imply a reciprocal relationship from *B* to *A*. Hence, we assigned a weight of 0.45 to all *foaf:knows* relationships in the FOAF dataset. The cases where a *foaf:knows* relationship exists in both directions have a weight of 0.9.

Another type of relationship we used is the *co-author* relationship, which is a good indicator for collaboration and/or social interactions among authors. However, counter examples can be found against assumptions such as "one researcher always has a positive bias towards his/her collaborator" because friendship or positive opinion is not necessary for performing collaborative research. A more reasonable indicator of potential bias is the frequency of collaboration, which we use to compute weights of *co-author* relationships. In the first technique, we used the ratio of number of co-authored publications vs. total of his/her publications as the weight for the *co-author* relationship. However, such measure resulted in relatively low weights for co-authors of researchers that have many publications. For example, a researcher with over 100 publications had a very low co-authorship weight with few of his doctoral students with whom has co-authored very few papers. Therefore, in the second technique we make use of

a different measure of collaboration strength that takes into account the number of authors in a paper as well as the number of papers two people co-authored [73]. The formula adds a weight of 1/(n-1) to the collaboration strength of two authors for each paper they co-authored together (where n is the number of authors in a paper). This measure captures quite well the cases where a paper has very few authors based on the assumption that their collaboration strength is higher than in the case of papers with a large number of co-authors. The computed collaboration strength for any two co-authors is symmetric.

Discovery of Relationships. Obtaining the semantic associations connecting two entities using currently available RDF query languages has disadvantages given that a semantic association is basically a path between two entities. For example, six queries are required to find all paths of up to length two connecting two entities [53]. In other applications, such as anti-money laundering, it is necessary to process longer paths [12]. We looked for semantic associations containing up to 4 relationships. This is due to the fact that the data contain implicit information about co-authorship in the form of two author entities being connected to a publication (by an intermediate RDF blank node that maintains the ordered sequence of the authors in papers). At execution time, semantic associations are reduced into shorter relationships such as co-author and same-affiliation (using some heuristics). The benefit of this is a level of abstraction whereby the algorithm is not concerned with representation details such as blank nodes. Hence, the work needed adapt an application for usage of different datasets would not be significant.

The algorithm works as follows. First, it finds all semantic associations between two entities. Second, each of the semantic associations found is analyzed to collapse it if applicable (as explained before) and then the strength of its individual relationships is computed. Since each semantic association is analyzed independently of the others, all directions of the different relationships are eventually considered.

4.4 Evaluation: Scenario of COI Detection in Peer-Review Setting

The dataset consisted of DBLP and FOAF data. The SwetoDblp ontology [10] provided the DBLP data in RDF (we used the March-2007 version). It consists of metadata of over 800K publications, including over 520K authors thereof. The FOAF data consisted of about 580K persons linked through

519K *foaf:knows* relationships. The disambiguation process produced close to 2,400 relationships establishing same-as relationships in the integrated dataset. There are 4,478,329 triples between entities and 7,510,080 triples between entities and literal values. The dataset size in terms of disk space was of approximately 845 MB of DBLP data and 250 MB of FOAF data.

We utilized BRAHMS RDF database for building the prototype as it was designed for this type of connecting-the-dots applications [53]. BRAHMS creates a snapshot file for fast loading as main-memory database in subsequent usage. It took about 50 seconds to load our integrated dataset. All tests were performed on an Intel-based laptop with 2 GB of RAM running OSX 10.4. This shows that building this type of application is feasible without the need of expensive or sophisticated equipment such as dedicated servers or 64-bit architectures. The datasets used, the source code ant the evaluation test cases (explained in the next section) are available online (http://lsdis.cs.uga.edu/projects/semdis/coi/). In addition, we want to point out that the development was initially done with the main-memory implementation of the SemDis API (http://lsdis.cs.uga.edu/projects/semdis/api/). This API was built as part of the SemDis project of the LSDIS Lab. The (Java) main-memory implementation uses the RDF parser included in Jena [67]. This implementation can easily handle datasets of around 50 MB file size. Hence, it is convenient to use during development and since the Java-bindings of BRAHMS implement the same API, then switching over to use BRAHMS is straightforward.

We will focus on the scenario of peer-review process for scientific research papers. Semi-automated tools such as conference management systems commonly support this process. In a typical conference, (typically) one person designated as Program Committee (PC) Chair, is in charge of the proper assignment of papers to be reviewed by PC members of the conference. State-of-the-art conference management systems support this task by relying on reviewers specifying their expertise and/or "bidding" on papers. These systems can then assign papers to reviewers and also allow the Chair to modify these assignments. A key task is to ensure that there are qualified reviewers for a paper and that they will not have a-priori bias for or against the paper. Conference management systems can rely on the knowledge of the Chair about any particular strong social relationships that might point to possible COIs. However, due

to the proliferation of interdisciplinary research, the Chair cannot be expected to keep up with the ever-changing landscape of collaborative relationships among researchers, let alone their personal relationships. Our method considered the following cases by means of analysis of relationships between an author of a paper and a potential reviewer.

1. *Reviewer and author are directly related* (through *foaf:knows* and/or *co-author*). The assessments of potential level of COI is set to "high" regardless of the value of collaboration strength. The rationale behind this is that even a one-time collaboration could be sufficient reason for COI since it might have come from collaborating in a significant publication. Direct relationships through a same-affiliation relationship are given a "medium" potential COI level since it does not imply that the reviewer and author know each other. For example, some affiliation information is not up to date in the available data.

2. *Reviewer and author are not directly related but they are related to one common person*. Let us refer to this common person as an intermediary. Thus, the semantic association contains two relationships. An assessment of "medium" is set for the case where there are strong relationships connecting to the intermediary person. Otherwise, the assessment is set to "low." In the scenario of peer-review process, a low level of potential COI can be ignored but in other situations it might have some relevance.

For evaluation with real-world data, we analyzed separately the accepted papers and Program Committee members of most tracks of the *2006 International World Wide Web Conference*. This choice was motivated by the lack of any benchmark for detection of COI, where human involvement is typically required to make final decisions. We selected this conference with the expectation that authors and reviewers in this field would be more likely to have made available some of their information using FOAF vocabulary. In addition, the organization of tracks in the *WWW Conference* facilitates evaluation due to their explicit grouping of papers per track where each track has a specific list of Program Committee members.

From Table 3 through Table 6, PC members and authors of the papers in our evaluation are listed for which a potential COI was detected. We do not show the obvious cases of definite COI where a PC member is author of a paper. Also, we do not show cases of 'low' potential COI since in the scenario of peer-review these could be ignored. The tables show authors for whom there was some level of COI detected but does not list authors for which the COI depends on another author. For example, a doctoral student typically has published only with his/her advisor and any detected COI passes through connections of the advisor. The different levels of COI detected are indicated on each cell containing a primary, and in some cases, a secondary level of COI. We compared our application with the COI detection approach of the Confious conference management system [75]. Confious utilizes first and last names to identify at least one co-authored paper in the past (between reviewers and authors of submitted papers). Confious thus misses COI situations that our application does not miss because ambiguous entities in DBLP are reconciled in our approach. Confious detects previous collaborations and raises a flag of possible COI. Our approach provides detailed information such as the level of potential COI as well as the cause. For example, our approach indicates that "Amit Sheth" and "Michael Uschold" have a "medium" level of potential COI due to co-editorship. Finally, compared to Confious, the results of our approach are enhanced by the relationships coming from the FOAF social network. However, in cases we tested there was no situation of two persons having a *foaf:knows* relationship and not having *co-author* or *co-editor* relationships between them.

The key of cell values in tables is as follows:

D: Definite COI: reviewer is one of the authors

Hc: High potential COI: due to previous co-authorship

Mcc: Medium potential COI: due to common collaborator

Ma: Medium potential COI: due to same-affiliation

Me: Medium potential COI: due to previous co-editorship

Table 3: Conflict of Interest Results – Browsers Track

WWW2006 Browsers Track	Krishna Bharat	Susan T. Dumais	Yoëlle S. Maarek	Paul P. Maglio	Andreas Paepcke	Dorée D. Seligmann	Terry Winograd
Prabhakar Raghavan	Hc	Hc	Hc		Ma		Ma
Alex Cozzi				Hc			
Jason Nieh						Ma	

Table 4: Conflict of Interest Results – E* Applications Track

WWW2006 E* Applications Track	John Domingue	Vincent P. Wade
Helen Ashman		Me
Amit P. Sheth	Hc	

Table 5: Conflict of Interest Results – Search Track

WWW2006 Wearch Track	Junghoo Cho	Monika Henzinger	Panagiotis G. Ipeirotis	Anna R. Karlin	Christopher Olston	Sridhar Rajagopalan	Andrew Tomkins
Farzin Maghoul Ravi Kumar						Hc Hc Mcc	Hc Hc Mcc
Ziv Bar-Yossef						Hc	Hc
Alexandros Ntoulas Marc Najork Mark Manasse	Hc	Hc	Hc	Hc	Hc		
Beverly Yang						Hc	
Soumen Chakrabarti	Hc				Hc	Hc	Hc

We manually verified the COI assessments for the tracks listed. In most cases our approach validated very well but in rare cases did not. For example, there is a 'high' level of potential COI between Amit Sheth and John Domingue due to co-authorship yet that particular case is from a 2-pages preface article in a Workshop organized by Drs. Sheth, Domingue and few others. A similar example is that of co-authors of Steffen Staab due to his *IEEE Internet Computing* column where one or more persons independently contribute with a section of the final article. In the resulting bibliography data of such

articles, all authors appear as co-authors although they did not really collaborate as in regular research papers. These cases (Table 6) illustrate the dependency on the quality of the underlying datasets and/or data representation details. We noticed that some researchers have high potential COI with a number of other people. We looked into the data to glean a reason for this. We found that researchers having over 50 publications listed in DBLP data tend to show up more frequently in COI results. This is more noticeable for researchers with over 150 publications (examples in the tables listed include Drs. Raghavan, Sheth, and Staab).

Table 6: Conflict of Interest Results – Semantic Web Track

WWW2006 Semantic Web Track	V. Richard Benjamins	John Davies	John Domingue	Frank van Harmelen	Enrico Motta	Steffen Staab	Michael Uschold
Mustafa Jarrar						Hc	
Peter F. Patel-Schneider Ian Horrocks				Hc Hc		Hc	Hc
Rudi Studer	Hc Mcc	Me		Hc	Hc	Hc Me	
Yolanda Gil	Me				Me		
Li Ding Amit P. Sheth Anupam Joshi Tim Finin	Hc		Hc			Hc Hc Me Hc Hc	Me

In addition to the evaluation with respect to conference tracks and their respective papers, we created a list of persons that appear in FOAF to evaluate COI detection on the FOAF part of the dataset. We randomly selected 200 FOAF person entities that are connected to at least another entity with a *foaf:knows* relationship. We evaluated them as factitious authors and reviewers. Table 7 illustrates a subset of the results that includes some researchers that also appear in the conference tracks listed before mentioned. The legend ‘Mcf’ indicates Medium potential level of COI due to common-friend; ‘Mcf’ indicates Medium potential level of COI due to common-friend. The difference between Low and Medium rating for common-friend is that for Medium level it is necessary that the *foaf:knows* relationship exists in both directions (i.e., from A to B and from B to A).

Table 7: Conflict of Interest Results – FOAF Persons and Reviewers

FOAF Person Entities	Craig Knoblock	Tim Finin	Yimin Wang	Lalana Kagal	Jos de Bruijn	Emmanuel Pietriga	Marcelo Tallis
Pat Hayes	Mcf	Mcf		Lcf	Lcf	Lcf	Lcf
Cartic Ramakrishnan		Lcf	Mcf	Mcf			Lcf
Rong Pan			Mcf	Mcf			Lcf

4.5 Observations

Finding semantic associations containing up to five relationships (i.e., a path of maximum 5 edges) not only has challenges with respect to efficient discovery but also for analysis of resulting associations. Discovery of longer associations can be improved through discarding or shortening certain paths based on the semantics of the sequences in the path.

One of the benefits of an ontology-based approach for analysis of relationships between entities is providing justification/explanation of the results by listing the semantic associations interconnecting the two entities. We measured the performance by excluding the time to load the dataset and dividing the remaining time by the number of pair-wise computations of COI detection (i.e., author and reviewer). On the average, it took 0.55 seconds to compute the COI between two persons. The majority of this timing is due to the search for the multiple semantic associations connecting them, of path length of up to 4 connections. Simple optimizations are possible such as starting the detection of COI with the authors that have published more papers. We identified some major stumbling blocks in building scalable applications that leverage semantics. In the next section we detail further insight in this respect.

4.6 Experiences Building large scale Semantic Web Applications

We take the opportunity to describe the common engineering and research challenges of building practical Semantic Web applications that use large-scale, real-world datasets. In fact, we have argued [9] that the success of this vision will be measured by how research in this field (i.e., theoretical) can contribute to increasing the deployment of Semantic Web applications [62]. In particular, we refer to Semantic Web applications that have been built to solve commercial world problems

[31][40][44][71][93][94][100]. The engineering process and development of a Semantic Web application typically involves a multi-step process. Figure 10 illustrates the multi-step process of building Semantic Web applications.

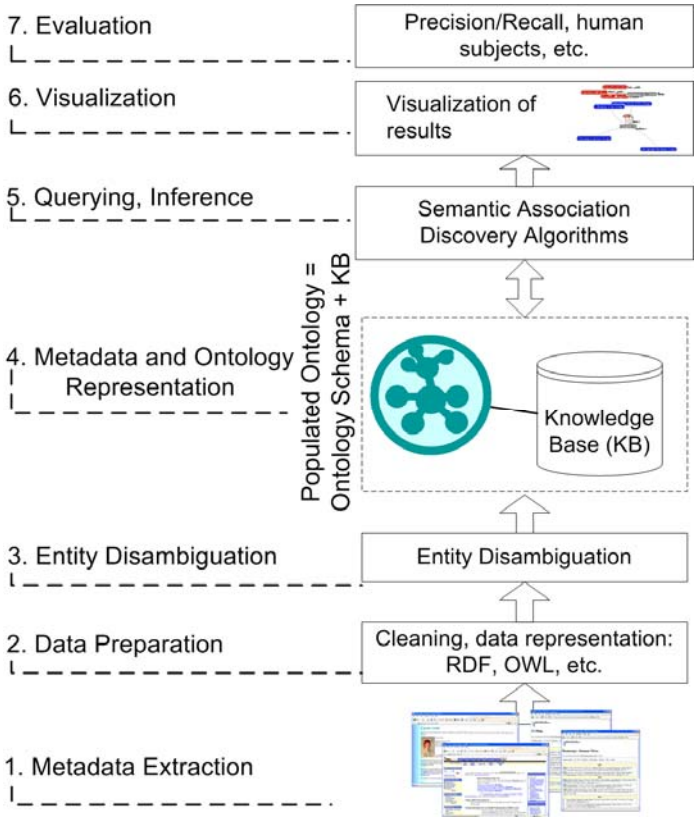

Figure 10: Multi-step process of building Semantic Web applications

1. *Obtaining high quality data*: Such data is often not available. Additionally, there might be many sites from which data is to be obtained. Thus, metadata extraction from multiple sources is often needed [29][59][88].

2. *Data preparation*: Preparation typically follows the obtaining of data. Cleanup and evaluation of the quality of the data is part of data preparation.

3. *Entity disambiguation*: This has been and continues to be a key research aspect and often involves a demanding engineering effort. Identifying the right entity is essential for semantic annotation and data integration (e.g., [19][49]).

4. *Metadata and ontology representation*: Depending on the application, it can be necessary to import or export data using standards such as RDF/RDFS and OWL. Addressing differences in modeling, representation and encodings can require significant effort.

5. *Querying and inference techniques*: These are needed as a foundation for more complex data processing and enabling semantic analytics and discovery (e.g., [13][52][54][88]).

6. *Visualization*: The ranking and presentation of query or discovery results are very critical for the success of Semantic Web applications. Users should be able to understand how inference or discovery is justified by the data (e.g., [30]).

7. *Evaluation*: Often benchmarks or gold standards are not available to measure the success of Semantic Web applications. A common option is comparing with results from human subjects.

We now list a few issues with the intention of shedding some light on the efforts required and available tools/research to build semantic applications that use large-scale, real-world datasets.

What does the Semantic Web offer today in terms of standards, techniques and tools? Technical recommendations, such as RDF(S) and OWL, provide the basis towards standard knowledge representation languages in Semantic Web. In addition, query languages (www.w3.org/TR/rdf-sparql-query/), path discovery techniques [13] and subgraph discovery techniques [78] are examples of existing techniques for analytical access on RDF data, including recent developments that address extensions to SPARQL for expressing arbitrary path queries [15][57]. With respect to data, the FOAF vocabulary has gained popularity for describing content (e.g., 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web, www.w3.org/2001/sw/Europe/events/foaf-galway). On the other hand, semantic annotation has been proven scalable [31] and supported by commercial products [44] gaining wider use.

What does it take to build scalable Semantic Web Applications today? As we have seen by addressing the problem of Conflict of Interest [9], building scalable Semantic Web applications is not a trivial task. At the current stage, development of these applications can be quite time consuming. As much as the Semantic Web is promoting automation, there is a lot of effort required in terms of manual efforts and in customization of existing techniques. The goal of full/complete automation is some years

away and it might be materialized in a different way as it was originally proposed [20]. Currently, quality and availability of data is often a challenge given the limited number of high quality and useful large-scale data sources. Significant work is required in certain tasks, such as entity disambiguation. Thus, it is not straightforward to develop scalable Semantic Web Applications because we cannot expect to have all the components readily available. Additionally, proving their effectiveness is a challenging job due to the lack of benchmarks. On the other hand, had the current advances not been available, some applications would not have been possible. For example, which other openly available social network other than FOAF could have been used? Then again, a number of tools are available today that can make the manual work less intensive. While conceptually there has been good progress, we are still in an early phase in the Semantic Web as far as realizing its value in a cost effective manner.

How are things likely to improve in the future? Standardization of vocabularies used to describe domain specific data is invaluable in building semantic applications. This can be seen in the bio-medical domain, e.g. the National Library of Medicine's MeSH (Medical Subject Heading) vocabulary, which is used to annotate scientific publications in the bio-medical domain. Further research in data extraction from unstructured sources will allow semi-automated creation of semi-structured data for specific domains (based on the vocabularies) for which analytic techniques can be applied to build semantic applications like the one described in this paper. Analytical techniques that draw upon graph mining, social network analysis and a vast body of research in querying semi-structured data, are all likely to facilitate the creation of Semantic Web applications. We expect that benchmarks will appear. In the future, there should be a large variety of tools available to facilitate tasks, such as entity disambiguation and annotation of documents.

CHAPTER 5

RANKING DOCUMENTS USING A RELEVANCE MEASURE OF RELATIONSHIPS

In previous chapters, various steps or components have been explained. These efforts provided insight on how to measure relevance by analyzing the relationships between entities. Such methods involve finding the list of associations between entities and then ranking them. The third component of this thesis is grounded on how to exploit semantic relationships of named-entities to improve relevance in search and ranking of documents. In this chapter, 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 using an 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. Figure 11 illustrates the schematic of the system architecture.

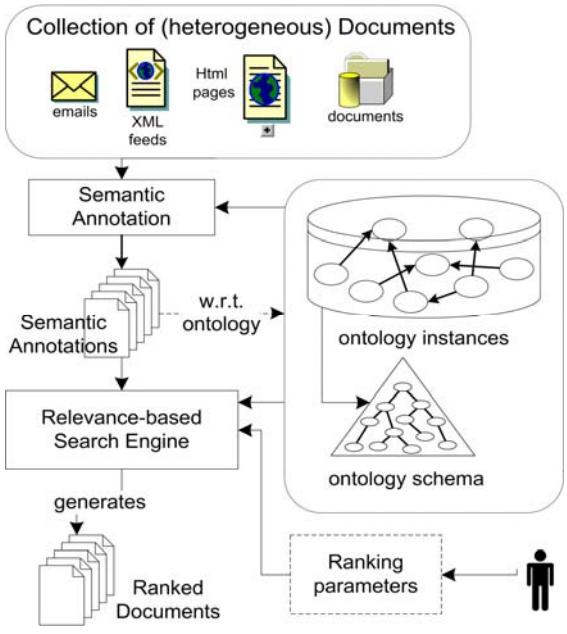

Figure 11: Schematic of the System Architecture

The relevance measure makes use of subjective knowledge by a domain expert. 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.

5.1 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 previous work, we determined relevant documents with respect to a set of classes/concepts [8]. The score of a document was the summation of the weights of paths from entities spotted in a document to the concepts. However, one of the issues was that 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 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. We use an example to illustrate this as follows.

Suppose that two documents mention the city *Pasadena*, but one document also mentions *California* whereas the other document mentions *Arizona*. If the input keyword from user was *pasadena*, then the entity *Pasadena* in the ontology would be the match-entity for the annotations at both documents. Suppose the ontology contains relationships *located-in* connecting cities and states, as well as states and countries. Then, there are sequences of relationship in the ontology connecting *Pasadena* to *California*, and *Pasadena* to *Arizona*. Both documents are related to the query, but arguably document that mentions *California* is more ‘closely’ related to the query because *Pasadena* is a city in the state of *California*. 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. Suppose that the input keyword from user is *Nevada* and both documents contain it. From the perspective that Nevada is the match-entity, the relevance of either document containing *California* or *Arizona* is computed differently than the example where the match-entity was *Pasadena*.

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. Instead of finding neighboring entities with a breadth-first search or similar algorithm, 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 *Pasadena*, it makes sense to consider as 'important' the entity *California*, which it is connected to *Pasadena* by a *located-in* relationship. On the other hand, if the entity *California* is the match-entity, then it might not make as much sense to consider each city located therein 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, a sequence considered important would be "*City* → *located-in* → *State*." An ontology with different vocabulary might need a sequence to be expressed as "*City* ← *has-city* ← *State*." That is, the directionality of the relationship is dependant upon the how data in the ontology is represented but it does not impose any restriction on how the data is used.

Determining sequences to be used by the algorithm can be thought of asking the question: "if the search term is an entity *e* of type *t*, then, according to the ontology-schema, what are the possible types of the entities connected to *e* and what is the relationship(s) in such a path?"

The previous examples illustrated paths of length one. However, paths of longer length can also specify that certain entities are important with respect to the match-entity. For example, the co-authors can be considered important entities for a match-entity of type *Researcher*. The sequence connecting two co-authors might have to go through a *Publication* entity if there is no co-author relationship directly

linking them. The sequence that specifies that co-authors are considered important would be "*Researcher → author-in → Publication ← author-in ← Researcher*," which is a path of length two. In fact, the path would be longer if the data in the ontology is modeled in RDF with "rdf:Seq" to keep ordering of authors – this would make the path length of size four.

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 was to use three levels: low, medium, and high. For example, the sequence "*City → located-in → State*" could be given a medium-importance where as the sequence "*City → located-in → State → Country*" 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 (low, medium, and high). The relevance measure then proceeds as follows.

i. Initialize total score to zero

ii. Each sequence is considered independently, for which:

ii.a. Each possible undirected path starting from the match-entity is evaluated with respect to a sequence to determine a set of neighboring entities that are important with respect to the match-entity.

ii.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"

iii.a. If it is in *lowSet*, then add the corresponding low-score to the total score

iii.b. If it is in *mediumSet*, then add the corresponding medium-score to the total score

iii.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 human assigns the "low/medium/high" scores. In our experience, these facilitate the scoring of a document whereby even small differences in scores has an impact on the ranked results.

5.2 Ranking of Documents Using Relevance Measure

One application of this method can be to re-rank results from a search engine, or to filter out non-relevant results depending on whether certain entities appear or not in a document. Additional query constraints can potentially provide more precise search results. For example, by including not/and operators, referencing classes or relationships from the ontology, explicitly indicating other entities important to the query yet not required to appear within the result set, and the relevance to a pre-defined context. The idea of a user context is to capture more accurately the focus of the search. This idea has been mentioned in the literature [27][60] yet it has not gained much attention by major search engines. Arguably, this is due to the fact that users find it easier to type simpler queries than complex ones.

Annotator for Named-Entities. The annotator for named-entities that we built in UIMA produces the same annotation type for all entities regardless of their class/concept in the ontology. An annotation-query therefore looks like this: <spotted>arnold</spotted>. It is possible that more than one URI (i.e., identifier) gets included into the search index in the cases where a name does match multiple named-entities in the ontology. For example, the appearance of the text "David Jefferson" is a match for different "David Jefferson" entities in the ontology such as "David Jefferson," "David K. Jefferson," and "David R. Jefferson." The entity-disambiguation problem is addressed when the score of a document is computed for ranking (explained later).

Retrieval of Documents using UIMA. We extended the semantic search component in UIMA to include our relationship-based relevance measure and its applicability in ranking documents. 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.

Ranking. The core of our ranking method takes place when the entity-matches from an annotation-query are re-ranked. Initially, the default score by UIMA is set to zero. 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 a 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. Examples where only one entity from the ontology is a match for an annotation include names of organizations, which in most cases are unique and unambiguous. However, it also depends on the level of granularity for which the ontology

contains information. For example, an ontology might have only one *IBM* entity whereas another ontology might have several *IBM* entities such as *IBM Almaden* and *IBM India Research Labs*. An input query *IBM* from user would match all such entities but a user query *IBM Almaden* would match only the annotations of *IBM Almaden*.

5.3 Document Score Adjustments for Ambiguous Entities

More frequently, there are various entities from the ontology with same name as an annotation that does match a user query. Our earlier example of the three different "David Jefferson" entities illustrates this ambiguity problem. In such case, the query from user would have been *David Jefferson*. If the query from user is just *Jefferson*, there would be still three different *David Jefferson* entities that match the annotation. Other entities that match the annotation may or may not have multiple matches. For example, suppose that there is an entity *Michael Jefferson*. This entity is also match the query but it is not ambiguous with respect to the other *David Jefferson* entities.

Let us refer to ambiguous entities as the set *E*. The next step is to be determine an entity *e* in *E* for which the score of document *d* is the maximum. The intuition behind selecting the entity *e* that maximizes the score of a document is based on the notion that it is more likely that *e*'s related entities would appear in the same document. In fact, this rationale has been used for disambiguation of entities based on "evidence" such as our previous work on disambiguating named entities in text [49]. The methods of that approach made use of relations to other entities appearing in text as clues to determine the right entity out of ambiguous entities.

There is another place where ambiguous entities might appear, which is the set A of the other annotations in the document. As mentioned earlier, the annotation step keeps the URI of the entities spotted in a document. Therefore, the set A contains URIs of all ambiguous entities in a document. This means that the relevance measure will eventually consider the right entity. It might seem that the ambiguous entities add noise to the relevance computation yet this was not the case due to the fact that the relevance measure does not penalize the score of a document if it contains entities that are not related to the input query. In fact, it is unclear whether such penalty, if implemented, would add value to the

method, because we foresee that the results have more potential for improvement by enhancing the ontology. The ambiguous entities that we came across more frequently were person names where the various matches are due to lack of middle initial. These observations are based on our experience on manual verification of specific queries and their results. A rigorous evaluation of these aspects is outside the scope of this paper.

5.4 Remarks About Usage of Ontology

It is worth mentioning that there are benefits of using an ontology in this approach. A sequence contains concepts (i.e., classes) from the ontology and therefore the concept of each element in a sequence has to correspond with the concept of the elements in the path. This is applicable not only to entities in the path but also to the *named-relationship* connecting the entities. Hence, traversal of the relationships connected to the match-entity is performed yet it is not exactly a 1-to-1 comparison. For example, in the case of finding co-authors based on the before mentioned sequence, there might be many instances of paths that correspond to such sequence.

The concepts in sequences can be as narrow or broad as needed (depending on the hierarchy of concepts in the ontology-schema). For example, the sequence that specifies that co-authors are considered important includes the *Publication* concept in the sequence. Entities in the ontology that are of type *Publication* would correspond to such concept but also is the case for entities that were defined using sub-concepts (i.e., sub-class-of) such as *Book*, *Journal Article*, and *Thesis*. Similarly, this is also applicable for the named-relationships because in RDF it is possible to define a hierarchy of properties. For example, a relationship *leader-of-organization* could be defined as a specialized type of a relationship *member-of-organization*. Both for concepts and relationships, it is possible to use the more specialized concepts to specify more restrictive sequences used by the relevance measure.

Other methods have used the ontology itself to assign different importance values to entities in the ontology [72][74]. We explored this possibility yet it is possible that newer elements in the ontology could not be assigned a good enough 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.

CHAPTER 6

EXPERIMENTAL EVALUATION

It is rather challenging to devise an evaluation method for techniques of search and ranking of documents. Traditional IR methods rely upon collections of documents revised in advance by humans (e.g., TREC datasets). The usefulness of such collections cannot be denied. However, their applicability is somewhat limited. In the case of entity-based search, it is necessary to know which document does match a given entity. Methods that exploit the value of ontologies would require a collection of documents containing named-entities in the ontology. Moreover, if an input query does match multiple named-entities, then it would be necessary to know which entity pertains to each document. For example, the input query *georgia* is a match for documents containing named entities *University of Georgia* and/or *Georgia Institute of Technology*. In spite of such difficulties, we figured out a combination of ontology and document collection for the evaluation of our ranking method.

6.1 Experiments Setup

We used the SwetoDblp ontology [10], which is based from data from the DBLP bibliography (dblp.uni-trier.de). SwetoDblp incorporates additional data not in DBLP including entities and relationships such as affiliations of authors as well as datasets of organizations and universities. This ontology is a good example of the increasing availability of large populated ontologies. It contains metadata of over 1/2 million authors and nearly 900K publications. There are over 1.5 million relationships among the different entities in SwetoDblp. This ontology carries the benefits of DBLP data yet with the addition of the advantages provided by semantic marked up data. For example, the concept (or 'class') Publication includes sub-concepts of different types of publications such as conference publication, journal article, chapter in book, and conference proceedings. SwetoDblp is available online at http://lsdis.cs.uga.edu/projects/semdis/swetodblp/. The data of DBLP has also been made available in RDF as DR2Q-generated RDF data [22], and Andreas Harth's DBLP dataset in RDF

(http://sw.deri.org/~aharth/2004/07/dblp/). One thing in common and an obvious benefit is that in these efforts (including SwetoDblp) an URI (i.e., identifier) is assigned to authors. It has been noted that DBLP itself does not have unique ID for authors [35]. The RDF-store used for processing ontology data is BRAHMS [53].

Second, the document collection used in the evaluations was chosen directly from the metadata in DBLP publications that links to the electronic edition ('ee') of (most of the) publications. For example, there are 'ee' metadata links to ACM Digital Library or IEEE Digital Library. Hence, we use such links for crawling the content in such sites, which contains many other named entities from the ontology. For example, the ACM Digital Library pages for publications typically include the listed references in the paper. Not all 'ee' hyperlinks had useful content due to broken links. However, it was expected to get good documents from *ee* hyperlinks containing *doi.acm.org* as prefix. The nearly 14K web documents added up, after detagging, to 1/2 GB in disk. The indexes and annotations that UIMA creates add up to 4 GB. The main benefit of using such web pages in the evaluation is that we can verify whether results from a keyword query and its accompanying named entity indeed match to the *known* (i.e., ee links) documents of the entity. For example, a query *lindsay* does match three authors named *Bruce Lindsay* (with different middle initials). However, the DBLP data of each Lindsay correctly points to their respective publications (there are few cases of incorrect names or publication listings in DBLP though). Hence, when our method finds and assigns documents that do match the different *Lindsay* entities, it is possible to verify whether each indeed matched the right document to its corresponding entity.

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 only computed precision measures when the number of results that were known through the ee-link was above 20. In many scenarios of search and Information Retrieval it can be hard to determine a set of relevant documents. Without knowing the relevant documents, it is not possible to compute recall. In general, it cannot be assumed that relevant documents could be known in

advance. However, by using the information of 'ee' hyperlinks, we can know in advance that certain documents have to be relevant results when a query involves any author of the corresponding publications. That is, we used DBLP 'ee' metadata to retrieve the document collection and also use same metadata to create queries and verify that results correspond to the same documents. We believe this is a valid setup for experimentation because the collected documents contain significant additional information than the obvious metadata of the papers. For example, lists of references, abstract, and collaborating authors.

In summary, we used DBLP data represented as an ontology. Second, we crawled documents that are linked from DBLP and performed semantic annotation with the ontology. The known link from publications of authors is then used to verify whether the results of a query do match with retrieved documents.

6.2 Evaluation

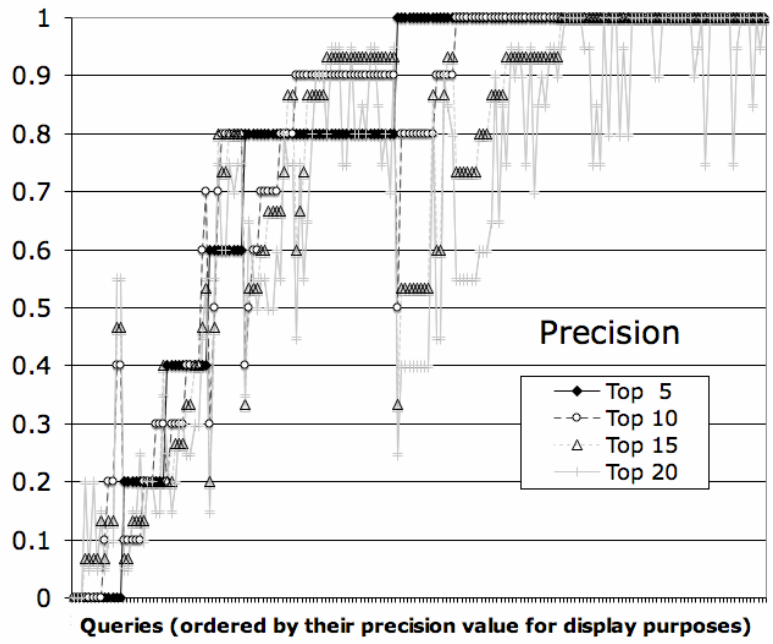

Figure 12: Precision for top 5, 10, 15, and 20 results

Figure 12 illustrates the measure of precision for the top 5, 10, 15 and 20 results for over 150 random queries. In fact, there are 178 data items because for some input queries there are different groups

of ranked-results for the different entity-matches that might exist for the user input. For the purpose of creating a clean and easier to read figure, we sorted the values of precision in ascending order. 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 12 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. Figure 13 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.

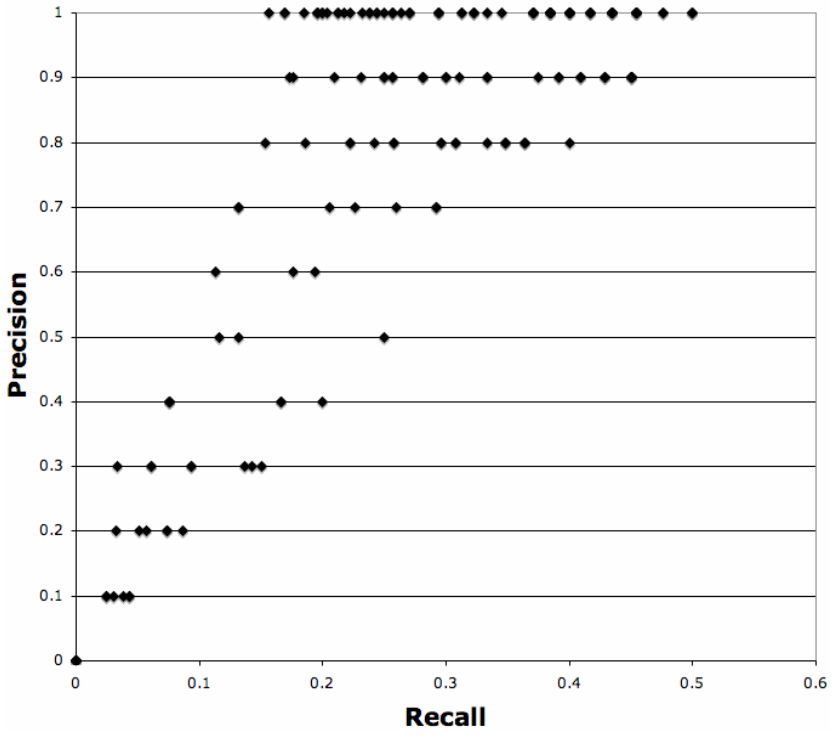

Figure 13: Precision vs. Recall for top 10 results

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 to study is whether or not the use of relationships brings benefits for finding relevant documents. It could be said that the high values of precision are indicative of this. However, a

small experiment clarified this issue. We manually verified the results of a dozen queries when the results are not ranked with out method but instead by comparing with UIMA's default score.

We found out that the ordering of results from UIMA provides good results in terms of grouping the right results to the respective entities matching a given semantic annotation. This is a byproduct of the underlying entity-search capability provided by relying upon the annotations for search. However, it was rare that the top 5 or top 10 results would be the same as those considered relevant (as explained in the experiment setup details). We also tried a handful of queries with the family-names that caused our method to have low precision. In this case, the ordering returned by UIMA was comparable as we would expect that such queries would return a large number of results where ambiguities would be more likely or there would be a large number of entities from the ontology that match the annotations in documents. Hence, we conclude that in fact by using our methods for ranking the results from UIMA there is increased relevance on the top 10 results.

CHAPTER 7

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. The research presented in this thesis addresses the problem of how to exploit semantic relationships of named-entities to improve relevance in search and ranking of documents. The approach taken was to first analyze the relationships of named-entities that match a user-query with respect to the other named entities appearing in documents matching the user query, which is a simple keyword query. The relevance measure takes into consideration sequences of relationships that were assigned relevance weights by a domain expert. In fact, this form of measuring relevance can be seen as addressing the need to take into account *edge semantics* beyond structural semantics [99]. The combination of semantic annotation with the relevance measure leads to determining a score of how relevant a document is to the input query from user. The resulting set of documents is ranked according to this relevance measure.

The use of relationships to rank documents shows promise. The average precision for the top 5, 10 and 15 results was of 77%, 77% and 73% respectively. The top 20 results had a 67% precision. In addition, this approach does not depend on the existence of links or structure in the documents. This can prove advantageous in search scenarios where it cannot be expected that the documents be interlinked. However, there is potential benefit of combining this method with those based on link analysis.

We found that the scoring method is robust for the cases when there are multiple entity-matches for a query. The ambiguous possible matches do not get disambiguated during the semantic annotation step. Instead, the score of a document is calculated for each entity-match and the entity that makes the document to have the highest score is chosen as the entity that the document refers to. Such choice is typically the best among the various ambiguous entities. In fact, this is easy to explain due to the fact that the other entities appearing in a document and their relationships are crucial for scoring the document.

The benefits of providing to the user the results from a query grouped by entity-match are similar to those of clustering, as in Clusty.com search engine (earlier Vivisimo.com). The value is on facilitating the user to focus on the results for the particular entity that s/he might have originally in mind. In fact, I argue that the various groupings of these results can be viewed as a *dimension* that segments the data, as it is the case in proposed frameworks for analysis of large and complex data sets [36].

The Value of Ontologies. In my experience, the value of ontologies as used for this thesis resides on few but key factors. Ontologies should contain a large number of instances, which should be interconnected as their value lies on the context given by the relationships they have with other entities. Additionally, the ontology should be easy to maintain and keep up to date. In our case, building an ontology from DBLP data facilitated the realization of the before mentioned aspects although it required its share of effort to create additional relationships in the ontology such as the *affiliation* and *has-publisher* relationships. For the cases when different persons have the same name, DBLP differentiates the two entries and this information is used to relate such name aliases explicitly, using a 'same-as' relationship between them. The benefit is that the semantic annotation step could find either alias without being concerned of the ambiguities, which are solved by referring to the ontology.

The methods presented are applicable for ontologies containing named entities that are expected to appear in documents. In addition, it is quite important that such entities in the ontologies be interconnected. We foresee that a minimum ratio of entities and their relationships in the ontology is required. Based on our experience, a ratio of two relationships to one entity seems to be the minimum necessary.

Weaknesses of the approach. There are a few weaknesses on the applicability of the methods proposed in this thesis. Although techniques for creating ontologies have improved, the availability of ontologies could be said to be a weak link. An ontology that is far from complete in its domain (i.e., low-quality) could negatively affect semantic annotation and retrieval steps. Measures of ontology quality [17][97][98] can serve as a guide to choose and or improve on a good ontology. Fortunately, there are

community efforts aiming at sharing and re-use of vocabularies for Semantic Web applications [11]. Nevertheless, it is challenging to keep ontologies up to update [50].

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). Challenges remain on spotting the right event in unstructured documents. Their applicability though, could be significant, for example, in search of events in *news*.

Future Work. In respect to future work, indexing of entities and semantic annotations at this time is done using UIMA indexing capabilities. The potential for improvement can be in using a *top-k* evaluation approach to estimate roughly whether a document has potential to be ranked high. However, the effort required to modify UIMA indexing capabilities might be significant. Other lines of future work include exploring the applicability of the methods presented in this thesis for semantic similarity, targeted advertisement, and recommender systems. Additionally, we will explore the applicability of this search using relationships in the domain of national security.

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. The goal would be to exploit and combine the benefits of different approaches.

REFERENCES

[1] L. A. Adamic, O. Buyukkokten, and E. Adar. A Social Network Caught in the Web. *First Monday, 8*(6), 2003

[2] H. Alani, S. Dasmahapatra, K. O'Hara, and N. Shadbolt. Identifying communities of practice through ontology network analysis. *IEEE Intelligent Systems, 18*(2):18-25, 2003

[3] B. Aleman-Meza, C. Halaschek, I.B. Arpinar, and A.P. Sheth. Context-Aware Semantic Association Ranking. In *Proc. First International Workshop on Semantic Web and Databases*, Berlin, Germany, pages 33-50, 2003

[4] B. Aleman-Meza, C. Halaschek, A.P. Sheth, I.B. Arpinar, and G. Sannapareddy. SWETO: Large-Scale Semantic Web Test-bed. In *16th International Conference on Software Engineering and Knowledge Engineering (SEKE2004): Workshop on Ontology in Action*, Banff, Canada, pages 490-493, 2004

[5] B. Aleman-Meza, C. Halaschek-Wiener, I.B. Arpinar, C. Ramakrishnan. and A.P. Sheth. Ranking Complex Relationships on the Semantic Web. *IEEE Internet Computing, 9*(3):37-44, 2005

[6] B. Aleman-Meza, P. Burns, M. Eavenson, D. Palaniswami, and A.P. Sheth. An Ontological Approach to the Document Access Problem of Insider Threat. In *Proc. IEEE International Conference on Intelligence and Security Informatics*, Atlanta, Georgia, pages 486-491, 2005

[7] B. Aleman-Meza, C. Halaschek, and I.B. Arpinar. Collective Knowledge Composition in a Peer-to-Peer Network. In *Encyclopedia of Database Technologies and Applications*, (L.C. Rivero, J.H. Doorn and V.E. Ferraggine, Eds), Idea-Group Inc., 2005

[8] 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

[9] B. Aleman-Meza, M. Nagarajan, C. Ramakrishnan, L. Ding, P. Kolari, A.P. Sheth, I.B. Arpinar, A. Joshi, and T. Finin. Semantic Analytics on Social Networks: Experiences in Addressing the Problem of Conflict of Interest Detection. In *Proc. 15th International World Wide Web Conference*, Edinburgh, Scotland, UK, pages 407-416, 2006

[10] 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*, (Accepted Manuscript) 2007

[11] B. Aleman-Meza, U. Bojars, H. Boley, J.G. Breslin, M. Mochol, L.J.B. Nixon, A. Polleres, and A.V. Zhdanova. Combining RDF Vocabularies for Expert Finding. In *Proc. Fourth European Semantic Web Conference*, Innsbruck, Austria, June 235-250, 2007

[12] R. Anderson, and A. Khattak. The use of information retrieval techniques for intrusion detection. In *Proc. of First International Workshop on the Recent Advances in Intrusion Detection*, 1998

[13] K. Anyanwu, and A.P. Sheth. ρ-Queries: Enabling Querying for Semantic Associations on the Semantic Web. In *Proc. Twelfth International World Wide Web Conference*, Budapest, Hungary, pages 690-699, 2003

[14] K. Anyanwu, A.P. Sheth, and A. Maduko. SemRank: Ranking Complex Relationship Search Results on the Semantic Web. In *Proc. 14th International World Wide Web Conference*, Chiba, Japan, pages 117-127, 2005

[15] K. Anyanwu, A. Maduko, A.P Sheth. SPARQ2L: Towards Support For Subgraph Extraction Queries in RDF Databases, In *Proc. 16th International World Wide Web Conference*, Banff, Alberta, Canada, pages 797-806, 2006

[16] I. Arce. The Weakest Link Revisited. *IEEE Security and Privacy*, pages, 72-76, 2003

[17] I.B. Arpinar, K. Giriloganathan, and B. Aleman-Meza. Ontology Quality by Detection of Conflicts in Metadata. In *Proc. Fourth International EON Workshop: Evaluation of Ontologies for the Web*, Edinburgh, Scotland, 2006

[18] 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

[19] S. Bergamaschi, S. Castano, and M. Vincini. Semantic Integration of Semistructured and Structured Data Sources. *SIGMOD Record, 28*(1):54-59, 1999

[20] T. Berners-Lee, J. Hendler, and O.Lassila. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 2001

[21] A. Bernstein, C. Kiefer, and M. Stocker. OptARQ: A SPARQL Optimization Approach based on Triple Pattern Selectivity Estimation. *Technical Report No. ifi-2007-03*. Department of Informatics, University of Zurich, 2007

[22] C. Bizer. D2R MRP - a Database to RDF Mappling Language. In *Proc. Twelfth International World Wide Web Conference*, Budapest, Hungary, 2003

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

[24] A.Z. Broder, and A.C. Ciccolo. Towards the Next Generation of Enterprise Search Technology. *IBM Systems Journal, 43*(3):451-454, 2004

[25] D. Cameron, B. Aleman-Meza, and I.B. Arpinar. Collecting Expertise of Researchers for Finding Relevant Experts in a Peer-Review Setting, In *Proc. 1st International ExpertFinder Workshop*, Berlin Germany, 2007

[26] 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

[27] J. Coutaz, J.L. Crowley, S. Dobson, and D. Garlan. Context is Key. *Communications of the ACM, 48*(3):49-53, 2005

[28] T. Coffman, S. Greenblatt, and S. Marcus. Graph-based Technologies for Intelligence Analysis. *Communications of the ACM, 47*(3):45-47, 2004

[29] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In *Proc. of the 27th International Conference on Very Large Data Bases*, Rome, Italy, pages 109-118, 2001

[30] L. Deligiannidis, A.P. Sheth, and B. Aleman-Meza. Semantic Analytics Visualization, In *Proc. IEEE International Conference on Intelligence and Security Informatics*, San Diego, CA, USA, pages 48-59, 2006

[31] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R.V. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin, and J.Y. Zien. SemTag and Seeker: Bootstrapping the Semantic Web via automated Semantic Annotation. In *Proc. 12th International World Wide Web Conference*, Budapest, Hungary, pages 178-186, 2003

[32] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran. A Case for Automated Large Scale Semantic Annotation. *Journal of Web Semantics, 1*(1), 2003

[33] L. Ding, P. Kolari, T. Finin, A. Joshi, Y. Peng, and Y. Yesha. On Homeland Security and the Semantic Web: A Provenance and Trust Aware Inference Framework. In *Proc. AAAI Spring Symposium on AI Technologies for Homeland Security*, Stanford University, California, USA, 2005

[34] L. Ding, and T. Finin. Characterizing the Semantic Web on the Web, In *Proc. 5th International Semantic Web Conference*, Athens, Georgia, pages 242-257, 2006

[35] E. Elmacioglu, and D. Lee. On Six Degrees of Separation in DBLP-DB and More. *SIGMOD Record, 34*(2):33-40, 2005

[36] R. Fagin, R.V. Guha, R. Kumar, J. Novak, D. Sivakumar, A. Tomkins. Multi-structural databases. In *Proc. Twenty-fourth Symposium on Principles of Database Systems*, Baltimore, Maryland, USA, pages 184-195, 2005

[37] D. Ferrucci, and A. Lally. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering, 10*(3-4):237-348, 2004

[38] J. Graupmann, R. Schenkel, and G. Weikum. The SphereSearch Engine for Unified Ranked Retrieval of Heterogeneous XML and Web Documents. In *Proc. 31st International Conference on Very Large Data Bases,* Trondheim, Norway, pages 529-540, 2005

[39] T. Gruber. A Translation Approach to Portable Ontologies. In *Knowledge Acquisition*, Chapter 5(2), 1993

[40] R.V. Guha, R. McCool, and E. Miller. Semantic Search. In Proc. *12th International World Wide Web Conference*, Budapest, Hungary, pages 700-709, 2003

[41] R.V. Guha, R. McCool. TAP: A Semantic Web Test-bed, *Journal of Web Semantics, 1*(1):81-87, 2003

[42] 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

[43] C. Halaschek, B. Aleman-Meza, I.B. Arpinar, and A.P. Sheth. Discovering and Ranking Semantic Associations over a Large RDF Metabase. (Demonstration Paper) In *Proc. 30th International Conference on Very Large Data Bases*, Toronto, Canada, 2004

[44] B. Hammond, A.P. Sheth, and K.J. Kochut. Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content. In *Real World Semantic Web Applications* (V. Kashyap and L. Shklar, eds.), Ios Press, pages 29-49, 2002

[45] F. Hakimpour, B. Aleman-Meza, M. Perry, and A.P. Sheth. Data Processing in Space, Time and Semantics Dimensions, *In Proc. International Workshop: Terra Cognita - Geospatial Semantic Web*, Athens, Georgia, USA, 2006

[46] F. Hakimpour, B. Aleman-Meza, M. Perry, and A.P. Sheth. Spatiotemporal-Thematic Data processing in Semantic Web. In *The Geospatial Web* (A. Scharl, and K. Tochtermann, Eds.), Springer-Verlag, 2007

[47] S. Handschuh, S. Staab, and R. Studer. Leveraging Metadata Creation for the Semantic Web with CREAM. In *Proc. 26th Annual German Conference on AI*, Hamburg, Germany, pages19-33, 2003

[48] S. Handschuh, and S. Staab. CREAM CREAting Metadata for the Semantic Web. *Computer Networks, 42*:579-598, Elsevier, 2003

[49] 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

[50] M. Hepp. Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies. *IEEE Internet Computing, 11*(1):90-96, 2007

[51] J. Hollywood, D. Snyder, K.N. McKay, and J.E. Boon. Out of the Ordinary: Finding Hidden Threats by Analyzing Unusual Behavior. *RAND Corporation*, 2004

[52] I. Horrocks, and S. Tessaris. Querying the Semantic Web: A Formal Approach. In *Proc. First International Semantic Web Conference*, Sardinia, Italy, pages 177-191, 2002

[53] M. Janik, and K.J. Kochut. BRAHMS: A WorkBench RDF Store and High Performance Memory System for Semantic Association Discovery. In *Proc. Fourth International Semantic Web Conference*, Galway, Ireland, pages 431-445, 2005

[54] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. RQL: A Declarative Query Language for RDF. In *Proc. Eleventh International World Wide Web Conference*, Honolulu, Hawaii, USA, pages 592-603, 2002

[55] V. Kashyap, and A.P. Sheth. Semantic and Schematic Similarities between Database Objects: A Context-Based Approach. *VLDB Journal*, *5*(4):276-304, 1996

[56] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM, 46*(5):604-632, 1999

[57] K.J. Kochut, M. Janik. SPARQLeR: Extended Sparql for Semantic Association Discovery, In *Proc. Fourth European Semantic Web Conference*, Innsbruck, Austria, pages 145-159, 2007

[58] V. Krebs. Mapping Networks of Terrorist Cells. *Connections, 24*(3):43-52, 2001

[59] 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

[60] S. Lawrence. Context in Web Search. *IEEE Data Engineering Bulletin*, *23*(3):25-32, 2000

[61] T. Laz, K. Fisher, M. Kostich, and M. Atkinson. Connecting the dots. *Modern Drug Discovery*, pages 33-36, 2004

[62] Y.L. Lee. Apps Make Semantic Web a Reality, *SD Times*, 2005

[63] S. Lin, H. Chalupsky. Unsupervised Link Discovery in Multi-relational Data via Rarity Analysis. In *Proc. Third IEEE International Conference on Data Mining*, 2003

[64] H. Liu, P. Maes, G. Davenport. Unraveling the Taste Fabric of Social Networks. *International Journal on Semantic Web and Information Systems*, *2*(1):42-71

[65] A. Maduko, K. Anyanwu, A.P. Sheth, and P. Schliekelman. Estimating the cardinality of RDF graph patterns. In *Proc. 16th International World Wide Web Conference*, Banff, Alberta, Canada, pages 1233-1234, 2006

[66] A. Maedche, S. Staab, N. Stojanovic, R. Studer, and Y. Sure. SEmantic PortAL – The SEAL approach. In *Creating the Semantic Web* (D. Fensel, J. Hendler, H. Lieberman, W. Wahlster, eds.) MIT Press, MA, Cambridge, 2001

[67] B. McBride. Jena: A semantic Web toolkit. *IEEE Internet Computing*, *6*(6):55-59, 2002

[68] R.F. Mihalcea and S.I. Mihalcea. Word Semantics for Information Retrieval: Moving One Step Closer to the Semantic Web. In *Proc. 13th International Conference on Tools with Artificial Intelligence*, Dallas, Texas, pages 280-287, 2001

[69] P. Mika, P. Flink: Semantic Web technology for the extraction and analysis of social networks. *Journal of Web Semantics, 3*(2-3):211-223, 2005

[70] G. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, *38*(11):39-41, 1995

[71] E. Miller. The Semantic Web is Here. In *Keynote at the Semantic Technology Conference*, San Francisco, California, USA, 2005

[72] 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

[73] M.E.J. Newman. Scientific collaboration networks: II. Shortest paths, weighted networks, and centrality, *Phys. Rev. E 64*:016132, 2001

[74] 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

[75] M. Papagelis, D. Plexousakis, and P.N. Nikolaou. CONFIOUS: Managing the Electronic Submission and Reviewing Process of Scientific Conferences. In *Proc. Sixth International Conference on Web Information Systems Engineering*, New York, NY, USA, 2005

[76] M. Perry, M. Janik, C. Ramakrishnan, C. Ibanez, B. Arpinar, A.P Sheth. Peer-to-Peer Discovery of Semantic Associations. In *Proc. Second International Workshop on Peer-to-Peer Knowledge Management*, San Diego, CA, USA, 2005

[77] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. KIM - Semantic Annotation Platform. In *Proc. Second International Semantic Web Conference*, Sanibel Island, Florida, pages 484-499, 2003

[78] C. Ramakrishnan, W.H. Milnor, M. Perry, and A.P. Sheth. Discovering Informative Connection Subgraphs in Multi-relational Graphs. *SIGKDD Explorations, 7*(2):56-63, 2005

[79] C. Ramakrishnan, K.J. Kochut, and A.P. Sheth. A Framework for Schema-Driven Relationship Discovery from Unstructured Text. In *Proc. Fifth International Semantic Web Conference*, Athens, Georgia, pages 583-596, 2006

[80] P. Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language, *Journal of Artificial Intelligence Research*, *11*:95-130, 1999

[81] 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

[82] M. Rodriguez, and M. Egenhofer. Determining Semantic Similarity among Entity Classes from Different Ontologies, *IEEE Transactions on Knowledge and Data Engineering*, *15*(2):442-456, 2003

[83] S.S. Sahoo, C. Thomas, A.P. Sheth, W.S. York, S. Tartir. Knowledge Modeling and its application in Life Sciences: A Tale of two Ontologies, In *Proc. 15th International World Wide Web Conference*, Edinburgh, Scotland, pages 317-326, 2006

[84] A. Sengupta, M. Dalkilic, and J. Costello. Semantic Thumbnails: A Novel Method for Summarizing Document Collections. In *Proc. 22nd Annual International Conference on Design of Communication: The engineering of Quality Documentation*, Memphis, Tennessee, pages 45-51, 2004

[85] N.R. Shadbolt, N. Gibbins, H. Glaser, S. Harris, and m.c. schraefel. Walking through CS AKTive Space: a demonstration of an integrated Semantic Web application. *Journal of Web Semantics, 1*(4):415-419, 2004

[86] N.R. Shadbolt, T. Berners-Lee, and W. Hall. The Semantic Web Revisited. *IEEE Intelligent Systems, 21*(3):96–101, 2006

[87] U. Shah, T. Finin, A. Joshi, R.S. Cost, and J. Mayfield. Information Retrieval on the Semantic Web. In *Proc. 10th International Conference on Information and Knowledge Management*, McLean, Virginia, USA, pages 461-468, 2002

[88] A.P. Sheth, C. Bertram, D. Avant, B. Hammond, K.J. Kochut, and Y. Warke. Managing Semantic Content for the Web. *IEEE Internet Computing, 6*(4):80-87, 2002

[89] A.P. Sheth, and V. Kashyap. So Far (Schematically) yet So Near (Semantically). *IFIP Transactions a-Computer Science and Technology, 25*:283-312, 1993

[90] A.P. Sheth, I.B. Arpinar, and V. Kashyap. Relationships at the Heart of Semantic Web: Modeling, Discovering and Exploiting Complex Semantic Relationships. In *Enhancing the Power of the Internet Studies in Fuzziness and Soft Computing* (M. Nikravesh, B. Azvin, R. Yager, and L.A. Zadeh, eds.), Springer-Verlag, 2003

[91] A.P. Sheth, and C. Ramakrishnan. Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis. *IEEE Data Engineering Bulletin, 26*(4):40-48, 2003

[92] A.P. Sheth. From Semantic Search & Integration to Analytics, In *Proc. Semantic Interoperability and Integration*, IBFI, Schloss Dagstuhl, Germany, 2004

[93] A.P. Sheth, B. Aleman-Meza, I.B. Arpinar, C. Halaschek, C. Ramakrishnan, C. Bertram, Y. Warke, D. Avant, F.S. Arpinar, K. Anyanwu, and K.J. Kochut. Semantic Association Identification and Knowledge Discovery for National Security Applications. *Journal of Database Management, 16*(1):33-53, 2005

[94] A.P. Sheth. Enterprise Applications of Semantic Web: The Sweet Spot of Risk and Compliance. In *Proc. IFIP International Conference on Industrial Applications of Semantic Web*, Jyväskylä, Finland, 2005

[95] H. Snoussi, L. Magnin, and J.-Y. Nie. Toward an Ontology-based Web Data Extraction. In *Proc. Workshop on Business Agents and the Semantic Web*, Calgary, Alberta, Canada, 2002

[96] N. Stojanovic, R. Studer, and L. Stojanovic. An Approach for the Ranking of Query Results in the Semantic Web. In *Proc. 2nd International Semantic Web Conference*, Sanibel Island, Florida, pages 500-516, 2003

[97] S. Tartir, I.B. Arpinar, M. Moore, A.P. Sheth, and B. Aleman-Meza. OntoQA: Metric-Based Ontology Quality Analysis. In *Proc. IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, Houston, TX, USA, 2005

[98] S. Tartir, and I.B. Arpinar. Ontology Evaluation and Ranking using OntoQA. In *Proc. First IEEE International Conference on Semantic Computing*, Irvine, California, USA, 2007

[99] C. Thomas, and A.P. Sheth. On the Expresiveness of the Languages for the Semantic Web - Making a Case for 'A Little More.' In *Fuzzy Logic and the Semantic Web* (E. Sanchez, Ed.) Elsevier, 2006

[100] J. Townley. The Streaming Search Engine That Reads Your Mind. *Streaming Media World*, 2000

[101] K. Tu, M. Xiong, L. Zhang, H. Zhu, J. Zhang, Y. Yu. Towards Imaging Large-Scale Ontologies for Quick Understanding and Analysis. In *Proc. Fourth International Semantic Web Conference*, pages 702-715, 2005

[102] N. Uramoto, H. Matsuzawa, T. Nagano, A. Murakami, H. Takeuchi, and K. Takeda. A Text-mining System for Knowledge Discovery from Biomedical Documents. *IBM Systems Journal, 43*(3):516-533, 2004

[103] M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *Proc. 13th International Conference on Knowledge Engineering and Management*, Sigüenza, Spain, 2002

[104] W. Woods. What's in a link: Foundations for Semantic Networks. In D. Bobrow & A. Collins (Eds.), *Representation and Understanding* (pp. 35-82). New York: Academic Press., 1975

[105] L. Xiao, L. Zhang, G. Huang, B. Shi. Automatic Mapping from XML Documents to Ontologies, In *Proc. 4th International Conference on Computer and Information Technology*, Wuhan, China, 2004

[106] B. Yu, and M.P. Singh. Searching social networks. In *Proc. Second International Joint Conference on Autonomous Agents and Multiagent Systems*, Melbourne, Australia, 2003