Technical Report, LSDIS Lab, Computer Science Department, University of Gerogia, January 2005.

Template Based Semantic Similarity for Security Applications

Boanerges Aleman-Meza, Christian Halaschek-Wiener[1],
Satya Sanket Sahoo, Amit Sheth, I. Budak Arpinar

Large Scale Distributed Information Systems (LSDIS) Lab,
Department of Computer Science
University of Georgia
Athens, GA. 30602-7404
{boanerg, sahoo, amit, budak}@cs.uga.edu
http://lsdis.cs.uga.edu/

Abstract. Today's search technology delivers impressive results in finding relevant documents for given keywords. However many applications in various fields including genetics, pharmacy, social networks etc. as well as national security need more than what traditional search can provide. Users need to query a very large knowledge base (KB) using semantic similarity, to discover its relevant subsets. One approach is to use templates that support semantic similarity-based discovery of suspicious activities, that can be exploited to support applications such as money laundering, insider threat and terrorist activities. Such discovery that relies on a semantic similarity notion will tolerate syntactic differences between templates and KB using ontologies. In this paper, we describe our approach on querying large KBs using template-based similarity performed as part of the SemDIS (Semantic Discovery) project. We also described the associated mechanism for ranking results that are complex relationships between objects (rather than documents in a typical Web search). This approach is prototyped in a system named TRAKS (Terrorism Related Assessment using Knowledge Similarity) and explained using scenarios involving potential money laundering.

1 Introduction

Since the terrorist attack on September 11, 2001, homeland security has been a major topic of both research and application. The identification and possible prevention of various contributing factors to terrorist activities, such as money laundering, identity theft, terrorist planning, etc, are in high demand. In particular, financial institutions are required, by the USA Patriot Act laws (Section 326), to implement a "Customer Identification Program". Anti-money laundering software can automate some of the processing and significantly aid in addressing this need. Primary approaches used for

[1] Currently a Ph.D. student in the Computer Science Dept. at University of Maryland, (email: halasche@cs.umd.edu)

this application are based on simpler solutions that involve person location and identity checking, transaction monitoring and watch-list filtering (e.g, LexisNexis, SearchSpace) as well as more comprehensive analytical processing of data involving data mining techniques and rule based mechanisms. Rule-based systems tend to generate enormous numbers of trivial alerts [15]. While these approaches have benefits, they also have significant limitations. First, detection of the relationships that compose money laundering operations depends on the structure of the database schema being used in a specific financial institution. For the system to be able to find money laundering operations there must be a structural match between the data within the database and known rules and/or patterns. This kind of structural or syntactic detection has an important limitation, since it can miss syntactically different yet semantically equivalent or similar events or acts. Hence it would be beneficial to find not only structural similarity but semantic similarities as well. Second, if the data is semantically annotated, there are possibilities of further finding interesting relations. For example, given a bank in Afghanistan that has personnel with semantic relationships with known terrorists a system based on semantic similarity can consider these relationships relevant in the domain of potential terrorist activities. In contrast, this type of interesting relations may not be easily discovered in traditional fixed rule based systems. This is because for this to be possible, the database schema of the financial institution would have to model such relationships. In this work we propose a system for the detection of money laundering, identity theft, terrorist planning operations and similar activities through the use of semantic relationships modeled using newly emerging semantic Web technologies.

Currently, due to the increasing move from data to knowledge and the rising popularity of the semantic Web vision[2], there is significant interest and ongoing research in automatically extracting and representing semantic metadata as annotations to both documents and services on the Web [9, 13, 21]. Several communities (e.g., Gene Ontology Consortium, Federal Aviation Administration (Aviation Ontology), Molecular Biology Ontology Working Group, just to name a few), 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 (SWB-SIG[3] provides a more extensive list).

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 KB annotated in RDF (Resource Description Framework) that contains information pertaining to authors, sports, companies, etc. [11]. Another repository, SWETO (Semantic Web Technology Evaluation Ontology) [2] is a comparatively narrower but more extensively populated knowledge-base annotated in either OWL (Web Ontology Language) [5] or in RDF (Resource Description Framework) [16] involving rich sets of relationships between entities. We used SWETO as one of our dataset because it includes entities and rela-

[2] http://www.w3.org/2001/sw/

[3] http://business.semanticweb.org/staticpages/index.php?page=20021016230045730

tions of relevance to the terrorism domain (e.g., organizations, persons, watch lists). Additionally, scalable capabilities for semantic metadata extraction and annotations are demonstrated by the WebFountain project, which has annotated and disambiguated data from over 2.5 billion documents [9], and by Semagix Freedom [13], which was used to build SWETO and many other domain ontologies populated with a few million instances [23], and used to semantically annotate millions of documents and Web pages. Examples of commercial deployment of applications exploiting these capabilities are also reported [23].

Research presented in this paper builds upon the above recent advances in the ability to build large populated domain specific ontologies and automatic metadata extraction capabilities. We aim to investigate the similarity of entities and relations based on templates specified through the ontologies. A *template* provides a means to represent a specific manner in which collection of entities are interconnected thus capturing a scenario or a set of circumstances of interest. In particular, we propose a similarity approach that exploits inheritance hierarchies in ontologies to detect similarities semantically. Similarity ranking allows the identification of the most interesting and closer matches. Computing similarity between entities typically requires looking at their syntactical, structural, and semantic properties. Hence a bigger challenge is to compute the similarity of entities with respect to a template that requires not only the entities to be similar to the template, but also to fit the interconnections of the template. To our knowledge, this (and related work in the NSF project on Semantic Discovery (SemDIS)) is the first attempts to semantically identify and rank complex relations in semantic metadata for national security purposes. The contributions of this paper are summarized as follows:

- *Definition of template-based similarity*. We describe querying of RDF data for template-based similarity based on the structural and semantic characteristics of both the template and the ontology.
- *A ranking mechanism to display the results of a template-based similarity query to a user*. The results are ranked based on considerations of their similarity/structural relevance with respect to the template query and the ontology.
- *A proof of concept of template-based similarity*. TRAKS (Terrorism Related Assessment using Knowledge Similarity) is a prototype system where sample scenarios on money laundering illustrate the importance of template-based similarity. The architecture of TRAKS is described.

The outline of this paper is as follows. Section 2 describes how we use Semantic Web technologies in our template based similarity approach. Section 3 provides the framework on which the design decisions are made. We provide the details of the current system implementation in Section 4. The preliminary results over the SWETO testbed are presented in Section 4. Finally, Section 5 gives our concluding remarks and lists future research directions.

2 Background and Related Work

This work is aligned with the current semantic Web vision where ontologies play a central role. Ontologies are conceptualizations of the real-world (i.e., class hierarchies with relationships between them) [10] that can be used to semantically annotate the current information on the Web, in turn help associate meaning to content. 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 to understand and act upon information regarding both the entities and relationships contained on the Web.

When we consider data on the Web, different entities can be related in multiple ways that cannot be pre-defined. In this respect, the RDF data model [16] aims to capture 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" [3]. 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 [24]; for example, in the latter, passenger threat assessment is determined depending on the connections between different people, places and events.

With the recent progress in semantic technology, heterogeneous data on the Web and in Enterprises can be semantically annotated in a scalable manner [9, 13]. Hence, there exist many sources that describe different characteristics of a given entity in diverse domains. We extract and use semantically marked up data along existing data from interested institutions to find potential undesirable and unlawful activities. Our proposed approach employs previously known money laundering, id theft, and terrorist attack templates to discover potential threats in the knowledge base based on novel semantic similarity algorithm that we have developed. Sources for past real money laundering operations are available[4]. Furthermore, financial institutions are required by the Bank Secrecy Act to identify suspicions of money laundering operations and notify authorities [17]. There are also money-laundering requirements imposed as a result of the USA Patriot Act (in particular, section 326).

We demonstrate these capabilities with a prototype (TRAKS) that makes use of data represented in the knowledge representation languages recommended by the W3C for the Semantic Web. In particular, RDF is one of the basic means to provide meaning to data by associating entities to an ontology [16]. We believe that RDF will follow the evolution that XML has had in industry. By adding *tags* to existing data in databases, companies will be able 'output' data marked up using RDF making reference to ontologies agreed upon. Hence, applications (such as TRAKS) can be built to work on RDF data, exploiting the explicit semantics in the data with respect to ontologies. Whereas RDF is intended to mark up data, RDF Schema [7] is the counterpart of RDF that provides the means to specifying a vocabulary of the hierarchy of classes in an ontology and the relationships among them. Our project is an effort to demonstrate that government and commercial organizations would want to use Semantic Web technologies to take advantage of the expressiveness of knowledge representation for diverse types of data (un-structured in nature) about their customers, such as citizenship, companies they own, business associations, credit history, etc. Admittedly, development and use of such technologies raise legitimate privacy con-

[4] For example, see http://www1.oecd.org/fatf/index.htm

cerns specially when certain data for unintended purposes, and will have to be addressed before research represented here can be put to operational use. In that sense, we have addressed only part of the overall challenge. For testing our approach, we use the SWETO dataset, given that it contains entities and relations in the domain of national security.

There have been proposals of query languages for RDF data, such as RQL [14], TRIPLE [25], and RDQL (part of the Jena [18]). However, they do not completely provide means to query RDF data based on a template[5]. Our research addresses querying based a template, thus requiring entities matching the template to be interconnected in the same manner as the template. This 'structure' matching can probably be performed by a quite large set of constrains on a 'WHERE' clause, of say, RQL. However, our template includes capabilities to define optional parts to be matched in a query, which cannot be expressed in the above RDF query languages. Only very recently, querying for optional components has been introduced into SPARQL [19]. While the optional components of our notion of template-based similarity could be expressed in SPARQL, it is not possible to express in a query the semantic similarity measure of the relevance of a template match with respect to the hierarchy of classes and relationships of an ontology. That is, we exploit the explicit (structure) and implicit (semantic) implications of a template with respect to the data in order to find semantically *similar* matches. This is critical in scenarios of money laundering, terrorism activities and identity theft where known operations are being tracked by current money laundering systems. The strength of our approach relies on using *semantic similarity* to find results that are not an exact match of known operations of interest to national security expressed as a template.

We introduce the concept of a 'core' template that captures the essence (that is, the must-have pieces) of a known scenario in either money laundering, terrorism activities and/or identity theft. After filtering out potential matches to known scenarios with the core template, the rest of the template definition provides means to our system to flag a set of entities that are related in a *similar* way to those known a priori.

Earlier applications using graph models have aimed at detecting frequently occurring substructures. The context was in law enforcement applications, in order to track groups involved in crimes [8]. The main aspect that differentiates our work is the use of explicit semantics that allow detecting similar scenarios based on similarity.

3 Template Based Semantic Similarity

The foundations of TRAKS are based on the concepts of a template, core template, ontology, datasets, semantic similarity, and semantic ranking. Templates are a means to capture illegal activities, such as known money laundering operations or an identity theft scheme. The purpose of the template is to model the entities participating in a scenario or event, so that similar scenarios or events could be searched for in a large dataset. In terms of information retrieval, a template can be viewed as a query. The essence of querying through templates is that of interconnecting classes by named

[5] http://lists.w3.org/Archives/Public/www-rdf-interest/2003Nov/0057.html

relationships thus resembling a real-life known a-priori scenario. An intelligence analyst could discuss a terrorism related scenario with colleagues and describe it using a template.

Definition 1. (**Template**) A template is a (connected) directed graph where the nodes represent classes of the ontology of reference, say *O*, and the edges represent relationships connecting the classes. The interconnection of classes and relationships captures a scenario (or model) provided by the user. We formally define a template *T* as follows: $T = <V, E>$ where *V* is the set of vertices (nodes) and *E* is the set of edges. Each $v \in V$ has being given a type *t* that corresponds to a class in the ontology *O* (for example, using rdf:type in the RDF model). Each *v* has a unique identifier (such as URI) thus allowing in the scenario to include more than one node representing entities of the same class. Each edge $e = (u_1, u_2)$, where $e \in E$, and $u_1 \in V$, $u_2 \in V$, the type of *e* must be an existing relationship in the ontology *O* (that is, an existing *property* in the RDF model, or more precisely, in the RDF Schema of *O*).

Definition 2. (**Core template**) A core template is a non empty subset of the template of definition 1. More specifically, a core template *CT* is defined as $CT \subset T$. A core template captures the essence (or most important subset) of the scenario represented in a template. The restriction that the core template is intended to impose is that the classes and relationships that form part of it must match exactly those in the dataset. A benefit of a core template is to provide a means for the system to filter out potential matches to a template.

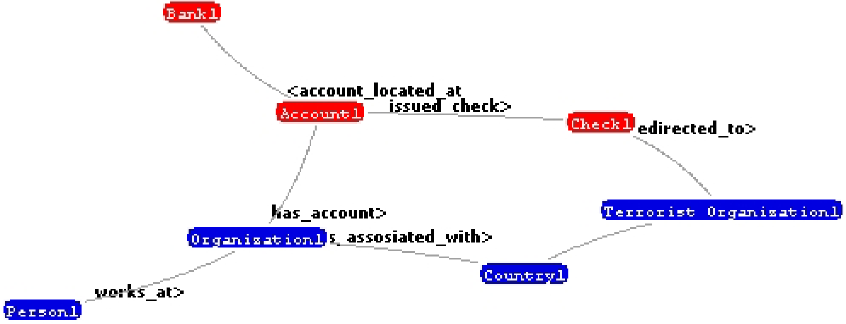

Fig. 1. Visualization of a 'template'

Figure 1 illustrates the visualization of a template. The nodes *Bank1*, *Account1*, and *Check1* denote the 'core template' nodes. The other nodes denote the rest of the (non-core) nodes of the template. The scenario of this particular template involves a person (Person1) that is loosely related to a terrorist organization by means of financial relations to a terrorist organization from the organization for which Person1 works for.

Definition 3. (**Template-based Similarity Query**) The template, as defined above, is the actual query. The items being queried are all those sets of interconnected entities (in a user-specified dataset, e.g., SWETO) that match the template by considering a set of similarity criteria. A result from a template-based *similarity* query is a set of entities from the dataset which are interconnected resembling the graph of the template. The *similarity* matching criteria consists of the following:

i. Each entity *q* matching a node *v* in the core template must belong to the same class as that of node *v*.

ii. Each entity *q* matching a node *v* in the template must either: (a) belong to a class that is subsumed by that of node *v*; or (b) belong to a class that has a common parent class with that of node *v*.

iii. Each relationship *p* matching a relationship *r* in the core template must be exactly the same.

iv. Each relationship *p* matching a relationship *r* in the template must either: (a) be exactly the same; or (b) be subsumed (by means of the transitivity of the 'subProperty' relationship) to relationship *r*.

Two semantically similar entities are, for example 'John', who belongs to the class "CEO", and 'Anna', who belongs to the class "CTO" where both classes have a common parent "Management Board". Two semantically similar relations are, for example 'associated with' and 'member of' where 'member of' is a specialization of the relation 'associated with'.

The semantic ranking criteria considers the overall number of entities and relations that match a template by assigning an evaluation to each of them depending on how distant the class they belong is with the class of the slot in the template. A perfect match is ranked higher and consists of entities and relations that reflect exactly the classes of the template.

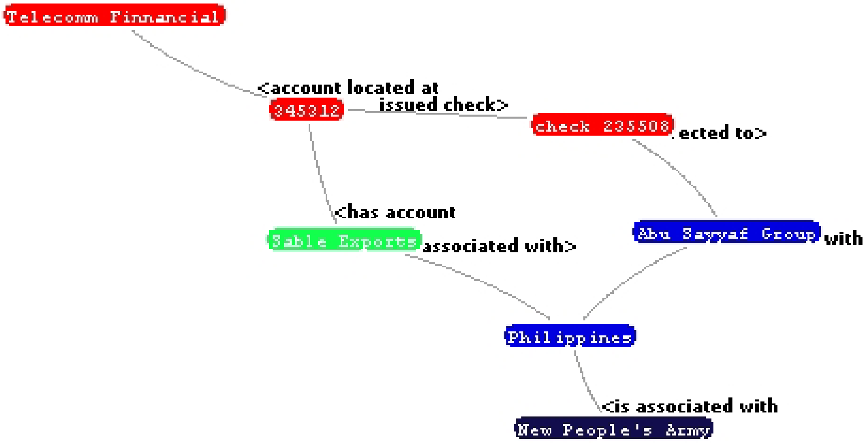

Fig. 2. Result of a template-based similarity query

The results from template-based similarity queries can be displayed in the form of a graph. Figure 2 shows the specific instances of the dataset that are related in a similar form to that of the query template. Coloring of nodes, as in Figure 1, denote matches with core-template and template nodes. Moreover, nodes in other colors represent similarity matches to the type of the node of the template. For example, 'Sable Exports' is a more specific type of 'Organization' in the ontology hierarchy, namely a 'Company'.

Semantic ranking is based on the consideration that entities or relations closer to their respective place in the template are more representative of the scenario that the template aims to capture. Related research in similarity measures [20] considers for example, that entities located at a lower place in an ontology are more specialized than those located at a higher place in the ontology. In order to capture how similar are a class in the template and one in the dataset are, we use the following formula,

$$\mathrm{Sim}(t, c) = \begin{cases} 1, & \text{if typeOf}(t) = \text{typeOf}(c) \\ 1 / \mathrm{Dist}(t, c), & \text{otherwise} \end{cases} \qquad (\mathbf{1})$$

where *t* is a class type from a template and *c* is that corresponding class type from the template match in the result. Dist(*t*,*c*) computes the distance between types in the ontology whereby a penalty factor is added to give a lower ranking to the case where the types *t* and *c* share a common parent class (similarity criterion ii.b above). Given Equation 1, this, the overall rank of a template match is computed by

$$\mathrm{Rank}(m) = \frac{\sum_{i=1}^{\#Nodes(m)} Sim(t_i, m_i)}{\#Nodes(m)} \times \frac{\#Nodes(m)}{\#Nodes(t)} \qquad (\mathbf{2})$$

where *m* is a template match to be ranked and #*Nodes*(*m*) returns the number of nodes (and relations) in a given result or template *m*.

4 System Architecture and Experimental Results

Below, details regarding the various system components and their implementation are provided. 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 template based similarity detection components (such as template creation, template matching engine, ranking, etc). However, the other portions of the system architecture are discussed for here completeness.

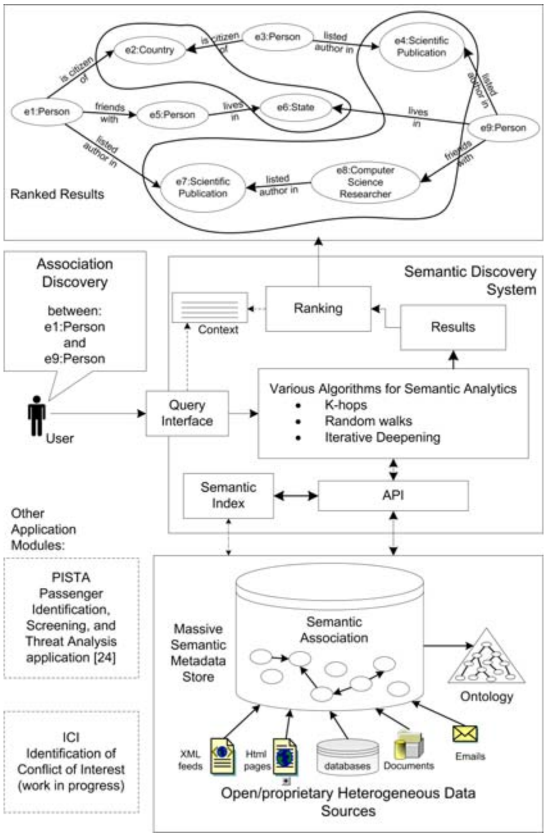
e1:Person
is citizen of
e2:Country
is citizen of
e3:Person
listed author in
e4:Scientific Publication
friends with
e5:Person
lives in
e6:State
lives in
e9:Person
listed author in
listed author in
e7:Scientific Publication
listed author in
e8:Computer Science Researcher
friends with
Ranked Results
Association Discovery
between: e1:Person and e9:Person
User
Query Interface
Semantic Discovery System
Context
Ranking
Results
Various Algorithms for Semantic Analytics
• K-hops
• Random walks
• Iterative Deepening
Semantic Index
API
Other Application Modules:
PISTA Passenger Identification, Screening, and Threat Analysis application [24]
ICI Identification of Conflict of Interest (work in progress)
Massive Semantic Metadata Store
Semantic Association
Ontology
XML feeds
Html pages
databases
Documents
Emails
Open/proprietary Heterogeneous Data Sources

Fig. 3. SemDIS System Architecture

The similarity approach presented in this work has been implemented and tested within the LSDIS lab's SemDIS[6] project. The main components of the SemDIS system architecture are illustrated in Figure 3. The inputs for a semantic association query engine are two entities in the dataset. The query engine then finds all semantic associations between the entities of interest and forwards the results to the ranking module. The ranking module is implemented in Java and responsible from ranking semantic associations [1, 12]. The ranking scores are based on a notion of context (among other factors). A 'context' (for the purpose of this work) is specified by the user as the relevant types and/or relationships that s/he deems important, such as 'Geographic' or 'Scientific Publication' domains (see context areas at top of Figure 3). For this, we utilized a modified version of Touchgraph[7], a Java applet for visual interaction with a graph. Another version of the ranking component has also been

[6] http://lsdis.cs.uga.edu/Projects/SemDIS/

[7] http://www.touchgraph.com

developed which provides the user a capability to move along the dimension of *conventional* to *discovery* search mode [4].

The query processing algorithms for discovery of semantic associations include adapted ideas based on k-hops, random walks, and iterative deepening. We are developing heuristics to prune the search space based on semantics (e.g. through context), as well as index structures in order to reduce the time to perform a search.

It is worth noting that the ranking of semantic associations is quite different than ranking results from a template-based similarity query. In the implementation of the ranking for semantic associations, a user can interact with the ranking module to specify context, whether to favor long or short paths, sources, trustworthiness, etc. Additionally, the user is also able to assign a weight to each of these individual ranking criteria. In contrast, ranking of template-based similarity results follows the intuition on measuring how close a result is to the template. Therefore, both the structural and semantic dimensions are considered. Hence, a separate ranking component is implemented for template-based similarity technique in the TRAKS prototype.

In the SemDIS system, the knowledge extraction module (including the metadata extraction and storage) is implemented using Semagix[8] Freedom, a commercial product which evolved from the LSDIS lab's past research in semantic interoperability and SCORE technology [22]. Using this technology, we have created SWETO, a populated ontology with a large number of instances. It includes organizations, countries, people, researchers, conference, publications, etc., that are explicitly connected by named relationships. Extractors are 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 by the extractors, the extracted entities are disambiguated and 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 in either RDF or OWL syntax.

[8] http://www.semagix.com

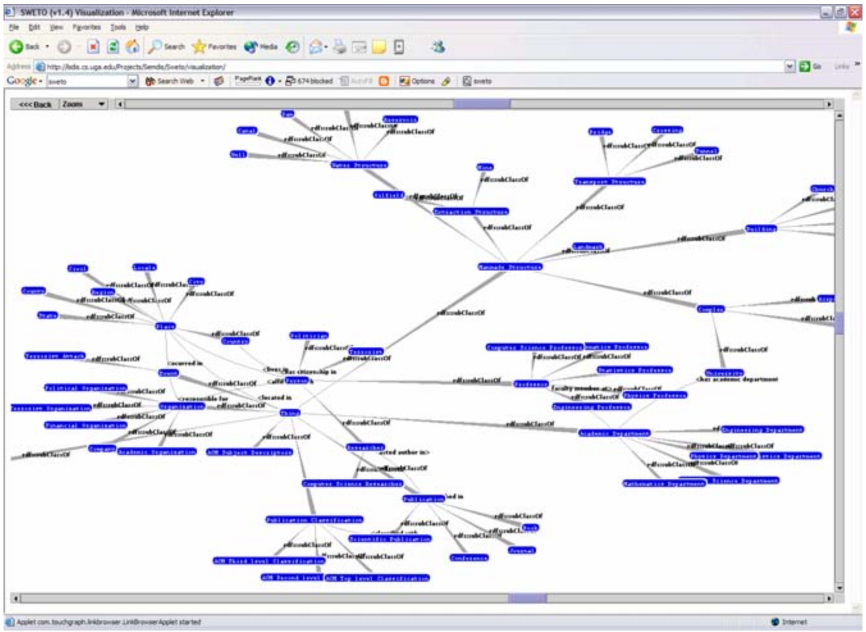

Fig. 4. SWETO ontology (schema)

The creation of the SWETO ontology used in SemDIS and TRAKS evaluations 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. In order to query the SWETO ant its knowledge base, we have implemented a Java API that allows for loading the ontology and its instances into main memory. Thus the system is provided with fast access to the data. Additionally, we implemented a C++ RDF main-memory database that makes use of check-pointing for faster loading (this is work in progress). More details regarding SWETO can be found at the project homepage[9] or in [2].

[9] http://lsdis.cs.uga.edu/Projects/SemDis/Sweto/

4.1 TRAKS Architecture

Figure 5 illustrates the system architecture of TRAKS. The main components are (i) the Web-based user interface, (ii) ranking, (iii) the knowledge base consisting of ontologies, templates and datasets, and mainly (iv) the template matching engine. In order to deal with scalability we are experimenting with data stores for semantic metadata relying on the RDF model (e.g., Jena [18]), and modules that are part of SemDIS designed primarily to support fast algorithms of semantic analytics.

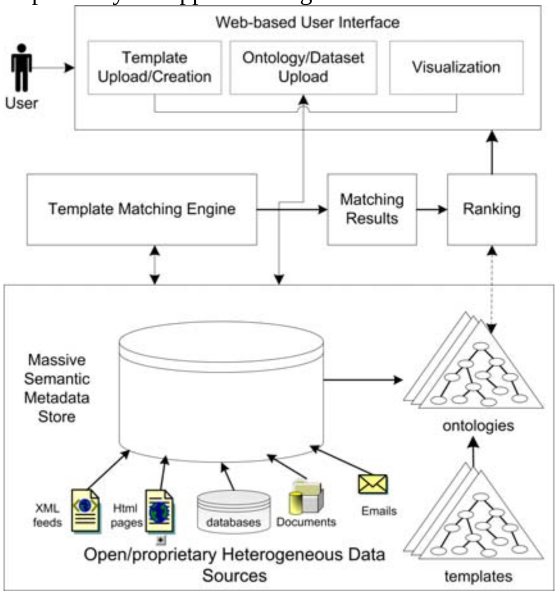

Fig. 5. TRAKS System Architecture

As part of our proof of concept prototype, one of our goals was to test the template-based similarity engine with available datasets and ontologies. Hence we developed TRAKS as a Web application where a user can specify his/her own ontology, dataset, and template. The Web interface[10] of TRAKS contains an "upload" module that allows users to specify their own ontology, dataset and template. An alternative to uploading a template is provided by a module for "template creation". It provides a means to create a template without (XML) syntax complications.

Designing the visual component of the Web interface was facilitated by the usage of Touchgraph. We used a modified version with functionality of labeled edges. For the case of the visualization of templates, coloring of nodes differentiate the classes and relations that belong to the core template. Similarly, input data and ranked results are as well presented to the user with the graphical Touchgraph environment.

Visualization of the results and/or templates gains importance due to the complexity of some well known money laundering operations. As stressed in [26], a visualiza-

[10] http://lsdis.cs.uga.edu/proj/traks/

tion tool provides a means to model the data and also is valuable to investigate a potential money laundering case.

Template Matching Engine. This first phase of the template matching engine proceeds to find core template matches. The matching of the core template proceeds in ascending ordering of the statistics of the number of entities per class. From this point, a recursive depth-first search starts by matching the edges that connect to other non-visited core template nodes. We note here again that the core template matches must be exact matches. This will result in a set of core template matches from the instance data.

Given this, the next phase of the template matching engine is initiated. In this phase, all core template matches are expanded to include any matching regular template edges and classes. We note here that these edges and class types do not have to be exact matches, but can also be similar matches. As in the first phase, the matching of general templates proceeds in ascending ordering of the statistics of the number of entities per class in each core template. Again, from this point a recursive depth-first search starts by matching the edges that connect to other non-visited core template or general template nodes.

Semantic Ranking. The ranking component assigns an estimate value to a matching result based on how the set of entities and relationships reflect those captured by the template. Ranking a set of interconnected entities and relations is more complex than ranking a set of documents or paths of semantic associations [1, 4].

Knowledge Base. This module organizes the ontologies, templates, and datasets provided via Web by users. Besides the Web-accessible uploaded ontologies, we have designed an internal structure that relates ontologies to templates and datasets. Instead of processing XML serializations of these sets, we store them in a prototype knowledge base. The operations of the knowledge base include typical maintenance utilities as well as an API that is used by the template matching engine.

4.2 Template-based Similarity Results

Available data for our experiments is based on the research of SemDIS and Semantic Association Identification and Knowledge Discovery for National Security Applications projects [24]. Therefore, the preliminary ontology is within the domain of terrorism. This has been extended to include entities in the financial domain, such as banks (domestic and international), CEO’s of companies, and so on.

With a subset of the ontology in [24], in this paper we illustrate two cases of template-based similarity matching. These results are then ranked based on distance similarity measures that take into consideration the ontology itself as described in Section 3. Figures 1 and 2 show a template and matching result, respectively. These figures are snapshots of the (TouchGraph) applet we used to visualize the results in a graphical manner. The second case of similarity matching is shown in Figure 6. These are five ranked results of the 16 results of a template similarity query. The result at the bottom of the figure serves to illustrate a case where only nodes of the core-template matched (having the lowest ranking).

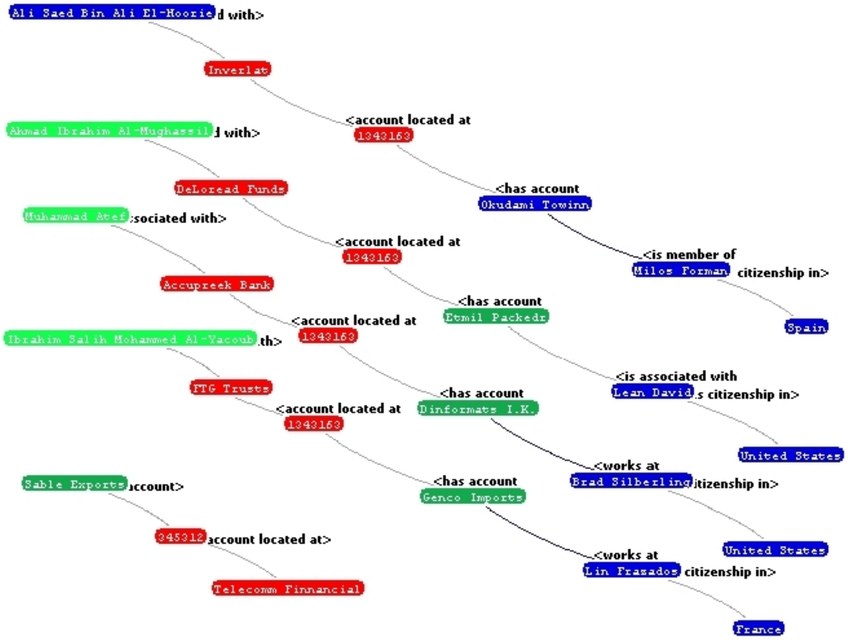

Fig. 6. Matching results of template-based similarity

5 Conclusions and Future Work

Detection of money laundering activities, identity theft operations, and terrorism related activities is a driving motivation to enhanced knowledge discovery research. We addressed the problem of identifying known scenarios described in form of a notion of template-based similarity that we have defined. Search for matching a template in large datasets made use of structure, explicit semantics, and similarity considerations in order to go one step closer to realistic situations where variations of known scenarios are intended to be hidden from traditional exact matching algorithms. We described the TRAKS system, as a proof of concept of the notions of template-based similarity and its associated ranking method. The architecture of TRAKS was designed by considering open standards for knowledge representation that are becoming more popular with the vision of the Semantic Web. TRAKS is a system that is Web-based and makes use of XML and RDF technologies for an open access to it. Therefore, it is possible for anyone who has data annotated with respect to an ontology to use our system. The only step required, template creation, is carried out in a Web-based form. Furthermore, the graph-based visualization component provides support an easier understanding of the results found. The results, ranked based on degree of semantic similarity, as defined in the paper, provide means to financial institutions and/or intelligence agencies to detect potential scenarios of importance for increased economic and national security.

With respect to the API to the knowledge base of TRAKS, we will evaluate the proposed API by [6] that was developed considering different approaches taken by tools that read RDF or OWL data, such as ontology editors, knowledge sources management, and inference engines among others. Several research directions will address scalability issues. Presentation issues can help improve usability. Our initial design of the module of template creation could be further improved with one that only uses Touchgraph events manipulation.

Acknowledgements. This work is funded by NSF-ITR-IDM Award#0325464 titled ‘SemDIS: Discovering Complex Relationships in the Semantic Web’ and NSF-ITR-IDM Award#0219649 titled ‘Semantic Association Identification and Knowledge Discovery for National Security Applications.’ We also acknowledge our use of Semagix Freedom (which is in part a commercialization of the LSDIS lab’s research) as the semantic technology that enabled us to build large populated ontologies used by our TRAKS application.

References

1. Aleman-Meza, B., Halaschek, C., Arpinar, I.B., Sheth, A.: Context-Aware Semantic Association Ranking. In: Proceedings of the First International Workshop on Semantic Web and Databases, pp. 33-50. Berlin, Germany, September 7-8, 2003
2. Aleman-Meza, B., Halaschek, C., Sheth, A., Arpinar, I.B., Sannapareddy, G.: SWETO: Large-Scale Semantic Web Test-bed. Proceedings of the 16th International Conference on Software Engineering and Knowledge Engineering (SEKE2004): Workshop on Ontology in Action, Banff, Canada, June 21-24, 2004, pp. 490-493
3. Anyanwu, K., Sheth, A.P.: r-Queries: Enabling Querying for Semantic Associations on the Semantic Web. Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary, May 20-24, 2003
4. Anyanwu, K., Maduko, A., Sheth, A.P.: SemRank: Ranking Complex Relationship Search Results on the Semantic Web, Proceedings of the WWW2005, Japan, 2005 (accepted, to appear)
5. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference, W3C Canddate Recommendation, August 18, 2003
6. Bechhofer, S., Volz, R., Lord, P.: Cooking the Semantic Web with the OWL API. In: Proceedings of the 2nd International Semantic Web Conference, Sanibel Island, Florida, USA, 2003
7. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, February 10, 2004
8. Chawathe, S.S.: Tracking Hidden Groups Using Communications, NSF/NIJ Symposium on Intelligence and Security Informatics, Tuscon, Arizona, 2003

9. Dill, S., Eirol, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: SemTag and Seeker: Bootstrapping the semantic Web via automated semantic annotation. Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary, May 20-24, 2003

10. Gruber, T.: A Translation Approach to Portable Ontologies. Knowledge Acquisition, 5(2), 1993

11. Guha, R.V., Mccool, R.: TAP: An Semantic Web Test-bed. Journal of Web Semantics, 1(1), December 2003

12. Halaschek, C, Aleman-Meza, B., Arpinar, I.B., and Sheth, A.P., Discovering and Ranking Semantic Associations over a Large RDF Metabase (Demonstration Paper). Proceedings of the 30th International Conference on Very Large Data Bases, Toronto, Canada, August 29–September 3rd, 2004

13. Hammond, B., Sheth, A.P., Kochut, K.: 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

14. Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis D., Scholl, M.: RQL: A Declarative Query Language for RDF, Proceedings of The Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA, May 7-11, 2002

15. Kingdon, J.: “Applying technology to fight money laundering”. Available online at: http://www.bankingmm.com/money_laundering.htm

16. Lassila, O., Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, 1999

17. Lodish, A.D.: How Well Do You Need To "Know Your Customer?" Bank Systems and Technology December 2, 2003, online at:
http://www.banktech.com/story/amLaundering/showArticle.jhtml?articleID=16401323

18. McBride, B.: Jena: Implementing the RDF Model and Syntax Specification. In Proceedings of the Second International Workshop on the Semantic Web, May 2001

19. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF, W3C Working Draft, October 12, 2004

20. Rodriguez, M.A., Egenhofer, M.: Determining Semantic Similarity among Entity Classes from Different Ontologies, IEEE Transactions on Knowledge and Data Engineering, 15(2), 2003

21. Shah, U., Finin, T., Joshi, A., Mayfield J., Cost, R.S.: Information Retrieval on the Semantic Web. In Proceedings of the 10th International Conference on Information and Knowledge Management, November 24, 2002

22. Sheth, A.P., Bertram, C., Avant, D., Hammond, B., Kochut, K., Warke, Y: Managing semantic content for the Web. IEEE Internet Computing, 6(4):80-87, 2002

23. Sheth, A.P.: From Semantic Search & Integration to Analytics, Dagstuhl Seminar on Semantic Interoperability and Integration, September 19-24, 2004, 10 pages. Also presented at KM World, Santa Clara, CA, October 26, 2004

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

25. Sintek, M., Decker, S.: TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. In Proceeding of the International Semantic Web Conference, Sardinia, June, 2002

26. Thomson, J., Brace, M., Hurst, R: A tour of anti-fraud technology. Fraud Intelligence, issue 58, pp. 10-12, June, 2003