A FLEXIBLE APPROACH FOR RANKING COMPLEX RELATIONSHIPS ON THE SEMANTIC WEB

by

CHRISTIAN HALASCHEK-WIENER

(Under the Direction of I. Budak Arpinar and Amit P. Sheth)

ABSTRACT

The focus of contemporary Web information retrieval systems has been to provide efficient support for the querying and retrieval of relevant documents. More recently, information retrieval over semantic metadata extracted from the Web has received an increasing amount of interest in both industry and academia. In particular, discovering complex and meaningful relationships among this metadata is an interesting and challenging research topic. Just as the ranking of documents is a critical component of today's search engines, the ranking of complex relationships will be an important component in tomorrow's Semantic Web analytics engines. Building upon our recent work on specifying and discovering complex relationships in RDF (Resource Description Framework) data, called Semantic Associations, we present a flexible ranking approach which can be used to identify more interesting and relevant relationships on the Semantic Web. Additionally, we demonstrate our ranking scheme's effectiveness through an empirical evaluation over a real-world dataset.

INDEX WORDS: Semantic Web, Semantic Associations, Metadata, Semantic Ranking, Context, Semantic Web Ranking

A FLEXIBLE APPROACH FOR RANKING COMPLEX RELATIONSHIPS ON THE SEMANTIC WEB

By

CHRISTIAN HALASCHEK-WIENER

B.S., The University of Georgia, 2002

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

MASTER OF SCIENCE

ATHENS, GEORGIA

2004

A FLEXIBLE APPROACH FOR RANKING COMPLEX RELATIONSHIPS ON THE SEMANTIC WEB

by

CHRISTIAN HALASCHEK-WIENER

Major Advisor:	I. Budak Arpinar Amit P. Sheth
Committee:	E. Rodney Canfield John A. Miller

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2004

DEDICATION

To my parents for all their support and love…

ACKNOWLEDGEMENTS

I would like to thank my advisors Dr. I. Budak Arpinar and Dr. Amit P. Sheth for their guidance, foresight, and assurance in difficult times. Both Dr. Arpinar and Dr. Sheth have been very generous with their time and wisdom. I would additionally like to thank my parents and sister for their continuous encouragement and love. I would also like to thank my girlfriend, Melissa, for her patience and support over the years. I would like to thank Dr. E. Rodney Canfield for his valuable suggestions and being a part of my committee. Additionally, I would like to thank Dr. John A. Miller and Dr. Krzysztof J. Kochut for their valuable suggestions and guidance as well. Lastly, I would like to thank the entire SemDIS project team, especially Boanerges Aleman-Meza and Cartic Ramakrishnan for all they have done.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

The focus of contemporary Web information retrieval systems has been to provide efficient support for the querying and retrieval of documents. There has been significant academic and industrial research in mainstream search engines, such as Google, Vivisimo, Teoma, etc. These systems have made considerable progress in the ability to locate relevant pieces of data among the vast numbers of documents on the Web.

Currently, due to the increasing move from data to knowledge and the rising popularity of the Semantic Web vision [Berners-Lee01], there is significant interest and ongoing research in automatically extracting and representing semantic metadata as annotations to both documents and services on the Web [Shah02, Hammond02, Dill03]. Several communities such as the Gene Ontology Consortium, Federal Aviation Administration (Aviation Ontology), Molecular Biology Ontology Working Group, Stanford University's Knowledge Systems Lab (Enterprise Ontology), etc., are also coming together to effectively conceptualize domain knowledge and enable standards for exchanging, managing and integrating data more efficiently. Additionally, research in the Semantic Web has spawned several commercially viable products through companies such as Semagix, Ontoprise, and Network Inference to name a few.

Due to this ongoing work, large scale repositories of semantic metadata extracted from Web pages have been created and are publicly available. For example, TAP KB (knowledge-base) is a fairly broad but not very deep knowledge-base annotated in RDF (Resource

Description Framework) that contains information pertaining to authors, sports, companies, etc. [Guha03]. Additionally, SWETO (Semantic Web Technology Evaluation Ontology) is a comparatively narrower but deep knowledge-base annotated in either OWL (Web Ontology Language) [Bechhofer03] or RDF [Lassila99] that has been populated with over 800,000 entities and 1.5 million explicit relationships between them [Aleman-Meza04]. Additionally, scalable capabilities for semantic metadata extraction and annotations are demonstrated by the WebFountain project, which has annotated and disambiguated data from well over a billion documents [Quint03], and by Semagix Freedom [Hammond02], which uses SWETO and other domain ontologies to semantically annotate millions of documents or Web pages.

Given these developments, the stage is now set for the next generation of technologies, which will facilitate getting actionable knowledge and information from semantic metadata extracted from Web documents, the deep Web and large enterprise repositories. Traditionally, many users analyze information by either browsing the Web, or using search engines, most of which are systems that only locate documents based on keywords or key phrases. Often, these approaches do not directly satisfy the information needs of the end user. This is because many retrieved documents are either irrelevant or more importantly, contain the actionable information buried deep within the document. Through our earlier work [Anyanwu03], we aim to provide a different type of analysis based on semantic relationships, in which users can discover previously unknown and potentially interesting complex relations between entities, through a set of relationships between the meta-data/annotations of Web sources/documents. We have defined these complex relationships between two entities as Semantic Associations [Anyanwu03]. Arguably, these relationships are at the heart of semantics (e.g., [Sheth03a]), lending meaning to

information, making it understandable and actionable and providing new and possibly unexpected insights.

When querying for Semantic Associations, users are frequently overwhelmed with too many results. For example, a typical Semantic Association query involving two ‘Computer Science Researchers’ over the SWETO test-bed, results in hundreds to thousands of associations. Their associations vary from co-authorship relations through their publications, to relationships through the geographic locations they live in. As with traditional search engine queries where thousands of documents are returned, a user cannot be expected to sift through these large numbers of resulting associations in search of those that are highly relevant to his/her interest. Thus, the Semantic Associations need to be filtered according to their perceived importance and automatically ranked based on their relevance.

While investigating Semantic Association query results, we have found that ranking them is inherently different from ranking documents. This is due to the fact that a Semantic Association is a sequence of complex relationships between entities in the metadata extracted from heterogeneous documents, as opposed to a single document. This is illustrated below in Figure 1, which shows entities and relationships that originate from disparate sources (note that in Figure 1, there are two associations between the red/bold entities). In fact, we have found that for each association, there is no one way to measure its relevance. Thus, we see the need for a flexible, query dependant approach to automatically analyze and relevantly rank the resulting associations, potentially based on factors such as the association length, frequency of occurrence of certain associations, domain from which the associations can be classified (e.g., the geographic domain), etc.

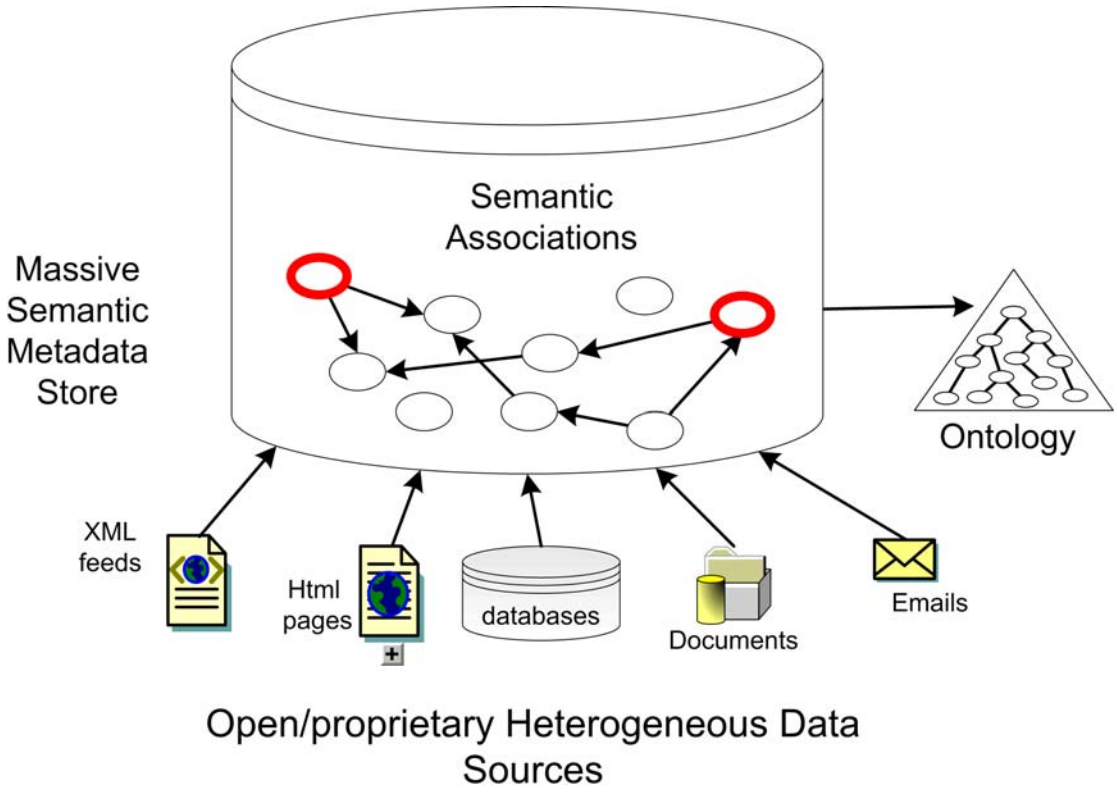

Figure 1.1 Semantic Associations Illustration

In this work, we aim to investigate the challenging component of ranking complex relationships on the Semantic Web. Specifically, we propose a flexible ranking approach that supports automatically filtering irrelevant Semantic Associations and allows the identification of the most interesting relationships. To our knowledge, this is one of the earliest attempts to relevantly rank these types of relationships in semantic metadata. Additionally, we provide details of the current system implementation and perform an evaluation of the approach over the SWETO test-bed. Through this, we demonstrate the effectiveness of the ranking approach by means of the obtained results, thus highlighting the contributions of this work in ranking complex relationships on the Semantic Web.

The next chapter presents some background information. In Chapter 3, we discuss the related work, and Chapter 4 presents the ranking approach. The implementation and programming interface are described in Chapter 5. Then in Chapter 6, performance results are presented and discussed. Lastly, we close this work with a conclusion and discussion of potential future work.

CHAPTER 2

BACKGROUND

This work has been motivated by our previous research in defining and discovering Semantic Associations [Anyanwu03], and is aligned with the current Semantic Web vision [Bernere-Lee01]. In the Semantic Web vision, ontologies, which are essentially conceptualizations of the real-world (i.e., class hierarchies with relationships between them), are used to semantically annotate the current information on the web. Currently, there are a variety of semantic representation languages, including RDFS (RDF Schema) [Brickley00], OWL, etc. Given these ontologies and semantic annotations of data (known as semantic metadata) with respect to them, machines will thus be able to efficiently and in a more automated manner, interpret the data on the Web. Hence, machines, or agents, will be able understand and act upon information regarding both the entities and relationships contained on the Web. In the remaining portions of this chapter, an overview of some of the basic technologies and current advances in achieving this vision will be outlined, in additional to a general discussion of Semantic Associations.

2.1 Metadata Extraction Techniques

Metadata extraction, which is the process of extracting additional information from Web resource with respect to an ontology, has been an active research area over the past years. Both semi-automatic [Handschuh02] and automatic [Hammond02] techniques and tools have been developed and significant work continues [Vargas-Vera02]. Various tools exist, including Cream

[Handshuh03], S-Cream [Handschuh02], Semagix Freedom toolkit [Hammond02], and SemTag [Dill03], etc. Semagix Freedom has typically been used to populate ontologies that average more than one million instances [Sheth03b] and can process over a million documents per day per server, while SemTag, which is part of IBM's WebFountain project, has used a smaller ontology but has demonstrated Web scale metadata extraction from well over a billion pages. Below, we will describe the Freedom toolkit as it has been used as the infrastructure technology to create the data set, SWETO, for our evaluations [Aleman-Meza04].

2.1.1 Semagix Freedom

Semagix Freedom is a commercial product which evolved from the LSDIS Lab's past research in semantic interoperability and the SCORE technology [Sheth02]. The Freedom toolkit provides a variety of functionality including, but not limited to, ontology creation, ontology population through the use of extractors, and entity disambiguation.

Freedom provides an interface for the creation of an ontology, in which a user can define classes and the relationships that it is involved in by using a graphical environment. Thus, the user is relieved of the burden of serializing the ontology to the RDF or OWL syntax. Additionally, extractors can be 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 'crawled' and analyzed (e.g., for name spotting [Hammond02]) by the Freedom extractors, the extracted entities 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. Freedom also provides an API for exporting

both the ontology and its instances in either RDF or OWL syntax. 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 raises issues in the area of entity disambiguation. This is the process of differentiating two entities (e.g., is 'John Smith' the same person as 'John P. Smith'). Using Freedom, entity instances can be disambiguated using syntactic matches and similarities (aliases), customizable ranking rules, and relationship similarities among entities. Freedom is thus able to automatically disambiguate entities as they are extracted [Sheth02]. Furthermore, if Freedom detects ambiguity among new entities and those within the knowledge-base, yet it is unable to disambiguate them within a preset degree of certainty, the entities are flagged for manual disambiguation with some system help on possible matches. Additional details regarding Semagix Inc. and their Freedom toolkit are available at their website (http://www.semagix.com), as well as in [Sheth04b, Sheth02, Hammond02].

2.2 RDF Query Languages

Our lab's research in Semantic Associations is centered on the analysis of semantic metadata. In addition to these types of analytic operations, there currently exists a variety of query languages which facilitate interaction with RDF data. Below, a limited number, yet more popular, of the languages will be briefly discussed.

2.2.1 RDQL

RDQL (RDF Data Query Language) has been developed by HP labs (http://www.hpl.hp.com/) and has potential to become the RDF query language standard. In fact it is the only RDF query

language, as of yet, that has been submitted to the WWW Consortium [Seaborne04]. RDQL is an evolution from numerous query languages, and has been implemented in a variety of RDF storage systems, including Jena [McBride01], Sesame [Broekstra02], etc. [Seaborne04].

An RDQL query is composed of a SELECT, which specifies return variables, WHERE, that refers to the RDF model being accessed, FROM, that provides a set of patterns that have to be matched by the RDF data in the model, and USING, where abbreviations for XML namespaces can be defined to simplify the query. Essentially, an RDQL query consists of a graph pattern, expressed as a list of triple patterns. Each triple pattern is comprised of named variables and RDF values, both URIs and literals (e.g., strings). An RDQL query can additionally have a set of constraints on the values of those variables, as well as a list of the variables required in the answer set [Seaborne04]. Lastly, it should be noted that RDQL is not directly aware of the schema, as it is not directly integrated in the language but it can be provided by the underlying data source.

2.2.2 RQL

RQL (RDF Query Language), in addition to RDQL, is one of the more widely known and used RDF query language. RQL was developed by ICS-FORTH research labs (http://www.ics.forth.gr/) located in Greece. RQL relies on a formal model for directed labeled graphs permitting the interpretation of superimposed resource descriptions by means of one or more RDF schemas [Karvounarakis02]. In contrast to RDQL, RQL provides a rich set of operators for specifying the query result, including explicit operators for navigating the schema [Karvounarakis02]. It is worth noting that this capability is not possible directly with RDQL.

RQL additionally, allows the use of path expressions for navigating the RDF graph, providing further performance benefits.

2.3 RDF Databases and Storage Systems

Currently there are numerous RDF storage systems. Below a very small subset of them will be briefly discussed.

2.3.1 Jena

Jena is a java-based RDF/OWL API for parsing, storing and accessing both RDF and OWL data, which is developed by the HP research labs in Bristol [McBride01]. The Jena API provides both statement centric and resource centric methods for manipulating an RDF/OWL model. Additionally, the API provides built in support for RDF containers (bag, alt and seq) and typed literals. Jena also provides integrated parsers (ARP) and writers for RDF in various formats. This easily allows the importing and exporting of serialized RDF/OWL [McBride01, Jena].

Jena provides a persistence subsystem that provides persistence for models through the use of a back-end database. The default Jena database layout uses a de-normalized schema in which literals and resource URIs are stored directly in statement tables [Jena]. Additionally, the persistence subsystem provides support for RDQL, which is dynamically transformed into SQL queries. Jena is currently compatible with MySQL, Oracle and PostgreSQL.

Jena also provides a reasoner subsystem that includes a rule based inference engine together with configured rule sets for RDFS and basically the OWL-Lite subset of OWL Full. The reasoner subsystem is extensible in that it is possible to use a variety of external reasoners in Jena. Additionally, it provides an ontology API, which is designed to be used by programmers

who are working with ontology data based on RDF. Currently, OWL, DAML+OIL and RDFS are supported.

2.3.2 RDFSuite

RDFSuite is a set of highly scalable tools used for managing volumes of RDF description bases and schemas [Alexaki01], which was developed by ICS-FORTH. Currently, RDFSuite includes a Validating RDF Parser (VRP), a RDF Schema Specific DataBase (RSSDB), and support for RQL. VRP provides support for analyzing, validating and processing RDF schemas and resource descriptions [RDFSuite]. The Parser syntactically analyzes the statements of a given RDF file according to the RDF specification. The Validator checks whether the statements contained in both RDF schemas and resource descriptions satisfy the semantic constraints derived by the RDF Schema Specification (RDFS) [RDFSuite].

Additionally, RDFSuite includes RSSDB, which is a persistent RDF data store for loading resource descriptions in an object-relational DBMS by exploiting the available RDF schema knowledge. The main goal of RSSDB schema-specific representation is the separation of the RDF schema from data information, as well, as the distinction between unary and binary relations holding the instances of classes and properties [RDFSuite]. RSSDB is comprised of a Loading and an Update module, both implemented in Java using a number of primitive methods (APIs) for inserting, deleting, and modifying RDF triples [RDFSuite]. Lastly, RDFSuite supports the RQL query language.

2.4 Semantic Associations

When we consider data on the Web, different entities can be related in multiple ways that cannot be pre-defined. In the Semantic Web vision, the RDF data model [Lassila99] captures the meaning of an entity (or resource) by specifying how it relates to other entities (or classes of resources). Each of these relationships between entities is what we call a "Semantic Association" [Anyanwu03]. In general, most useful Semantic Associations involve some intermediate entities and relations (properties). Relationships that span several entities may be very important in domains such as drug discovery or national security [Sheth04a]; for example, in the latter, this may enable analysts to see the connections between different people, places and events.

[Anyanwu03] presents a formalization of Semantic Associations over metadata represented in RDF. Below we provide definitions regarding the formalization of Semantic Associations (adapted from [Anyanwu03]).

Definition 1 (Semantic Connectivity): Two entities e_1 and e_n are *semantically connected* if there exists a sequence e_1, P_1, e_2, P_2, e_3, …, e_{n-1}, P_{n-1}, e_n in an RDF graph where e_i, $1 \leq i \leq n$, are entities, P_j, $1 \leq j < n$, are properties, , and entities e_i and e_{i+1} are in relationship P_i. A sequence of entities and properties represents a *semantic path*.

Definition 2 (Semantic Similarity): Two entities e_1 and f_1 are *semantically similar* if there exist two *semantic paths* e_1, P_1, e_2, P_2, e_3, … e_{n-1}, P_{n-1}, e_n and f_1, Q_1, f_2, Q_2, f_3, …, f_{n-1}, Q_{n-1}, f_n *semantically connecting* e_1 with e_n and f_1 with f_n, respectively, and that for every pair of properties P_i and Q_i, $1 \leq i < n$, either of the following conditions holds: $P_i = Q_i$ or $P_i \subseteq Q_i$ or Q_i

$\subseteq P_i$ ($\subseteq$ means rdf:subPropertyOf, which is essentially property/relationship inheritance). We say that the two paths originating at e_1 and f_1, respectively, are *semantically similar*.

Definition 3 (Semantic Association): Two entities e_x and e_y are *Semantically Associated* if e_x and e_y are either *semantically connected*, or *semantically similar*.

Given these operators, users can uncover new insights regarding the manner in which entities on the Web are both inter-connected and similar. This type of analytics is currently not supported by current RDF query languages, such as RQL or RDQL.

It should be noted that we are currently working on ranking techniques for *semantic similarity* associations, but these are not discussed in this work. For simplicity, in the remaining sections of this document we will refer to *semantically connected* entities as *Semantic Associations* (or simply associations) and leave the presentation of other types of associations to further work. Note that *entity* and *instance* are used interchangeably throughout this document. Similarly, *property* and *relation* are used interchangeably as well.

CHAPTER 3

RELATED WORK

While the issue of ranking semantic relationships is fundamentally different from the ranking of documents in search results as those addressed in contemporary information retrieval approaches, it is worth discussing a limited number of these techniques. [Brin98] presents the PageRank algorithm used by Google (http://www.google.com/). PageRank weights are assigned on the basis of page references, thus more popular pages have a higher rank. Teoma [Teoma] employs the technique of subject specific popularity, in which a page's rank is based on the number of same-subject pages that reference it, not just its general popularity. In general, contemporary ranking approaches also focus on finding relevance with respect to keywords and primarily rely on statistical information retrieval, social networking and lexical techniques. While relevant, these ranking algorithms lack the consideration of formal semantics and explicitly specified context when assigning ranks.

[Lin03] presents the notion of using rarity as a relevance measure in the context of data mining relational databases. Essentially, [Lin03] considers infrequently occurring relationships (i.e., rare events) to be more interesting when compared to those that are more commonly occurring. This idea is adapted for context of this work.

Research in the actual area of Semantic Web ranking techniques is rather limited, but includes [Maedche01], where the notion of "semantic ranking" is presented to rank queries returned within 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 relevance of results depends on the criteria defined by a user (i.e., the query context, path length, etc.). In [Stojanovic03], the authors expand on the techniques presented in [Maedche01] and assess the relevance of a query result as a function of the specificity of the instantiation of the ontology and the inference process in which the answers were implied. The task of ranking complex semantic relationships, as discussed in this paper, is inherently different than the ranking task addressed in [Maedche01, Stojanovic03]. Additionally, the approach presented here is designed to be very flexible due to the various ways in which these associations can be interpreted (depending on the user's interests).

In earlier work [Anyanwu03], introduces using "context" as a basis for ranking semantic relationships. In [Anyanwu03], a notion of context is defined to include a set of ontologies and a set of relationship name pairs with a value. The value indicates the precedence level, a degree of importance for a particular context. This approach considers context based on value assignments for different ontologies. Later in [Aleman-Meza03, Halaschek04b], we expand on [Anyanwu03], in that the context specification is now defined at a level (of classes and properties) that allows precise definitions of areas of interest for the user. This approach is adopted for the purpose of this work. Other attempts to model context include [Guha91], in which the author uses context to address some of the problems with the traditional model of AI (artificial intelligence). In [Guha91], the author provides both a model and proof theory for contexts, as applicable to AI, as well as demonstrates some sample applications. Additionally, [Kashyap96] proposes a context representation mechanism to solve conflicts of semantic and schematic similarities between

database objects. Lastly, [Crowley02] introduces an ontology that captures users' context and situations by considering goals, tasks, actions and system's context in order to observe and model human activities. The approach is mainly focused to use context to reduce user intervention in the system.

CHAPTER 4

RANKING APPROACH

As discussed earlier, Semantic Associations are a series of complex relationship connecting two entities, which can span multiple domains, Web documents and involve any number of entities and properties. In [Aleman-Meza03], we describe a preliminary approach that defines an association rank as a function of various intermediate criteria. As an association is traversed, many different intermediate weights are assigned to it, which ultimately contribute to the overall association rank. Our work in [Halaschek04b] expands on [Aleman-Meza03] by reassessing the previously described ranking criteria, introducing new criteria and presenting an empirical evaluation. This work will further detail the approach outlined in [Halaschek04b]. In general, we classify the ranking criteria into two categories, *Semantic* and *Statistical* metrics, both of which are detailed below. Note that in the following sections, we generically refer to the entities and properties in an association as the *components* of the association.

4.1 Semantic Metrics

In our ranking approach, we categorize a set of criteria as *Semantic* metrics which are based on semantic aspects of the ontology, instances and associations themselves. The remaining portion of this Chapter presents an overview of and formally defines the *Semantic* metrics that contribute to the overall Semantic Association rank.

4.1.1 Context

Consider a scenario in which a user is interested in the way two '*Persons'* are related to each other in the domain of '*Computer Science Publications*'. Taken from the SWETO ontology, associations that include entities of type (as defined in the schema) '*Scientific Publication*', '*Computer Science Professor*', etc. would be most relevant, where entities of type '*Financial Organization*' would not. Thus, to capture the relevance of a (complex) relationship, we have defined the notion of a *query context* [Aleman-Meza03]. This *query context* is made up of various ontological *regions* (or sub-graphs) specified by the user. Since the types of the entities are described using RDF Schema (RDFS), we can use the associated class and relationship types to restrict our attention to the entities and relations of interest. Hence, by defining *regions* (or sub-graphs) of the RDF Schema, the user can define his/her domain(s) of interest. Lastly, because a user can be interested in a variety of different *regions* with differing degrees of interest, a weight is associated with each *region* specified, where the sum of all context *region* weights adds up to 1.0. Thus, using the *context* specified, it is possible to rank an association according to its relevance with a user's domain of interest [Aleman-Meza03, Halaschek04b].

To illustrate this approach, consider three sample associations between two entities in the SWETO test-bed, as depicted in Figure 4.1, where a user has specified a contextual *region 1* containing classes '*Scientific Publication*' and '*Computer Science Researcher*'. Additionally, assume the user has defined *region 2*, containing classes '*Country*' and '*State*'. The resulting *regions, 1* and *2*, refer to the computer science research and geography domains, respectively.

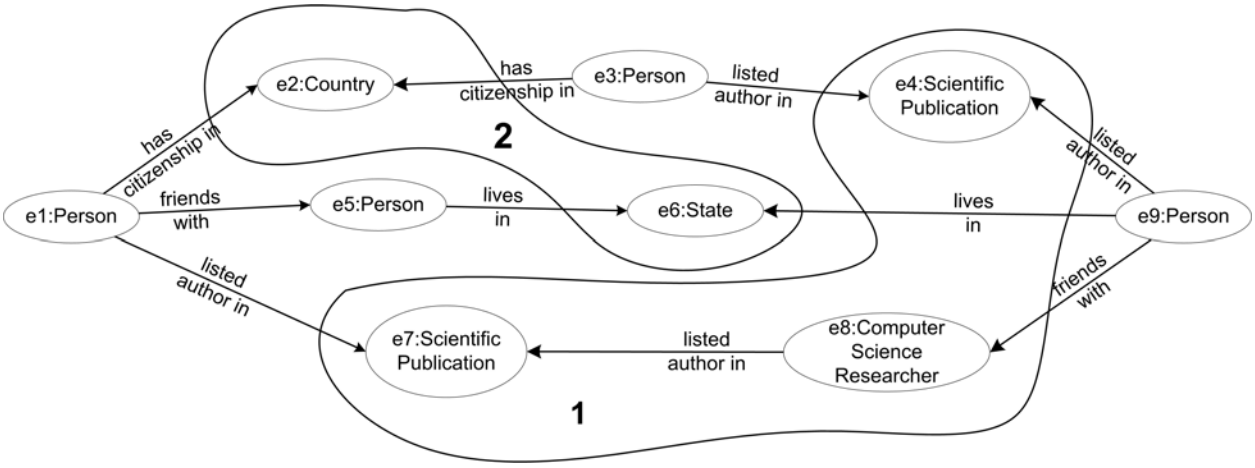

Figure 4.1 Contextually Relevant Semantic Associations (note that the *rdf:type* of the entities is displayed)

A weight assignment of 0.8 for *region 1* and 0.2 for *region 2* illustrates that the user is more interested in the computer science research domain but also wants to consider geography domain related associations, albeit with lesser priority. Then, for the associations in Figure 4.1, the bottom-most association would have the highest rank because all of its components (entities and properties) are contained within the region with highest weight. The secondly ranked association would be the association at the top of the figure because it has a component in *region 1*, but (unlike the association in the middle) also has a component in *region 2*. We will now define the *Context* weight as it is used in the ranking approach.

Let *A* represent a Semantic Association, that is, a path consisting of components, nodes (entities) and edges (properties), that connects the two entities. Let R_i represent the *region i*, that is, the set of classes and properties that capture a domain of interest. Additionally, let *c* be a component of *A*, either a node or an edge. Note that for purposes of all the ranking formulas, we consider each component to be unique within an association (regardless of its URI). We define

the following sets for convenience, using the notation $c \in R_i$ for determining whether the type of *c* (rdf:type) belongs to *region* R_i:

$$X_i = \{c \mid c \in R_i \wedge c \in A\} \ ,$$

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

where *n* is the number of *regions A* passes through. 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

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

where *n* is the number of *regions A* passes through and *length(A)* is the number of components in the association. That is, for each *region* that *A* passes through, sum the total number of components in *A* that are in the *region* R_i and multiply it by the weight attributed to that *region*. This assesses the context regions that this association passes through. To favor associations in which all components are included in some *region*, we then penalize the *Context* weight by the ratio of the total number of components not in any *region*, represented by |*Z*|, and the total number of components. Lastly, the *Context* weight is normalized by the total number of components in the association to account to varying length associations. Note that a property component is considered to be in some *region* if it is entirely included in that *region* or if one of the entities it is involved with (either its subject or object) is in that *region*. If the two entities in which some property is involved are contained in two separate *regions*, then the higher of the two *region* weights is assigned to the property.

4.1.2 Subsumption

When considering classes in an ontology, those that are lower in the hierarchy can be considered to be more specialized instances of those further up in the hierarchy [Rodriguez03]. Thus, lower classes convey more detailed information and have more specific meaning. Figure 4.2 depicts a class, "*Person*", as well as various subclasses of it, where it is apparent that as the hierarchy is traversed from the top down, subclasses become more specialized than their super-classes. Additionally, Figure 4.2 shows the *component subsumption weights* (as defined below) of each class in the hierarchy. The intuition is to assign higher relevance to associations which convey more meaning, based on *Subsumption*.

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

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

where H_{c_i} is the position of component c_i in hierarchy H (the topmost class has a value of 1, the next class has a value of 2, etc.) and H_{height} is the total height of the class/property hierarchy of the current branch.

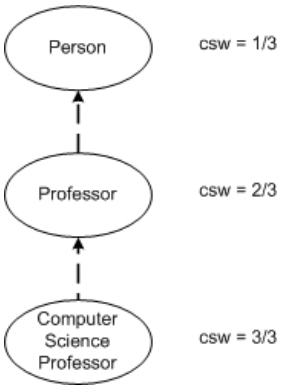

Figure 4.2 Class Hierarchy Example

We now define the overall *Subsumption* weight of an association *A* such that

$$S_A = \frac{1}{length(A)} \times \sum_{i=1}^{length(A)} csw_i \ ,$$

again where *length(A)* is the number of components in *A* and csw_i is the *component subsumption weight* of the i^{th} component in *A*. We do note here that the quality and completeness of the ontology become important to assure the effectiveness of the *Subsumption* weight.

4.1.3 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 of the other news organizations). Thus, trust values need to be assigned to the meta-data extracted depending on its source. In the context of SWETO, trust values are assigned to sources by the extractor writers (assumed to be domain experts). Thus, the trust values of the data sources are stored with all extracted entities and relationships. When extracted data is exported from Freedom to RDF syntax, the trust values are maintained through the use of rdf:Literals. Future work will allow users the option of assigning trust values to the extraction sources, yet this is out of the scope of this thesis.

When assigning a *Trust* weight to an association, we adopt the following intuition: the strength of an association is only as strong as its weakest link. Thus, the *Trust* weight of an association is the value of its least trustworthy component. Hence, we can now define the *Trust* weight of an association.

First, let t_{c_i} represent the *component trust weight* of the component, c_i, in an association, *A*. We now define the *Trust* weight of an overall association *A* as

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

4.2 Statistical Metrics

Additionally, we categorize a variety of ranking criteria as *Statistical* metrics. These criteria are categorized as such because they are based on statistical aspects of the ontology, particularly on number and connectivity aspects of the instances in the knowledge-base, as well as the associations themselves. The remaining portion of this chapter identifies and defines the *Statistical* metrics that contribute to an association's overall rank.

4.2.1 Rarity

Given the size of current Semantic Web test-beds (e.g., SWETO, TAP KB), many relationships and entities of the same type (rdf:type) will 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 [Lin03], which discusses the notion of rarity in the context of data mining of relational databases. [Lin03] considers infrequently occurring relationships (i.e., rare events) to be more interesting when compared to those that are more commonly occurring.

In some queries however, the opposite may be true. For example, in the context of money laundering, often individuals engage in normal looking, common case transactions as to avoid detection [Semagix03]. In this case, most of the relationships and entities in an association may be frequently occurring (common). Thus the user should determine, depending upon the query, which *Rarity* weight preference s/he has, if any.

In our approach, we will 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 (instances and relationships only). Note that we consider all relationships in the knowledge-base to be unique. 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 instances and relationships in K), and

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

with the restriction that in the case res_j and c_i are both of type rdf:Property, the subject and object of c_i and res_j must be of the same rdf:type. Thus rar_i captures the frequency of occurrence of the rdf:type of component c_i, with respect to the entire knowledge-base. We can now define the overall *Rarity* weight, *R*, of an association, *A*, as a function of all the components in *A*, such that

$$\textbf{(a)}\ R_A = \frac{1}{length(A)} \times \sum_{i=1}^{length(A)} rar_i \text{ , or}$$

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

where *length(A)* is the number of components in *A* and rar_i is *component rarity* of the i^{th} component in *A*. If a user wishes 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*.

4.2.2 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 Web page

ranking algorithm [Kleinberg99], as well as the PageRank algorithm [Brin98], our approach takes into consideration the number incoming and outgoing relationships of entities. In our approach, we view the number of incoming and outgoing edges of an entity as its *Popularity*. In some queries, associations with entities that have a high *Popularity* may be more relevant. These entities can be thought of as *hotspots* in the knowledge-base. For example, in the SWETO test-bed, highly cited authors would have a high popularity. In certain queries, associations that pass through these *hotspots* could be considered very relevant. This could be the case if a user were interested in the way two authors were related through co-authorship relations. In this situation, associations which pass through highly cited authors may be more relevant. Yet, in other queries, one may want to rank very popular entities lower. For example, in SWETO, entities of type '*Country*' have an extremely high number of incoming and outgoing relationships, yet convey little information when querying for the way to persons are associated through geographic locations.

Similar to our assessment of association *Rarity*, we define the *Popularity* rank of an association in terms of the *Popularity* of the entities contained within the association itself. We now define 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 knowledge-base, pop_{e_i} 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 knowledge-base 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 rdf:type in the knowledge-base. We know define the overall *Popularity* weight, P, of an association A, such that

$$\textbf{(a)}\ P_A = \frac{1}{n} \times \sum_{i=1}^{n} p_i \ \text{ or}$$

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

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 **(b)** is used. Thus, P_A is essentially the average *Popularity* of all entities in A.

4.2.3 Association Length

In some queries, a user may be interested in more direct associations (i.e., shorter associations). This may imply a stronger relationship between two entities. Yet in other cases a user may wish to find possibly hidden, indirect or discrete associations (i.e., longer associations). The latter may be more significant in domains 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 [Krebs02] or money laundering [Semagix03] involves deliberate innocuous looking transactions that may change several hands. Hence, the user should determine which *Association Length* influence, if any, should be used.

We now define the *Association Length* weight, L, of an association A. If a user wants to favor shorter associations, **(a)** is used, again where *length(A)* is the number of components in the A. In contrast, if a user wants to favor longer associations **(b)** is used.

$$\textbf{(a)}\ L_A = \frac{1}{length(A)} \ \text{ or}$$

$$\textbf{(b)}\ L_A = 1 - \frac{1}{length(A)}$$

4.3 Overall Ranking Criterion

In the above sections, we have defined various association ranking criteria. We will now define the overall association *Rank*, W_A, using these criteria 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 add up to 1.0. This is intended to allow fine-tuning of the ranking criteria (e.g., *Popularity* can be given more weight than *Association Length*). Additionally, this provides the functionality to completely omit certain criteria if that is what the user desires. This provides a flexible, query dependant ranking approach that assesses the overall relevance of Semantic Associations.

CHAPTER 5

SYSTEM IMPLEMENTATION

The ranking approach presented in this work has been implemented and tested within the LSDIS lab's SemDIS[1] and SAI[2] projects. The main components of the SemDIS system architecture are illustrated in Figure 5.1. The entire system, except for the Knowledge Extraction Module (Semagix Freedom), is Web-accessible, and all code was written in Java [Halaschek04a]. Below, details regarding the various system components and their implementation are provided. Because the project has just begun, some sections of the architecture will be omitted, as they have not been addressed yet. It is important to note that the SemDIS prototype implementation has been the result of the work of the entire SemDIS project team. The focus and primary contribution of this work pertain to the portions directly related to ranking and the user interface. However, the other portions of the system architecture are discussed for completeness.

[1] NSF-ITR-IDM Award #0325464, titled 'SemDIS: Discovering Complex Relationships in the Semantic Web.'
[2] NSF-ITR-IDM Award #0219649, titled 'Semantic Association Identification and Knowledge Discovery for National Security Applications.'

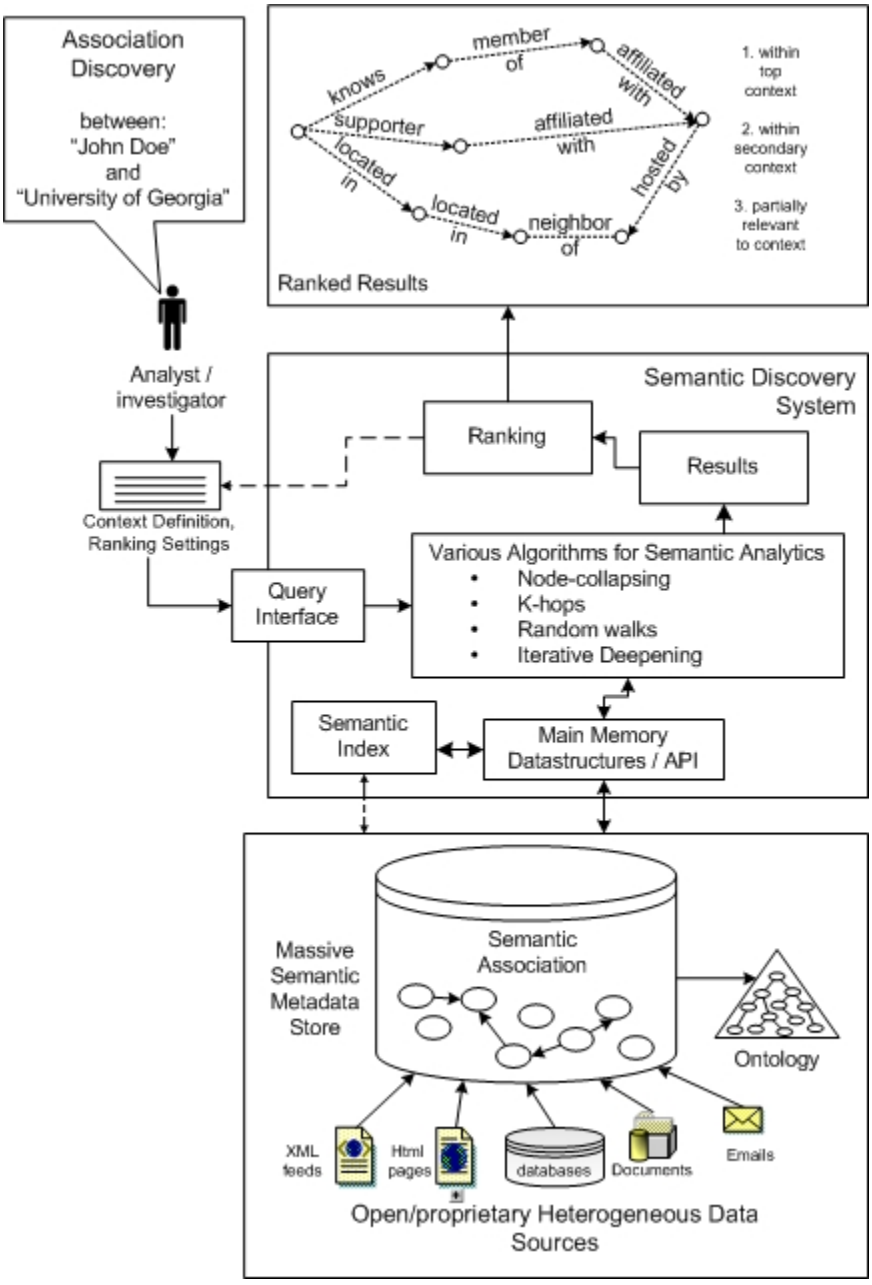

Figure 5.1 SemDIS System Architecture

5.1 Massive Semantic Metadata Store

As discussed in Chapter 2.3, there are a variety of scalable RDF/OWL databases available today. However, our work on the SemDIS project has exposed shortcomings of a majority of these systems. As mentioned before, in the project we have formally defined complex relationships

between two entities in a semantic knowledge-base as Semantic Associations [Anyanwu03, Sheth04a]. Thus, in the project's most generic and simplistic form, we are developing algorithms to traverse semantic graphs (i.e., RDF knowledge-bases), in search of connections between entities. Given this, we have found that many of the APIs of the current systems are often lacking support for this type of functionality. Often, their APIs are not comprehensive enough to provide efficient traversals of the RDF graph. The query mechanisms/languages provided within the systems also seem to lack the primitives and expressiveness to effectively achieve this goal as well. Additionally, once the Semantic Associations are discovered, the project then requires that they be indexed. The team has also found that the current systems provided few, if any, capabilities (e.g., APIs) for building customized indexes for increased performance.

This has led the team to design our own native, main memory representation of the RDF/OWL data, with an accompanying API. The main focus was to provide an effective and efficient API to traverse the data in search for Semantic Associations, as well as allow for the creation of indexes at the time the data was parsed, in addition to when associations were discovered. In the future, the team potentially hopes to collaborate with ICS-FORTH, in the expansion of RDFSuite to be ideal for handling the discovery of Semantic Associations. The details of this main memory structure are omitted from this discussion, as they are not the focus of this work. All code for the main memory representation, as well as documentation for the API is publicly available on the SemDIS project page (http://lsdis.cs.uga.edu/Projects/SemDis/).

5.2 Query Engine

Due to the underlying graph data model of RDF, Semantic Association queries between two entities can be viewed as a 'find all paths' problem. In this respect, the team has adapted, fully implemented and tested various graph traversal algorithms based on k-hops, random walks and iterative deepening. We are additionally developing heuristics for semantics-based discovery (e.g., exploiting context; see Chapter 4.1.1 and Chapter 7), as well as index structures in order to reduce the time to perform a search. For the purpose of this work, when we discuss the discovery of Semantic Associations and the Query Engine, we assume the use of a naïve depth-first search.

5.3 Semantic Index

Currently, the main index implemented within the SemDIS project is an *Entity Index*. Essentially, when RDF or OWL data is parsed and loaded into main memory (see Chapter 5.1), an index is built based on the entities contained within the RDF data. When the data is parsed, the labels (e.g., strings) associated with the entity are used as keys in a hash table. Thus, strings map to actual entities in the knowledge-base. The motivation for this was to ease the user's interaction (e.g., identifying entities to search for associations between) with the system. Essentially, s/he can refer to entities with simple/smaller strings, rather than URIs.

5.4 Ranking Configuration

In the current SemDIS system implementation, the user is provided with a Web interface that gives her/him the ability to customize the ranking criteria as defined in Chapter 4. Of particular interest is the manner in which the user defines the query context. In the SemDIS project, we utilize a modified version of TouchGraph [TouchGraph], a Java applet for visual interaction with

a graph, to define a query context. Prior to a query, the user can define contextual *regions* (subsets of the visualized ontology schema), with their associated weights using this graphical interface. An enlarged snapshot of the context definition interface is provided below in Figure 5.2 [Halaschek04a].

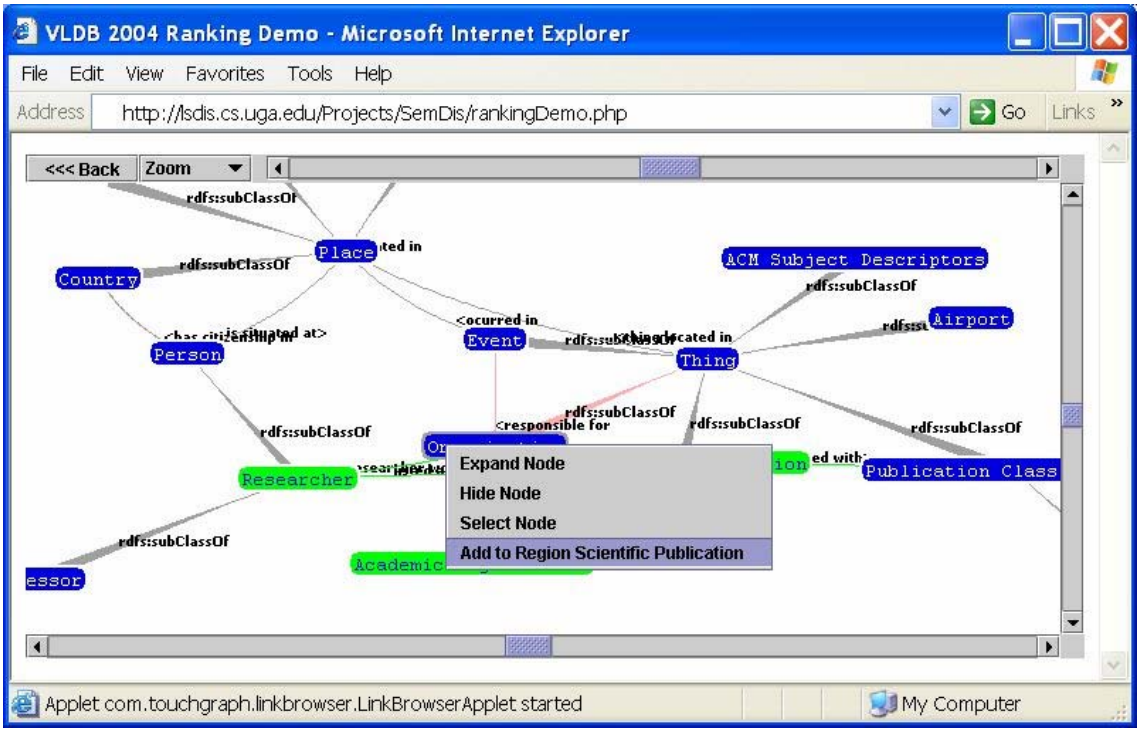

Figure 5.2 Context Definition Interface

This interface is embedded into a main ranking configuration screen in which the user can configure the other metrics and their associated values as well. This is demonstrated below in Figure 5.3.

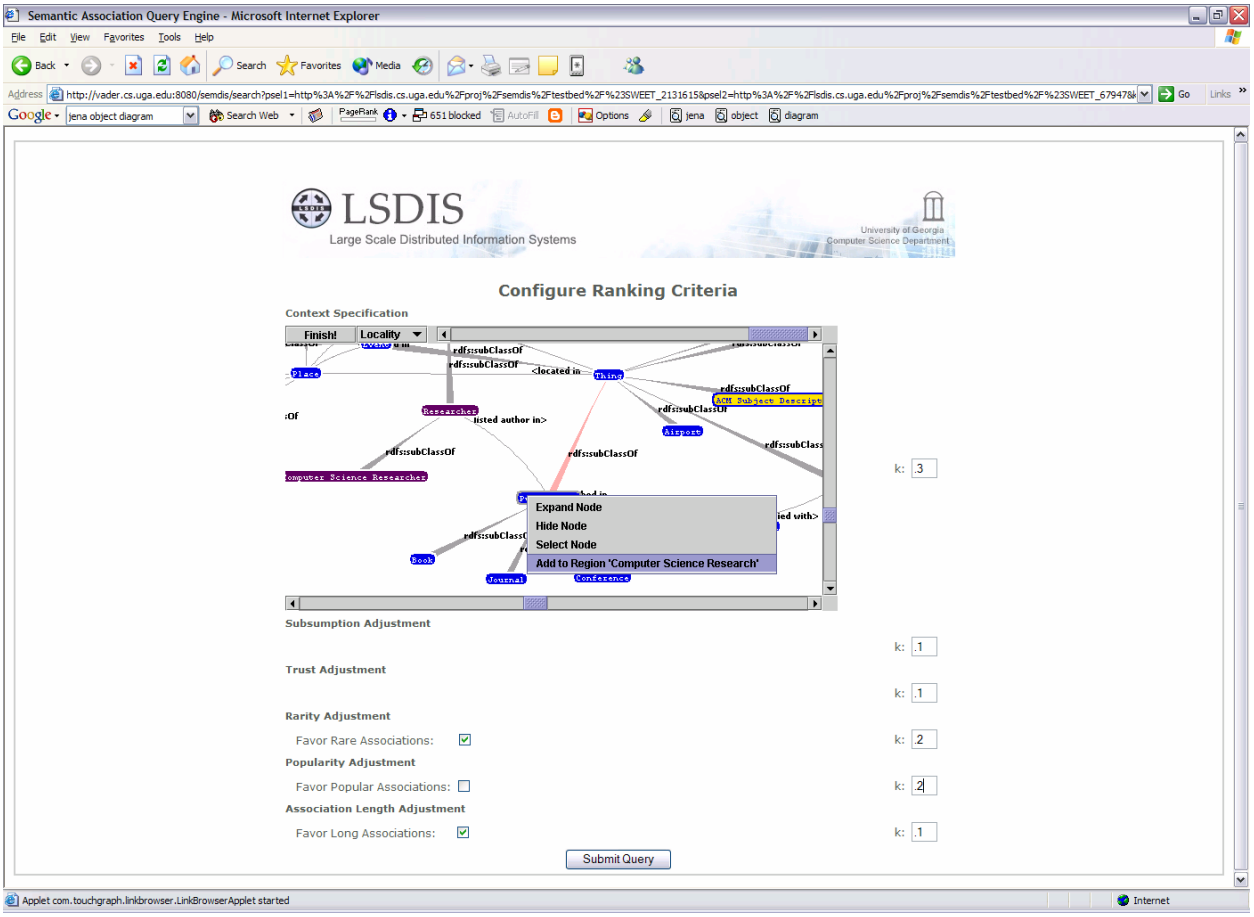

Figure 5.3 Ranking Configuration Interface

5.5 Ranking Module

After the ranking criteria are customized, a Semantic Association query is issued to the Query Engine. This results in an unranked, randomly sorted list of *Associations* (a Java class representing a sequence of entities and properties). These unranked *Associations* are then passed to the Ranking Module. Essentially, the unranked associations are traversed and ranked according to the ranking criteria defined by the user. The task of assigning a rank to an associatoin is decomposed into finding the rarity, popularity, and subsumption rank of all entities in each association. Additionally, the popularity and trust value of each relationship should be determined during the traversal as well. In the current implementation, when the RDF/OWL data

is parsed, rarity, popularity, trust, and subsumption statistics (as required in the formulas defined in Chapter 4) of both entities and relationships as determined with simple counters associated with the data structures. Determining the association length of an association is trivial, as well. Determining the context rank of an association, again, is a simple process of checking which context regions, if any, each entity or relationship in each association belongs to. The pseudocode for the algorithm, given an association, is presented in Table 5.1. Note that in the algorithm, we generically refer to both nodes and properties as *resources*.

Table 5.1 Pseudocode for the Ranking Algorithm

```
/* Initialize variables for ranking score */
double Rank ,rS, rR, rP, rC, rPL, rT = 0.0

/* Loop through the association */
For each resource, r, in association
  {
        /* Get the type of the resource (class type or property type) */
       type = resource.type

       /* Update Subsumption rank */
       rS = rS + ( type.getLocationInHeirarchy() / type.getHeirarchyHeight() )

       /* Update Trust rank */
      If rT < resource.trust
         rT = resource.trust

      /* Check if the resource is a node */
      If resource is a 'node'
         {
            /* Update Rarity rank */
           rR = rR + ( ( graph.instances.getNumNodes() -
                         graph.instances.getNumNodeType(type) ) /
                         graph.instances.getNumNodes() )

            /* Update Popularity rank */
           rP = rP + resource.getProperties().size() /
                     graph.instances.getNumNodeMaxEdges( type ))
         }
```

```
        /* Check if the resource is a property */
        If resource is a 'property'
           {
              /* Update rarity rank */
              rR = rR + ( ( graph.instances.getNumRels() -
                            graph.instances.getNumRelType(type) ) /
                            graph.instances.getNumRels() )
           }

           /* Get context weight of the resource */
           cw = context.relevancy( resource )

           /* Increment Context rank */
           if( cw == 0.0 )
              notContext++
           else
              rC = rC + cw
}

/* Set overall Subsumption rank */
rS = ( rS / association.length( ) )

/* Set overall Context rank */
rC = ( 1.0 / association.length( )) * (rC  * (1.0 - ( notContext / association.length() ) ) )

/* Set overall Rarity rank */
If favor rare associations
      rR = ( rR / association.length( ) )
else
      rR = 1.0 – (rR / association.length( ) )

/* Set overall Popularity rank */
If favor popular associations
      rP = ( rP / association.length( ) )
else
      rP = 1.0 – (rP / association.length( ) )

/* Set overall Association Length rank */
If favor long associations
      rPL = 1.0 - ( 1.0 / association.length( ) )
else
      rPL = 1.0 / association.length( )

/* Set overall association rank */
Rank = (KC *  rC) + (KS * rS) + (KL * rPL) + (KT * rT)  + (KR * rR)  + (KP * rP)
```

Once the associations are ranked, a sorted (by rank) list of *Ranked Associations* (Java class with the original association as well as the various criteria ranks associated with the particular association) is returned to the user interface. This then allows for the presentation of the ranked associations, along with a summary of their criteria influence. A screenshot of ranked results is presented below in Figure 5.4.

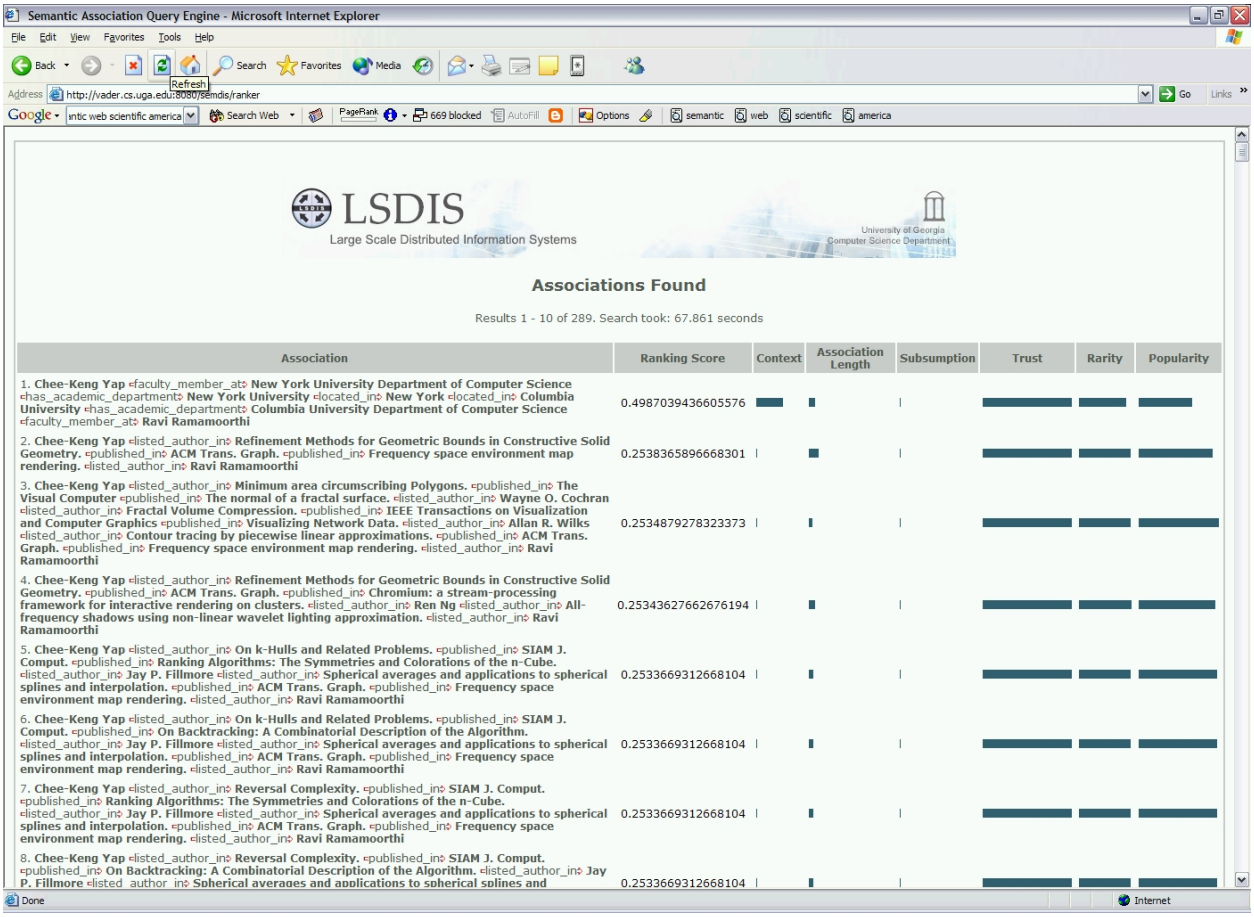

Figure 5.4 Ranked Results Interface

CHAPTER 6

RANKING EVALUATION

The ranking approach presented in this work has been evaluated over the SWETO test-bed. This chapter presents some background information regarding the development of SWETO, as well as the details and findings of the ranking evaluation.

6.1 SWETO Test-bed

SWETO is an ontology that has been developed by the SemDIS project team. It is an ongoing effort for the development of a large scale test-bed ontology that incorporates instances extracted from heterogeneous Web sources. The SWETO ontology was created in a bottom-up fashion where the data sources dictate the classes and relationships defined in the ontology, similar in spirit to the concept of emergent semantics [Staab02, Kashyap01]. In SWETO, the ontology was created using Semagix Freedom (as detailed in Chapter 2.1.1). Figure 6.1 presents a visualization of the current SWETO schema.

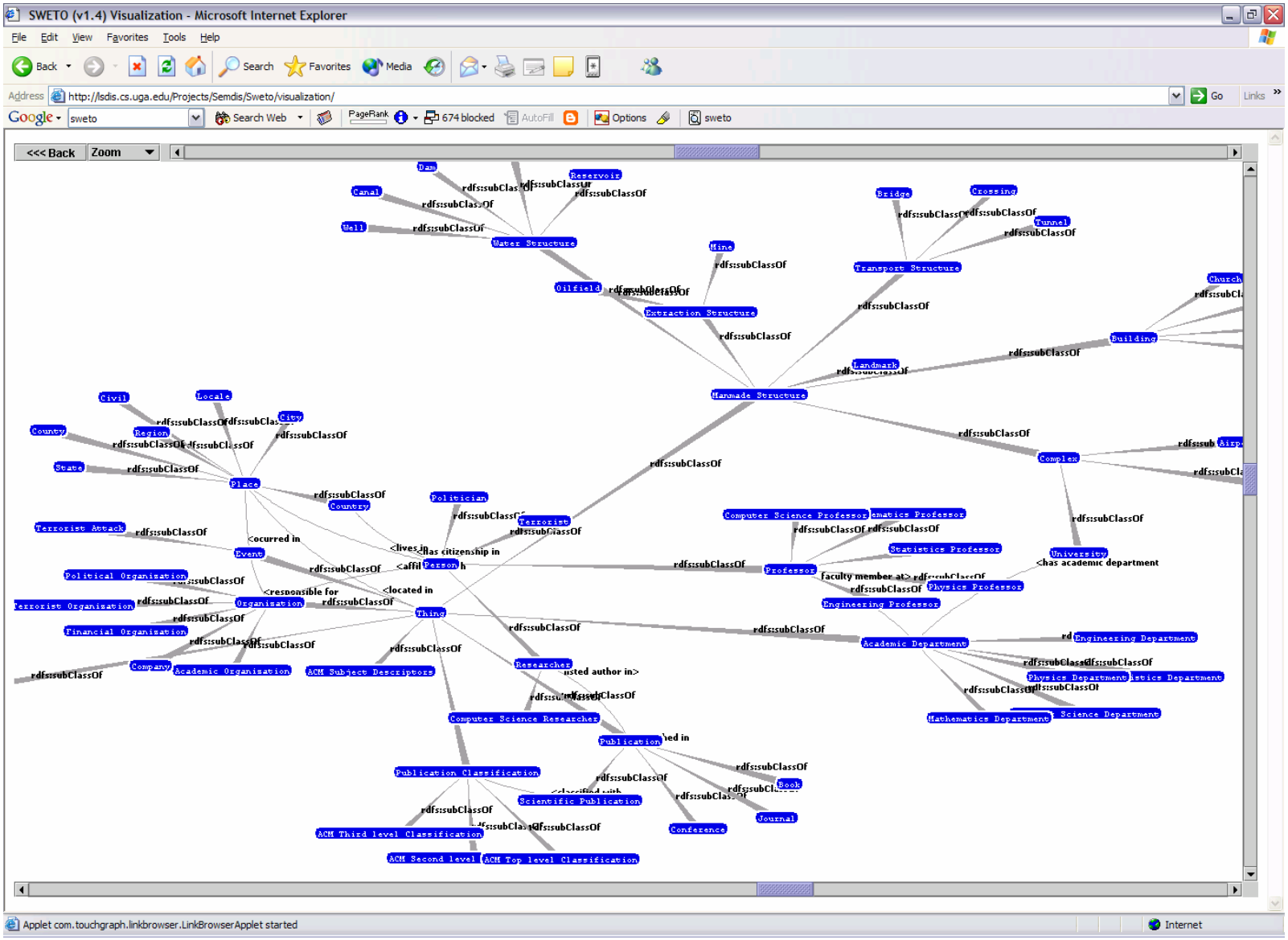

Figure 6.1 Current SWETO Schema Visualization

The creation of the SWETO test-bed required meticulous selection of data sources. Sources were selected based on the following factors:

(i) Selecting sources which were highly reliable Web sites that provide entities in a semi-structured format, unstructured data with parse-able structures (e.g., html pages with tables), or dynamic web sites with database back-ends.

(ii) The team carefully considered the types and quantity of relationships available in a data source. Therefore, we preferred sources in which instances were interconnected.

(iii) We considered sources whose entities would have rich metadata. For example, for a 'Person' entity, the data source also provides attributes such as gender, address, place of birth, etc.

(iv) Public and open sources were preferred, such as government Web sites, academic sources, etc. because of our desire to make SWETO openly available.

The current population of the SWETO ontology includes over 800,000 entities and over 1,500,000 explicit relationships among them. More details regarding SWETO can be found at the project homepage (http://lsdis.cs.uga.edu/Projects/SemDis/Sweto/) or in [Aleman-Meza04].

6.2 Evaluation Overview

Due to the subjective nature of ranking Semantic Associations, traditional evaluation metrics such as precision and recall do not accurately measure the effectiveness of our ranking approach. In fact, recall provides no insight into the algorithm due to the fact that the discovery engine

finds all associations. Given the various ways to interpret these relationships, we evaluated our ranked results with respect to those obtained by a panel of five human subjects. The human subjects were given randomly sorted query results from different Semantic Association queries (each consisting of approximately 50 results). 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 relations 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 (i.e., target match).

6.3 Sample Queries

Due to the large number of ways in which the criteria can be customized (e.g., favor long and rare vs. short and popular associations), we have evaluated five combinations. While this is a small test set, we feel it is a representative sample of these combinations. In each of the test queries, we have emphasized (highly weighted) two of the criteria. Table 6.1 presents the ranking criteria and broader impact of each query.

Table 6.1 Sample Queries

Query #	Query Details	Impact
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 and journal publications.	Demonstrates the ability to capture the user interest in finding strong or close 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 (math departments and professors).	Shows the systems flexibility to highlight common relationships. This may be relevant, for example, when trying to model the way a person is related to entities in a similar manner as the common public.
4	Between a '*Person*' and a '*Financial Organization*', in which long associations and the financial domain context are favored.	Generally relevant for semantic analytics applications, such as those involving money laundering detection [Krebs 2002, Semagix Inc. 2003].
5	Between two '*Persons*' where 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 (hotspots), such as countries.

6.4 Evaluation

In order to demonstrate the effectiveness of the ranking scheme, we illustrate, below in Figure 6.2, 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). Additionally, Figure 6.3 illustrates the average distance of the rank (based on relative order) assigned by the system from that given by the human subjects.

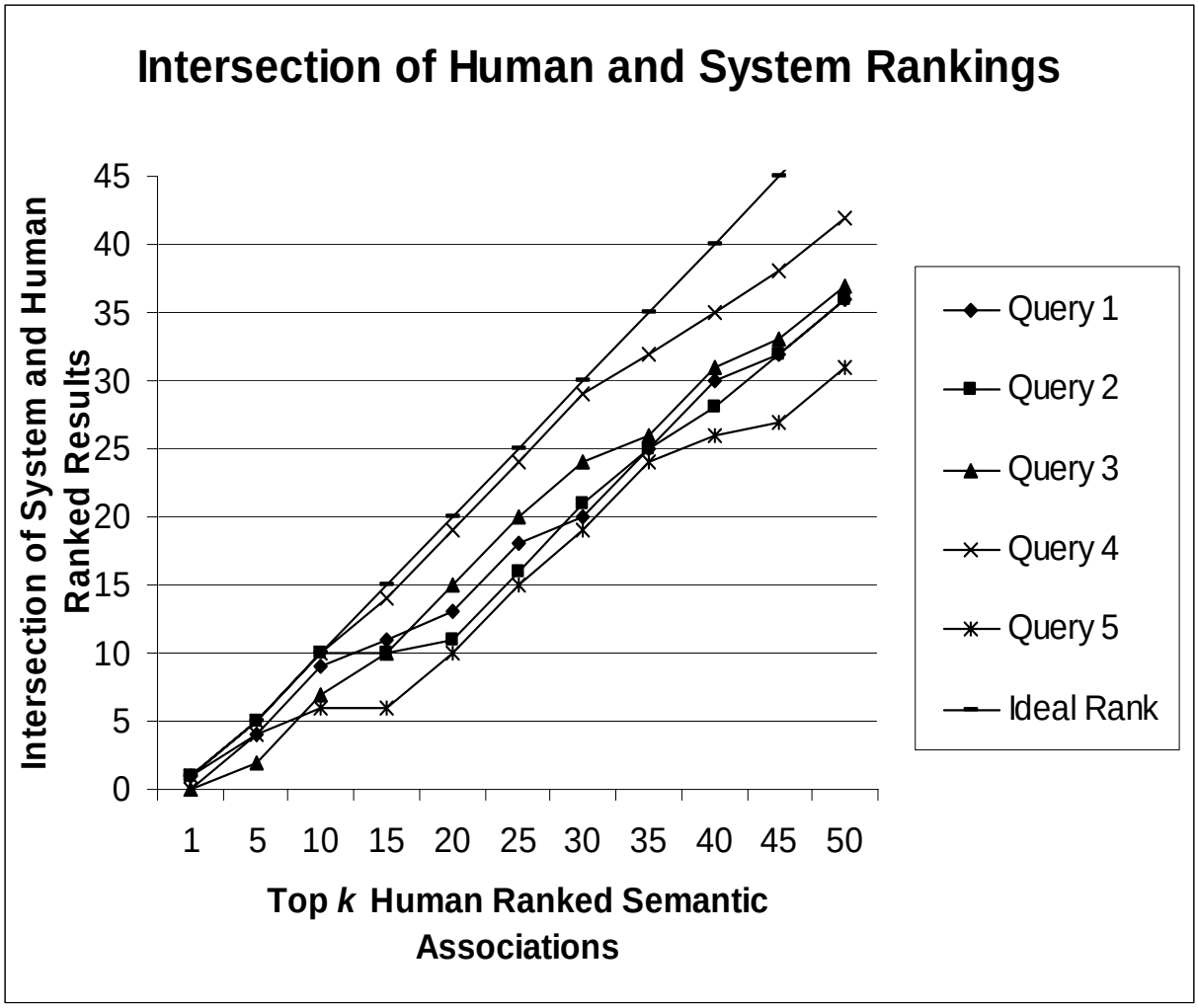

Figure 6.2 Measures of Rank Intersections

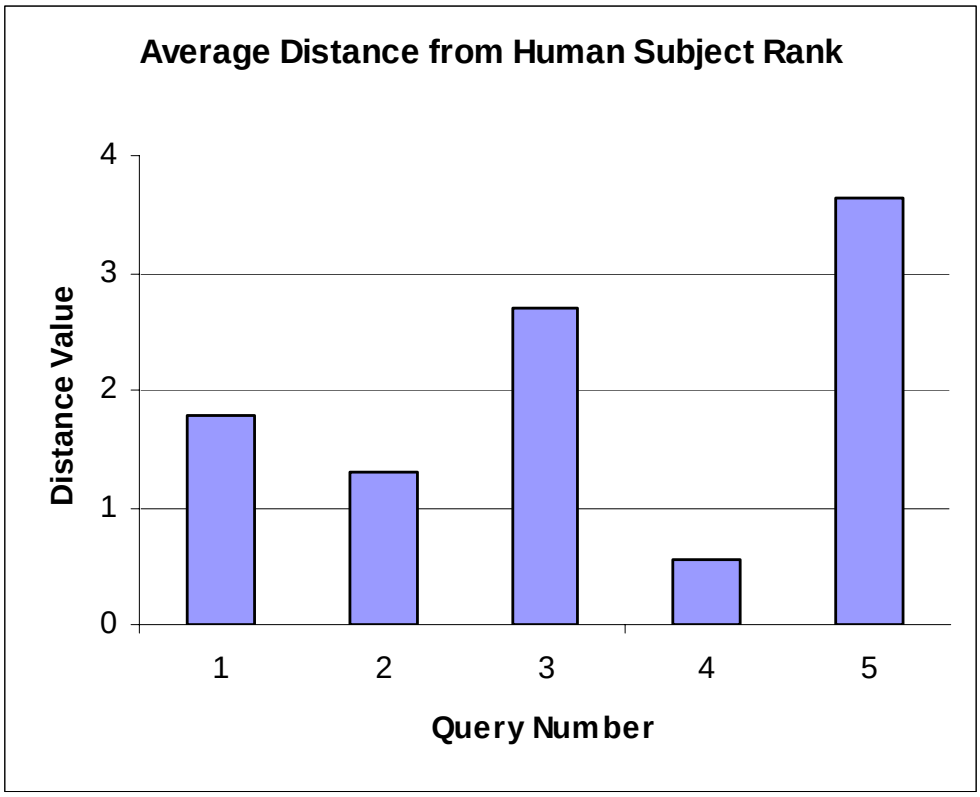

Figure 6.3 Average Distances of Human and System Ranks

Through the results illustrated the figures above, some interesting observations about the preliminary evaluation can be made. First, it is shown in Figure 6.2 that 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. This demonstrates, over this general sample, the approach's ability to locate the most relevant result with respect to a user-defined criterion. The results are even more promising, given that out of the top ten human-ranked results, the system averaged 8.4 matches. In Figure 6.3, it is interesting to note that the minimum average distance of the system assigned ranks from that of the human subjects for a query (considered in relative order) was 0.55, while the maximum never exceeded 4. This demonstrates that the error in the ranking,

when compared to the human subject's, was minimal. While this is a limited, initial evaluation, we conclude that 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.

6.5 Advantages and Limitations

Our particular ranking approach offers several advantages, but suffers from some limitations as well. The key advantages of our approach stem from its ability to model and capture a user's interest. This is a result of comprehensive coverage of the ranking criteria presented to the user upon issuing a query. However, when assessing some of the criteria, some limitations can be identified.

Through our investigations, we have found that the Subsumption criterion is most effective when the underlying ontology of the knowledge-base being queried is very specific and complete; this is where the metric gains its effectiveness. Therefore, if the ontology is not very specific, in respect to its class and relationship hierarchies, then many associations will have the same Subsumption rank. However, this can be overcome by accurate, in depth, domain modeling when developing an ontology schema, as well as highly accurate and specific classification of annotated entities with respect to the ontology.

For the Trust rank to be as effective as possible, there should be a large number of data sources (Web pages) from which the metadata is extracted. A consequence of having a limited number of data sources will be a great deal of similarities among trust values. This again will result in many associations having the same Trust ranks.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Given the current developments in Semantic Web research, next generation technologies that facilitate getting actionable knowledge and information from semantic meta-data extracted from Web documents, the deep Web and large enterprise repositories are emerging. Through our past and ongoing work in metadata extraction, as well as the definition and discovering for complex relationships on the Semantic Web, called Semantic Associations, we see the need for new ranking techniques to assess the relevance of these associations due to the large number of results from queries.

Since Semantic Associations are based on metadata extracted from heterogeneous documents and a set of potentially complex relationships between these metadata, we have discovered that there is no one way to measure their relevance. Thus, though this work we have researched and defined a flexible, query dependant approach for automatically analyzing and relevantly ranking the resulting associations. Additionally in this work, we have presented an implementation of such an approach, as well as empirically evaluated the ranking scheme. Through this evaluation we have found that our proposed approach is able to capture the user's interest and rank results in a relevant fashion.

Our potential future work includes many directions. First, is the notion of '*ranking-on-the-fly*'. By this, we mean that as the Query Engine traverses the RDF graph, partial ranks could be assigned to the potential associations which the algorithm is traversing. This could result in

performance improvements, as the associations would not have to be ranked after the discovery process. Related to this, are additional ideas for improving the Semantic Association discovery algorithms using the ranking scheme that we have presented in this work. This could potentially provide for better scalability in finding Semantic Associations in very large data sets. One idea is to utilize context (as discussed in Chapter 4.1.1) as a heuristic in guiding the depth-first search discovery algorithm. Essentially, associations which are stepping into contextual regions of interest could be traversed first. Another idea related to '*ranking-on-the-fly*', is that while the discovery algorithm is running, associations that fall below a predetermined minimal rank could be pruned, under the assumption that they will have little to no relevance to the user's query.

Lastly, in this work we have only proposed an approach to rank *Semantic Connectivity* associations (as defined in Chapter 2.4). A next step to further this effort is to address the ranking of Semantic Similarity associations. This could potentially reuse some of the ranking criteria presented here, as well as introduce new criteria specifically designed for ranking of these different types of Semantic Associations.

REFERENCES

[Aleman-Meza03] ALEMAN-MEZA, B., HALASCHEK, C., ARPINAR, I. B., AND SHETH, A. 2003. Context-Aware Semantic Association Ranking. In Proceedings of the First International Workshop on Semantic Web and Databases (Berlin, Germany 2003).

[Aleman-Meza04] ALEMAN-MEZA, B., HALASCHEK, C., SHETH, A., ARPINAR, I. B., AND SANNAPAREDDY, G. 2004. SWETO: Large-Scale Semantic Web Test-bed. In Proceedings of the International Workshop on Ontology in Action (Banff, Canada June 20-24, 2004).

[Alexaki01] ALEXAKI, S., CHRISTOPHIDES, V., KARVOUNARAKIS, G., PLEXOUSAKIS, D., AND TOLLE, K. 2001.The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases. In 2nd International Workshop on the Semantic Web, in conjunction with Tenth International World Wide Web Conference (Hong Kong, May 1, 2001).

[Anyanwu03] ANYANWU, K., AND SHETH, A. 2003. r-Queries: Enabling Querying for Semantic Associations on the Semantic Web. In Proceedings of the 12th International World Wide Web Conference (WWW-2003) (Budapest, Hungary, May 20-24 2003).

[Bechhofer03] BECHHOFER, S., HARMELEN, F. V., HENDLER, J., HORROCKS, I., MCGUINNESS, D. L., AND PATEL-SCHNEIDER, P. F. 2003. OWL

Web Ontology Language Reference. W3C Proposed Recommendation, WWW Consortium (Cambridge, MA 2003).

[Berners-Lee01] BERNERS-LEE, T., HENDLER, J., AND LASSILA, O. 2001. The Semantic Web. Scientific American, (May 2001).

[Brickley00] BRICKLEY, D., AND GUHA, R. V. 2002. Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation (March 2000).

[Brin98] BRIN, S., AND PAGE, L. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Proceedings of the 7th International World Wide Web Conference.

[Broekstra02] BROEKSTRA, J., KAMPMAN, A., AND HAEMELEN, F. 2002. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. Semantic Web Conference (Sardinia, Italy 2002).

[Crowley02] CROWLEY, J. L., COUTAZ, J., REY, G., AND REIGNIER, P. 2002. Perceptual Components for Context Aware Computing. International Conference on Ubiquitous Computing (Goteborg, Sweden, September 2002).

[Dill03] DILL, S., EIROL, N., GIBSON, D., GRUHL, D., GUHA, R., JHINGRAN, A., KANUNGO, T., RAJAGOPALAN, S., TOMKINS, A., TOMLIN, J. A., AND ZIEN, J.Y. 2003. SemTag and Seeker: Bootstrapping the semantic Web via automated semantic annotation. In Proceedings of the 12th International World Wide Web Conference (WWW-2003) (Budapest, Hungary, May 20-24 2003).

[Guha03] GUHA, R. V., AND MCCOOL, R. 2003. TAP: An Semantic Web Test-bed. Journal of Web Semantics, Volume 1, Issue 1 (December 2003).

[Guha91] GUHA, R. V. 1991. Contexts: A Formalization and Some Applications. PhD thesis, Stanford University.

[Halaschek04a] HALASCHEK, C., ALEMAN-MEZA, B., ARPINAR, I. B., AND SHETH, A. 2004. Discovering and Ranking Semantic Associations over a Large RDF Metabase. 30th International Conference on Very Large Data Bases (VLDB2004) (Toronto, Canada August 30 - September 03, 2004) Demonstration Paper (accepted).

[Halaschek04b] HALASCHEK, C., ALEMAN-MEZA, B., ARPINAR, I. B., RAMAKRISHNAN, C., AND SHETH, A. 2004. A Flexible Approach for Analyzing and Ranking Complex Relationships on the Semantic Web. 3rd International Semantic Web Conference (submitted).

[Hammond02] HAMMOND, B., SHETH, A., AND KOCHUT, K. 2002. Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content in Real World Semantic Web Applications. V. Kashyap & L. Shklar, Eds., IOS Press.

[Handschuh03] HANDSCHUH, S., AND STAAB, S. 2003. CREAM CREAting Metadata for the Semantic Web. Computer Networks. 42: 579-598, Elsevier.

[Handschuh02] HANDSCHUH, S., STAAB, S., AND CIRAVENGA, F. 2002. S-CREAM Semi-automatic CREAtion of Metadata. In Proceedings of the 13th International Conference on Knowledge Engineering and Management (Sigüenza, Spain October 1-4 2002).

[Jena] Jena Homepage: http://www.hpl.hp.com/semweb/jena.htm

[Karvounarakis02] KARVOUNARAKIS, G., ALEXAKI, S., CHRISTOPHIDES, V., PLEXOUSAKIS, D., AND SCHOLL, M. 2002. RQL: a declarative query language for RDF. Proceedings of the 12th International World Wide Web Conference, pp.592-603.

[Kashyap01] KASHYAP, V., AND BEHRENS, C. 2001. The Emergent Semantic Web: A Consensus approach for Deriving Semantic Knowledge on the Web. Proceedings of the International Semantic Web Working Symposium (Stanford, USA July 2001).

[Kashyap96] KASHYAP, V., AND SHETH, A. 1996. Semantic and schematic similarities between database objects: a context-based approach. VLDB Journal 5: 276–304.

[Kleinberg99] KLEINBERG, J. 1999. Authoritative sources in a hyperlinked environment. Journal of ACM (JASM), 46.

[Krebs02] KREBS, V. 2002. Mapping Networks of Terrorist Cells. Connections, 24(3): 43-52.

[Lassila99] LASSILA, O., AND SWICK, R.R. 1999. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, WWW Consortium (Cambridge, MA 1999).

[Lin03] LIN, S., AND CHALUPSKY, H. 2003. Unsupervised Link Discovery in Multi-relational Data via Rarity Analysis. The Third IEEE International Conference on Data Mining.

[Maedche01] MAEDCHE, A., STAAB, S., STOJANOVIC, N., STUDER, R., AND SURE, Y. 2001. SEmantic PortAL – The SEAL approach. In: Fensel, D., Hendler, J., Lieberman, H., Wahlster, W (eds.): In Creating the Semantic Web. D. MIT Press, MA, Cambridge.

[McBride01] McBride, B. 2002. Jena: Implementing the RDF Model and Syntax Specification. Proceedings of the Second International Workshop on the Semantic Web (Hong Kong, China, May 1 2001).

[Quint03] QUINT, B. 2003. IBM's WebFountain Launched - The Next Big Thing?. Information Today, Inc., (September 22, 2003) (available at http://www.infotoday.com/newsbreaks/nb030922-1.shtml).

[Rodriguez03] RODRIGUEZ, M., AND EGENHOFER, M. 2003. Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Trans. on Knowledge and Data Engineering, 15(2).

[RDFSuite] RDFSuite Homepage: http://139.91.183.30:9090/RDF/

[Seaborne04] SEABORNE, A. 2004. RDQL - A Query Language for RDF. W3C Submission, WWW Consortium (Cambridge, MA 2004), available at http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/

[Semagix03] Semagix Inc. 2003. Anti Money Laundering, Application White Paper. Available at http://www.semagix.com/pdf/anti_money_laundering.pdf

[Shah02] SHAH, U., FININ, T., JOSHI, A., COST, R.S., AND MAYFIELD, J. 2002. Information Retrieval on the Semantic Web. In Proceedings of the

10th International Conference on Information and Knowledge Management.

[Sheth04a] SHETH, A., ALEMAN-MEZA, B., ARPINAR, I. B., HALASCHEK, C., RAMAKRISHNAN, C., BERTRAM, C., WARKE, Y., AVANT, D., ARPINAR, S., ANYANWU, K., AND KOCHUT, K. 2004. Semantic Association Identification and Knowledge Discovery for National Security Applications. Special Issue of Journal of Database Management on Database Technology for Enhancing National Security, Eds: L. Zhou and W. Kim.

[Sheth04b] SHETH, A., AND AVANT, D. 2004. Semantic Visualization: Interfaces for exploring and exploiting ontology, knowledgebase, heterogeneous content and complex relationships. NASA Virtual Iron Bird Workshop (California March 31 - April 2, 2004).

[Sheth03a] SHETH, A., ARPINAR, I. B., AND KASHYAP, V. 2003. Relationships at the Heart of Semantic Web: Modeling, Discovering, and Exploiting Complex Semantic Relationships. In: Nikravesh, M., Az-vin, B., Yager, R., Zadeh, L. (eds): Enhanceing the Power of the Internet Studies in Fuzziness and Soft Computing. Springer-Verlag.

[Sheth02] SHETH, A., BERTRAM, C., AVANT, D., HAMMOND, B., KUCHUT, K., AND WARKE, Y. 2002. Managing semantic content for the Web. IEEE Internet Computing, 6(4), pp 80-87.

[Sheth03b] SHETH, A., AND RAMAKRISHNAN, C. 2003. Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search,

Integration and Analysis. IEEE Data Engineering Bulletin, Special issue on Making the Semantic Web Real (December 2003).

[Staab] STAAB, S. 2002. Emergent Semantics. IEEE Intelligent Systems 17(1) pp. 78-86.

[Stojanovic03] STOJANOVIC, S., STUDER, R., AND STOJANOVIC, L. 2003. An Approach for the Ranking of Query Results in the Semantic Web. In Proceedings of the 2nd International Semantic Web Conference (Sanibel Island, Florida, October 2003).

[Teoma] Teoma Homepage: http://www.teoma.com/

[TouchGraph] TouchGraph Homepage: http://www.touchgraph.com/

[Vargas-Vera02] VARGAS-VERA, M., MOTTA, E., DOMINGUE, J., LANZONI, M., STUTT, A., AND CIRAVEGNA, F. 2002. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In Proceedings of the 13th International Conference on Knowledge Engineering and Management, (Sigüenza, Spain October 1-4 2002).

APPENDIX A

GLOSSARY OF ACRONYMS

AI:	Artificial Intelligence
ARP:	Another RDF Parser
DAML+OIL:	Darpa Agent Markup Language + Ontology Inference Layer
DBMS:	Database Management System
KB:	Knowledge-Base
OWL:	Web Ontology Language
RDF:	Resource Description Framework
RDFS:	RDF Schema
RDQL:	RDF Data Query Language
RQL:	RDF Query Language
RSSDB:	RDF Schema Specific DataBase
SAI:	Semantic Association Identification
SCORE:	Semantic Content Organization and Retrieval Engine
SemDIS:	Semantic DIScovery
SWETO:	Semantic WEb Testbed evaluation Ontology
VRP:	Validating RDF Parser

XML: Extensible Markup Language