

Semantic Graph based Knowledge Discovery from Heterogeneous Information Sources

N. Adam, V. Atluri, V. Janeja, J. Vaidya, M. Youssef
CIMIC, Rutgers University
180 University Avenue, Newark, NJ 07102

A. Suenbuel, C. Bornhoevd, S. Raiyani, T. Lin
SAP Labs
3475 Deer Creek Road, Palo Alto, CA 94304

J. Cooper, J. Paczkowski
The Port Authority of New York and New Jersey

Abstract

Within the Homeland Security domain, it is critical to be able to identify actionable and credible knowledge for the prevention, response and recovery of all hazards to include threat and vulnerability assessment. In such an environment, the difficulty of knowledge discovery is compounded by the fact that data are collected by heterogeneous sources within various agencies operating in disparate mission spaces and having different semantics. One approach to knowledge discovery uses semantic graphs (SGs).

Given a set of SGs from a set of disparate sources and a set of related ontologies, we take a two-step approach for knowledge discovery: 1) Create a refined enhanced graph by combining multiple relevant SGs and combining relevant knowledge from ontologies. This involves identifying relevant ontologies, reconciling different terminology, inferring new facts, and checking consistency of information in the SGs gathered from different sources; 2) Having the enhanced and refined semantic graph, we employ a semantics driven approach to detect patterns.

Specifically, we identify the relevant knowledge by: 1) Ranking of the semantic links based on the scoring of the relevant links. The ranking can be event driven. Thus, being able to prune the non-relevant structures in the *enhanced semantic graph* (ESG); 2) Identifying possible threat structures in the ranked semantic graph; 3) Identify semantic based *collusion* in the ESG.

These techniques turned into tools are a powerful means to facilitate threat vulnerability assessment and can be used to generate targeted and automated threat-related alerts for the appropriate agencies

1. Introduction

Terrorism research reveals that the techniques and methods used by terrorists are increasingly getting sophisticated. It is increasingly becoming a challenge to generate actionable knowledge to thwart terrorist events. The knowledge that exists about potential terrorist activities is typically distributed and is present in various formats with respect to representation, language, semantics, other implicit or explicit information etc. In such a distributed, heterogeneous environment the critical question then becomes as to how to perform vulnerability and threat assessment based on increasingly distributed, heterogeneous and multimedia sources of information.

Each piece of information from different sources can produce a semantic graph, which consists of nodes and relationships between the nodes. Moreover each node is qualified with some metadata about the node. Thus for instance the information about individuals from one source, such as a document, can be converted into a semantic graph, which shows the relationships between the individuals and further each individual can be qualified with other metadata. This metadata can be from the same source or can be brought in from other sources. Further, more than one such semantic graph can be combined to generate a more comprehensive set of relationships in the form of an enhanced semantic graph. Based on this enhanced semantic graph knowledge can be inferred about implicit relationships or events that may take place due to these associations or relationships between individuals. The following example demonstrates the complexities in such a domain.

***Motivating Example:** Consider a shipment carrying liquid Urea entering through the port in Los Angeles, whose final destination is Phoenix, AZ. Assume there is another shipment entering through the port in Newark carrying cyclotrimethylene trinitramine (RDX) is bound for Wintersburg, AZ. These two shipments, when viewed in isolation appear to be benign. However, a closer analysis based on spatial proximity (shipments with spatially close destinations), temporal proximity (these two events occurring close in time), and semantic proximity (the materials being shipped have some semantic relationship, for example, can be combined to make explosives), may help in discovering information useful for customs inspectors. This is a discovery of co-occurrence of events rather than causation, and identifying occurrences of rare events namely collusions between entities based on spatial, temporal and semantic proximities. There is a need to devise algorithms to identify co-occurrence of rare collusions to incorporate knowledge of domain experts to confine the analysis to meaningful intervals of time, space and semantic distances in order to facilitate efficient analysis. Here we need to first identify such interesting entities, second identify preliminary linkages between them which puts them in some form of linkages such as spatial and temporal proximity and finally establish semantic proximities between them, without being delayed by the need for consulting a domain expert for each task. Thus there is a need to perform this analysis automatically to generate such potential collusions.*

This paper addresses two major questions in this direction: Given the semantic representation in the form of a semantic graph, a) how do we enhance the knowledge embedded in the Semantic graphs by combining it with other relevant semantic graphs and knowledge sources specifically ontologies. b) How do we derive knowledge from these enhanced semantic graphs to discover hidden patterns for threat and vulnerability assessment?

We note there that although this component has several associated challenges, there is other prominent work [], which addresses these challenges. We build on these initiatives, which would feed into our system the semantic graphs derived from each source. For the purpose of demonstrating a simple example of a semantic graph we utilize semantic graph of outliers where each outlier is annotated with information about the causal dimension and the linkages between nodes are the spatio-temporal proximities between them. Thus we demonstrate the creation of semantic graphs using outliers and their causal dimensions.

Given a set of semantic graphs from a set of disparate sources and a set of ontologies, we take a two-step approach: 1) Refine and enhance a semantic graph: by combining relevant knowledge from existing ontologies.

2) Once we have the refined semantic graph: detect patterns from it by extending the graph techniques to account for semantics and taking a semantics driven approach rather than a data driven approach.

2. Semantics based Collusion set detection: Approach Overview

We begin our approach by using semantic graphs. These semantic graphs may come from various sources. For the purpose of demonstrating this we generate a preliminary semantic graph by

identifying outliers and their causal dimensions, subsequently we generate spatial and temporal linkages between these outliers. Once we have such a semantic graph(SG) we enhance this SG to form an enhanced Semantic Graph (ESG) using ontologies and reasoning. That enhancement includes removing relations that are not supported by the reasoning and score the other relations.

Once we have such an ESG with semantic scores, we identify semantics based collusion sets, using graph properties enhanced to consider semantics such as semantic centrality and semantic cliques.

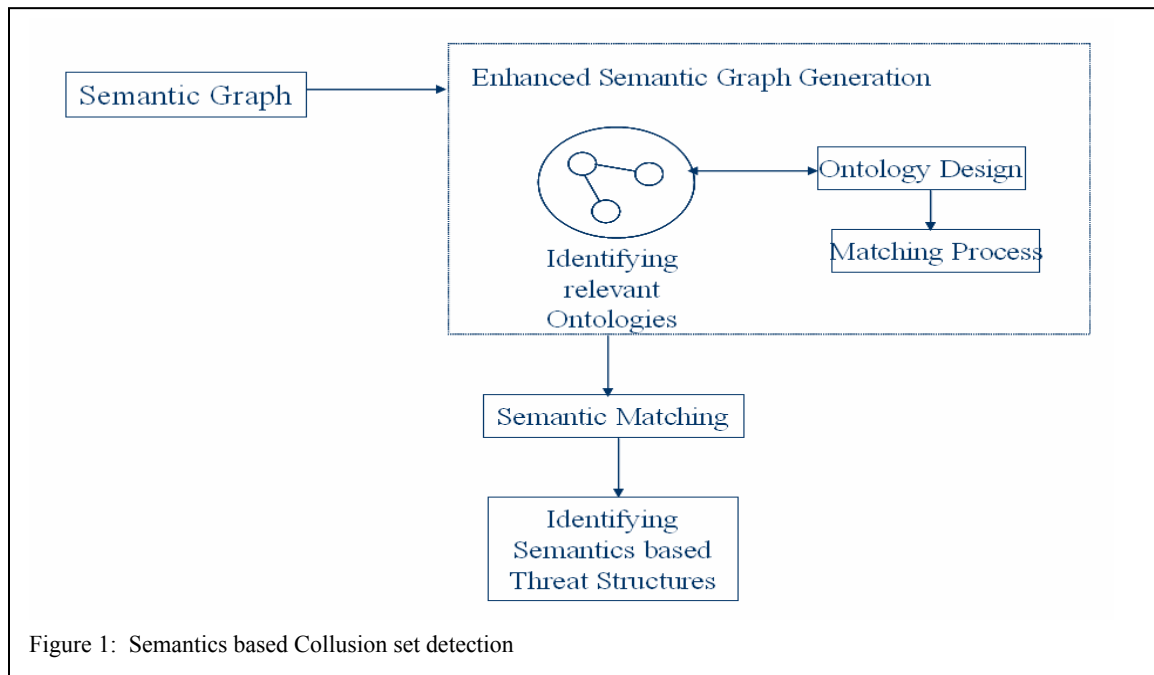


Figure 1: Semantics based Collusion set detection

3. Semantic Graph

3.1 Outlier discovery with Causal dimension

An outlier in a data set can be thought of as an observation, which is very different and inconsistent with the other observations. Many ways have been proposed to detect and subsequently predict such anomalies or outliers. Essentially outliers are considered to be those objects that do not have enough neighbors, here neighbors are defined based on distance from a given object. However most distance-based approaches [KN98, KN99, RRS00] have mainly use Euclidean distance, which poses certain limitations. Firstly, it does not provide intuitive knowledge of the outliers such as which dimension may be causing the extreme behavior. Secondly, it does not scale well for high dimensional data and is not very efficient for large data sets. Most distance-based approaches using Euclidean distance are sensitive to input parameters. [JAVA05] proposes a distance based outlier detection technique, where a new distance metric is proposed which addresses some of these limitations. We utilize this new distance metric, called the collusion distance metric. This distance metric is motivated from Hausdorff distance metric [R91]. This distance metric identifies the causal dimensions in which an object may be showing extreme behavior. For high dimensional data sets, it can identify outliers better and improves both precision and recall, when compared to the Euclidean distance based outlier detection [KN98]. These results were shown on real data sets (Prognosis Breast Cancer, Diagnostic Breast Cancer, Ionosphere and PIERS) and simulated datasets. However, collusion distance based method may

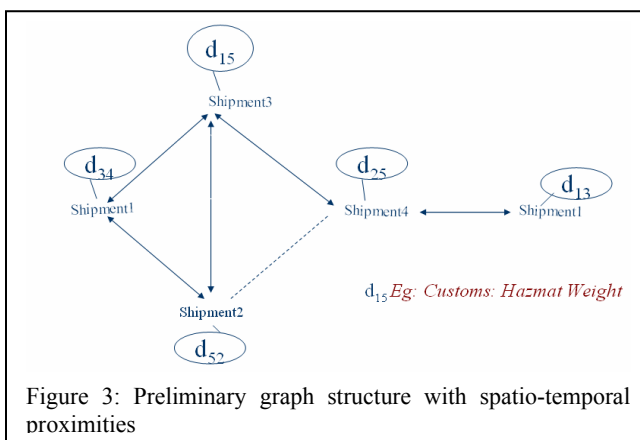
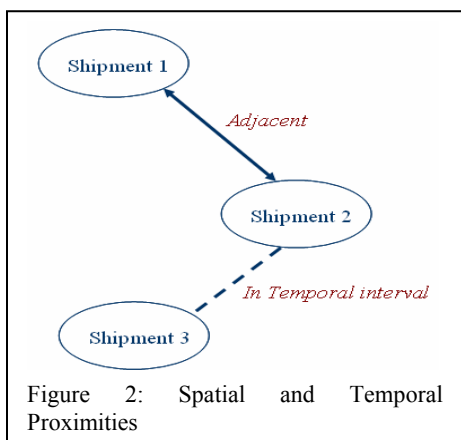
sometimes miss global outliers. Here we adapt the definition of Collusion distance metric [JAVA05].

Definition 1 [Collusion Distance]: Given two N dimensional objects $X = (x_1, \dots, x_N)$ and $Y = (y_1, \dots, y_N)$, the Collusion Distance $CD(XY) = \max |x_i - y_i|$. Along with the distance, the dimension i is identified as the causal dimension.

Essentially, given two points X and Y, it computes the maximum of the distances between each of the dimensions of X and Y. In doing so, it also identifies in which dimension(s) this maximum difference is occurring. Thus along with the distance, the dimension i is identified as the causal dimension. Thus, the *output* from this step is a set of outliers and the causal dimensions for each outlier.

3.2 Spatio Temporal Pruning

Spatial Proximity



Given a set of outliers, we would like to identify some spatial and temporal proximities (as shown in figure) between them. For *spatial proximity* a spatial neighborhood graph of outliers is formed first. A spatial neighborhood graph will comprise of nodes and edges where the nodes correspond to the outlier objects and there exists an edge between the nodes if and only if there exists a neighboring relation between the two nodes identified by certain relationships. These relationships include Topological relationships e.g.: adjacent, inside, disjoint, etc., Direction relationships e.g.: Above, below, north_of, etc., Distance relationships e.g.: “distance < 100”.

For *temporal proximity*, we identify relevant temporal intervals for example: Identifying temporal intervals based on expiration dates of chemicals. Secondly temporal pruning is also performed at the Ontological level. The output from this step is a preliminary semantic graph structure which shows the spatio-temporal linkages between the outliers and associated with each outlier is the meta data namely the causal dimensions for each outlier. We next need to identify the semantic proximities between these causal dimensions and strengthen this graph.

4. Enhanced Semantic Graph

The first step in enhancing the semantic graphs is to identify the relevant ontologies for the subsequent steps. We employ a simple keyword search where each ontology is mapped to a set of keywords. During the identification process, these keywords are obtained from the semantic

graph. In order to create an appropriate set of keywords for an ontology, the ontology is scanned for terminology upon creation and update.

After identifying the relevant ontology, the enhancement of the semantic graph is performed as semantic matching. We build the ontologies using the Ontology Web Language (OWL) and restrict it to the OWL DL (DL refers to Description Logic). Thus, we can perform the semantic matching using DL reasoning techniques. OWL DL possesses the expressive power of DLs, retains both computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time).

In this section, we first present a preliminary on Description Logics (DLs); we then follow by presenting the framework that we follow in designing the ontologies in order to support the semantic matching; and finally present the reasoning process and how is it used to enhance the semantic graph.

4.1 Description Logics

DLs are defined as a family of class-based knowledge representation formalisms that are equipped with well-defined model-theoretic semantics⁰. The basic building block of DLs is the “concept” which refers to a set of individuals. Concepts are used to represent some domain’s knowledge by building a hierarchical structure referred to as a “terminology” which provides intentional representation of the domain. On the other hand, the extensional representation is provided by the assertions that describe the individuals. The hierarchical structure of terminologies is defined by the IS-A relationship which provides the generality/specificity among concepts. The IS-A relationship provides also the basis for inheritance of properties of concepts. Nevertheless, DLs can represent other kinds of relationships that are usually called “roles”¹.

A role can be seen as a link from a concept (domain of the relationship) to another concept (range of the relationship). The objects in the range are referred to as role fillers. In DLs, roles can have value restrictions which express a limitation on the range of objects that can fill the role. Another type of restrictions that can be expressed on roles is the number restrictions that restrict the cardinality of the relationship. These types of relationships and their value restrictions and number restrictions are inherited from concepts to their subconcepts.

In addition to the atomic descriptions (i.e., the atomic concept and the atomic role), DL languages can have complex descriptions that are built inductively with concept constructors such as conjunction (denoted by \sqcap), and disjunction (denoted by \sqcup). In addition, the role restrictions in DLs can be quantified by the universal quantifier “for all” or “only” (denoted by \forall) and the existential quantifier “there exists” or “at least” (denoted by \exists).

DLs have well-defined semantics where the meaning of a DL language is specified via a model theoretic semantics whose purpose is to explicate the relationship between the language syntax and the intended model of the domain [HP-SH03]. A model consists of an interpretation function (\cdot^I) and a non-empty set of objects Δ^I that represents the domain of the interpretation. The interpretation function assigns to every individual a an interpretation $a^I \in \Delta^I$ and to every atomic concept A , a set $A^I \subseteq \Delta^I$. It also assigns to every role R , a binary relation $R^I \subseteq \Delta^I \times \Delta^I$. Based on this interpretation, an individual i is an instance of a concept C if $i^I \in C^I$ and a concept C is a subconcept of a concept D if $C^I \subseteq D^I$. If $C^I = D^I$ in every model, then C and D are said to be equivalent (denoted $C \equiv D$).

¹ In OWL terms, a role is called object property.

A DL family that provides a balance between expressiveness and computational efficiency is the *SHI* family. A member of this family is *SHIQ* (D) whose expressive power is almost equivalent to OWL. *SHIQ*(D) supports full concept negation, transitive roles, qualified cardinality restrictions, role hierarchies, inverse roles, and datatypes. The domain of interpretation of datatypes in *SHIQ* (D) (denoted by Δ^I_D) is strictly separate from the domain of interpretation of individuals and concepts Δ^I . A data type such as integer is represented as a subset of Δ^I_D and a value as the integer “13” as an element of Δ^I_D .

DLs support two types of terminology axioms: inclusions and equalities. Inclusions take the form $C \sqsubseteq D$ and equalities take the form $C \equiv D$. When the left side term of equality is an atomic concept, the equality axiom is referred to as definition. For instance, we may define the concept *WorkDay* as follows:

$$\text{WorkDay} \equiv \text{WeekDay} \sqcap \neg \text{NationalHoliday}$$

A finite set of definitions \mathcal{T} is called terminology or TBox. In order for a terminology to be consistent, it must not have cycles; that is, a concept cannot exist on both sides of the equality.

Axioms about individuals can take the forms $C(a)$ and $R(b, c)$ where the former asserts that a is an element of concept C and the later asserts that c is the filler of role R for individual b . A finite set of these assertions create an ABox.

4.2 Reasoning

An OWL ontology is similar to a DL knowledge base and consequently includes implicit knowledge that can be made explicit through reasoning $\mathbf{0}$. A DL knowledge base is comprised of a TBox and an ABox where each part has its types of reasoning. Assuming a knowledge base K , concepts C and D , and an individual a , TBox reasoning includes:

- 1- *Class subsumption queries*: given two classes C and D , determine if C is a subclass of D with respect to K .
- 2- *Class hierarchy queries*: given a class C , return all or the most-specific (most-general) superclasses (subclasses) of C in K .
- 3- *Class satisfiability queries*: given a class C , determine if C is satisfiable (consistent) with respect to K .

In addition, ABox reasoning includes $\mathbf{0}$:

- 4- *Ground*: determine whether a given individual a is an instance of C .
- 5- *Open*: determine all the individuals in K that are instances of C .
- 6- *All-classes*: given an individual a , determine all the classes in K that have element a .

4.3 Ontology Design Framework

The idea of the solution is to build an OWL ontology that captures the semantics of some domain and the asserted information from the semantic graph in concepts, properties, and restrictions, and then enhance the graph and score the links by matching the asserted information to some known dangerous sets by reasoning about them. Each ontology (Figure 4) includes two major taxonomies: the domain specific taxonomy, e.g., chemicals, and an operational taxonomy for the matching process that we refer to as the *Potentially Dangerous Combinations* (PDC). The PDC taxonomy is further divided into two sub-taxonomies: a taxonomy that represents dangerous combinations of elements that we refer to as *Known Dangerous Combinations* (KDC), and a

taxonomy for the information asserted from the semantic graph that we refer to as *Asserted Combinations*. The former taxonomy, i.e., the KDC, is provided by a domain expert.

The KDC taxonomy (Figure 5) has a root concept, KDC. The next level includes a set of concepts that represent score levels (e.g., 90%, 80%, etc). Under each score is a hierarchy of combinations (e.g., chemical elements) that belong to that score in terms of the strength of the relation. On the other hand, the asserted combinations taxonomy (Figure 6) is created as follows: upon receiving a semantic graph, a set of combinations ($C_1, C_2, \dots, C_{n-1}, C_n$) is created by extracting the relations from the graph. If a relation includes more than two nodes, multiple combinations are created incrementally starting from two nodes. Therefore, the asserted combinations represent all the possible combinations from the semantic graph.

4.4 The Matching Process

The first step of the matching process is to a parent create a concept that represents the union of all the nodes in a combination. For instance, in Figure 6, the concept C_n is the union of the dimensions d_{13} , d_{36} , and d_{64} . Since DLs assumes open world reasoning, we add a *closure axiom* to the definition of this concept and the concepts in the KDC taxonomy as well. The parent concept is then matched to the KDC taxonomy using two reasoning tasks: first, using a class hierarchy reasoning, the parent concept is matched to a combination from the KDC taxonomy where a KDC and a score are identified. If the score is zero, that is, the parent concept does not match any combination in the KDC taxonomy, the second step is skipped. In the second step, class subsumption reasoning is used to identify the type of matching between the parent concept and the KDC family. The result of the matching is one of three cases: exact match, the parent concept subsumes the KDC combination, or the KDC combination subsumes the parent concept. The latter case is taken to further processing where the score is reduced.

The enhanced semantic graph is created by eliminating the links that have score equals Zero and by adding the score to the other links in semantic graph.

4.5 Implementation of Semantic Matching

The matching mechanism (Figure 7) is composed of three modules that interact among each other and with the RACER reasoner [HM00]. The first module is the *ontology loader* which reads the ontology from an OWL file, converts it to DIG format [Bech03] using the OWL API [BLV03], requests an empty knowledge base from RACER, and submits the converted ontology as a set of *Tells*. DIG is a set of specifications for standardizing the interface to a DL reasoner. The second module, the *query loader*, receives a combination, creates a parent concept as the union of the combination, asserts the parent concept and the combination, and initiates the third module, the *query processor*. The third module, the query processor, performs the matching and returns the enhanced semantic graph.

While it is possible to create an ontology by entering its RDF/XML description in an OWL file, we found that the use of an ontology editor is indispensable. There are several ontology development tools that have been mainly developed by academia including Protégé [Prote05] by Stanford University, Swoop by University of Maryland, and OilEd by University of Manchester. We used Protégé for importing the UNSPSC ontology and for developing the rest of the taxonomies. Protégé provides several wizards and plug-ins that are very useful in saving time and checking correctness of work. Protégé can connect to any reasoner that supports the DIG interface. We used the RACER reasoner to check the consistence of the ontology during the development time. We also used RACER as part of the enforcement mechanism.

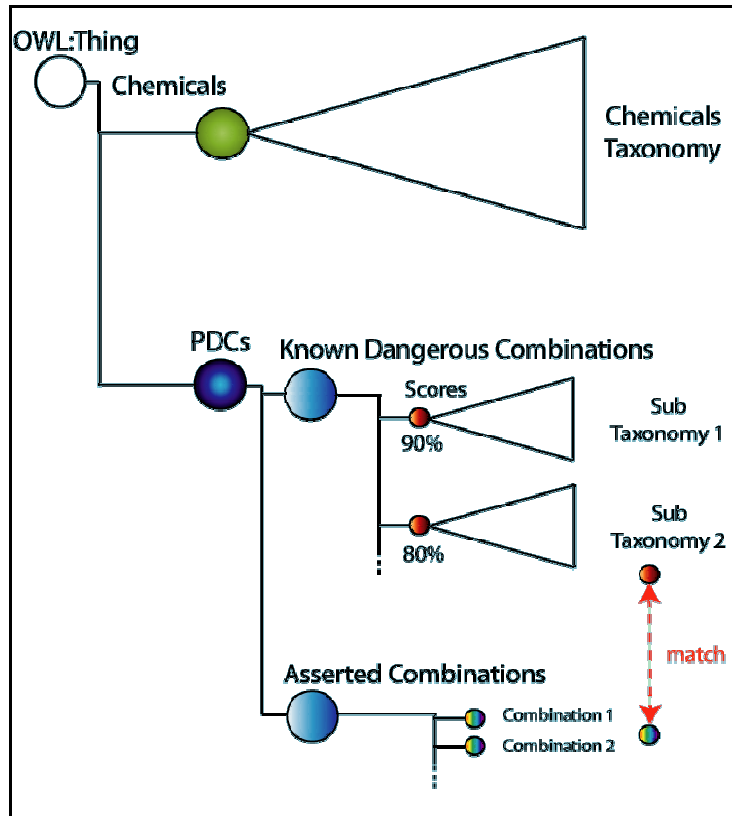


Figure 4: Design of Ontology

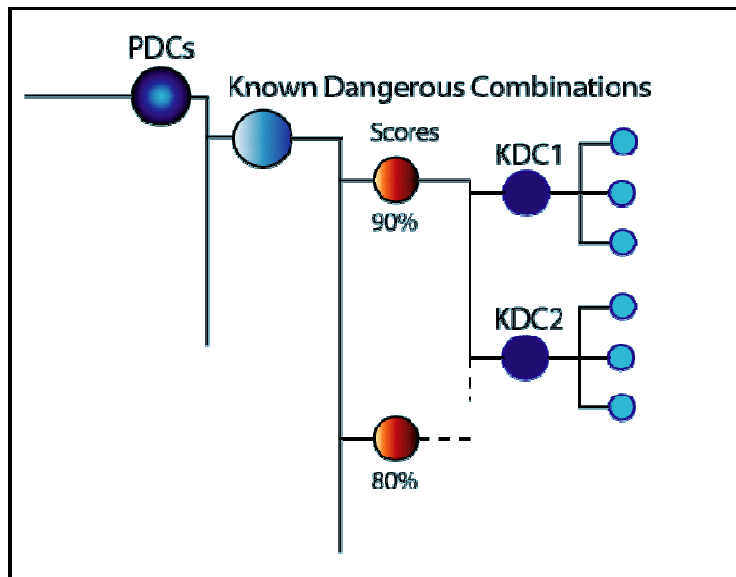


Figure 5. Known Dangerous Combinations

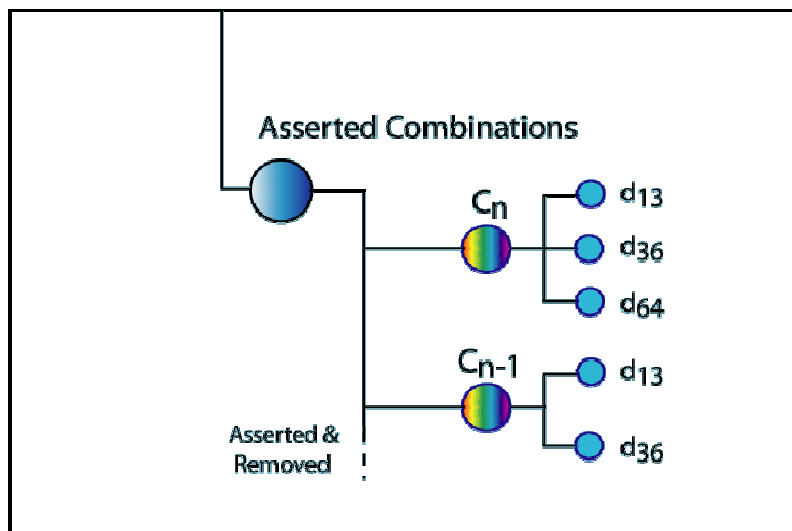


Figure 6: Asserted Combinations

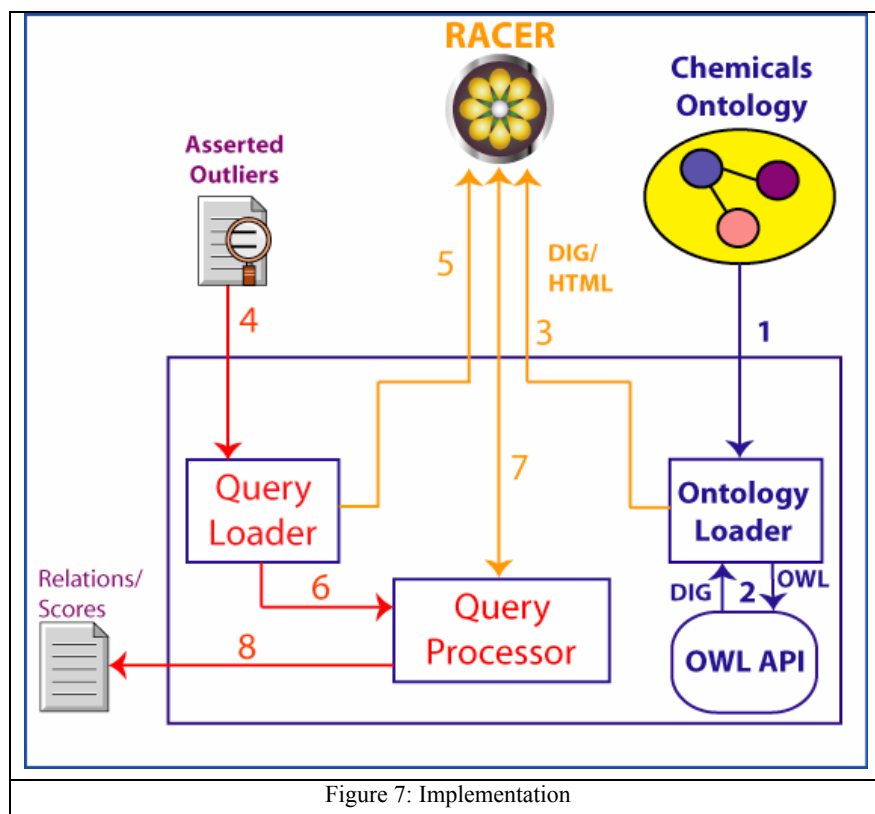
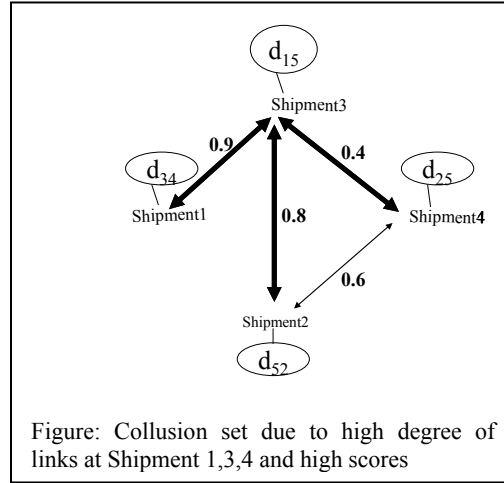
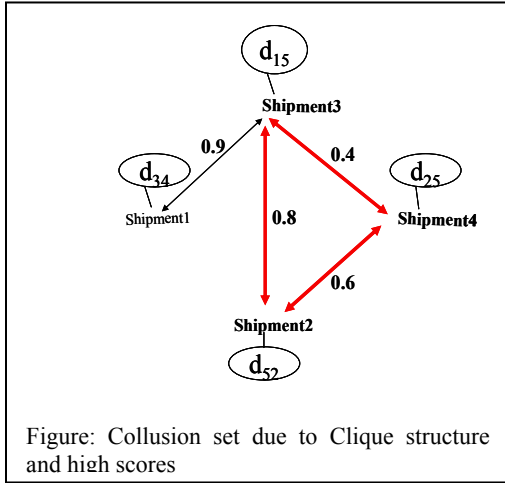


Figure 7: Implementation

5. Identifying Semantics based Potential Collision Sets



Given the enhanced semantic graph we mine for potential collision sets. These collisions could be formed due to the inherent properties of the ESG. We utilize the graph properties enhanced using semantics. We adapt the concept of semantic centrality and semantic cliques from [JAVA05B].

Definition 2 (Semantic centrality): Given an enhanced semantic graph E , the semantic centrality of node $a_i \in E$ is defined as $f [i=1 \text{ to } k] (w_{ij})$ where w_{ij} is the semantic weight of the edge between a_i and a_j , k is the number of edges incident on a_i , and $f = \{\text{sum; max; average}\}$.

Essentially, the more the value of the semantic centrality, the more semantically influential is that node in the ESG.

Definition 3 (Semantic clique): Given an enhanced semantic graph E , a semantic clique $sc = \langle c, sw \rangle$ where $c = \{a_i \dots a_k\}$ is a clique in E and $sw = f [i=1 \text{ to } k] (w_{ij})$ where w_{ij} is the semantic weight of the edge between a_i and a_j , k is the number of edges incident on a_i , and $f = \{\text{sum; max; average}\}$.

6. Related work

Outlier detection has been extensively studied in the statistical community [BL94]. One of the major limitations of these approaches is that the data is made to fit a certain distribution. The main problem due to this is that, in most cases, one does not have enough knowledge of the underlying data distribution. To address this issue, Knorr and Ng [KN98] have proposed a distance-based outlier detection approach that is both simple and intuitive, which states that a point is said to be an outlier in a dataset 'T' if no more than 'p%' of points are at or less than a threshold distance from the point.

An extension of this work [KN99] identifies intensional knowledge for outliers such as, which sets of dimensions explain the uniqueness of the outliers. Ramaswamy et al. [RSS00] have extended the approach based on the distance of a point from its 'Kth' nearest neighbor. After ranking the points by the distance to its 'Kth' nearest neighbor, the top 'K' points are identified as outliers. The concept of local outlier has been introduced in [BKNS00], where the outlier rank of a point is determined by taking into account the clustering structure in a bounded neighborhood of the point. However none of these approaches are specially designed to work for high dimensional problems. Other outlier detection approaches OPTICS-OF [MKNS99] and LOF

[BKNS00] also do not address high dimensional outlier discovery. To address the problem of dimensionality curse, Aggrawal and Yu [AY01] have proposed an approach that considers projections of the data and considers the outlier detection problem in subspace. He et al. [HE] propose a semantics based outlier detection, in which they assign class labels to data based on the semantics, and discover outliers within each group with the same class labels. While there exist a number of proposals for outlier detection, none of these approaches take the semantic relationships among dimensions into account. Moreover, none of these approaches identify the causal dimensions or other causal knowledge of the outliers, which is essential in addressing the problem being tackled in this paper.

The work in the area of social networks and link analysis is relevant to the problem considered in this paper. Social network analysis involves the study of prominent patterns within social networks, tracing the flow of information and resources, effect of relationships on various entities, etc [B72]. Xu et al. [XUC03] focus on detecting and specifying changes in criminal organizations using descriptive measures from social network analysis. Wang et al. [W04] address identification of record linkages using string comparators to link different deceptive criminal identity records. One key aspect of creating social networks is the formulation of ties. Various methods have been proposed for the sampling of ties such as full network, Snowball, Ego centric, Ego only methods [WF94]. Snowball method is of specific interest here because it is useful in detecting ties in the more impenetrable social groupings where the entities are few in number and members of such population can be considered deviant. This technique is basically used to identify entities, which are then used to refer to other entities, thus snowballing into a network of connectivities. This technique works well in sampling ties from people; however, this type of information cannot be gleaned from large data sets. Therefore, the results obtained through these approaches can be complimentary to those obtained using data mining approaches.

Kubica et al. [JMCS03] propose a graph-based approach for link analysis and collaboration queries. They propose a Bayesian network based technique to identify underlying associations in a social network graph. They account for different types of links, varying in frequency and extent of noise. They also incorporate temporal information pertaining to the link. It learns the underlying graph using weighted counts of co-occurrences to approximate the edge weights, which can be computed from counts gathered during a single scan of the data. The weighting function such as temporal weighting assumes that the recent links will be more indicative of the current graph. However, in many cases the anomalous objects may not co-occur and the only recent links may not provide the most intuitive knowledge. Most importantly, the links may originate from multiple data sources and the weighting functions would need to incorporate such knowledge dynamically, which are not addressed under this approach.

7. Conclusion and Future work

8. References

- [AY01] Charu C. Aggarwal and Philip S. Yu. Outlier detection for high dimensional data. In Proceedings of the 2001 ACM SIGMOD international conference on Management of data, pages 37{46. ACM Press, 2001.
- [BL94] Vic Barnett and Toby Lewis. Outliers in Statistical Data. John Wiley and Sons, 3rd edition, 1994.
- [BKNS00] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jorg Sander. Lof: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD, 2000.
- [KN98] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In Proceedings of the International Conference on Very Large Data Bases (VLDB 1998), pages 392{403, August 1998.

- [KN99] Edwin M. Knorr and Raymond T. Ng. Finding intensional knowledge of distancebased outliers. In Proceedings of 25th International Conference on Very Large Data Bases, pages 211-222, 1999.
- [KMCS03] Jeremy Kubica, Andrew Moore, David Cohn, and J. Schneider. Finding underlying connections: A fast graph-based method for link analysis and collaboration queries. In Tom Fawcett and Nina Mishra, editors, Proceedings of the 2003 International Conference on Machine Learning, pages 392-399. AAAI Press, 2003.
- [RRS00] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In Proceedings of the ACM SIGMOD, pages 427-438, 2000.
- [R01] G. Rote. Computing the minimum hausdorff distance between two point sets on a line under translation. *Inf. Process. Lett.*, 38(3):123-127, 1991.
- [JAVA05] V.P.Janeja, V.Atluri, J.S.Vaidya, and N.Adam. Collusion set detection through outlier discovery. In *IEEE Intelligence and Security Informatics*, 2005.
- [JAVA05B] V.P.Janeja, V.Atluri, J.S.Vaidya, and N.Adam. "Semantics based discovery of Affiliation Network" Technical report, 2005.
- [GHVD03] B. N. Grosz, I. Horrocks, R. Volz, and S. Decker, "Description Logic Programs: Combining Logic Programs with Description Logic", In Proceedings of the Twelfth International World Wide Web Conference, Budapest, Hungary, May 2003.
- [BCMNP-S03] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, "The Description Logics Handbook: Theory, Implementation and Applications", Cambridge University Press, UK, 2003.
- [HP-SH03] I. Horrocks, P. F. Patel-Schneider, and F. Z. van Harmelen, "From SHIQ and RDF to OWL: The Making of a Web Ontology Language", In *Journal of Web Semantics*, 1(1), December 2003.
- [SWM04] M. K. Smith, C. Welty, and D. L. McGuinness, Editors, "OWL Web Ontology Language Guide" A W3C Recommendation, Latest version available at <http://www.w3.org/TR/owl-guide/>, February 2004.
- [Bech03] S. Bechhofer, "The DIG Description Logic Interface: DIG/1.1", Available from <http://dl-web.man.ac.uk/dig/2003/02/interface.pdf>, 2003.
- [BLV03] S. Bechhofer, P. Lord, R. Volz, "Cooking the Semantic Web with the OWL API", In Proceedings of the 2nd International Semantic Web Conference, ISWC, Sanibel Island, Florida, October 2003.
- [Prote05] Stanford University, "The Protege Project", Available from <http://protege.stanford.edu>, 2005.
- [HM00] V. Haarslev, R. Moller, "RACER Users's Guide and Reference Manual", Available at <http://www.cse.concordia.ca/~Ehaarslev/racer/racer-manual-1-7-19.pdf>, 2000.