Mining Common Semantic Patterns from Descriptions of Failure Knowledge

Weisen Guo **GWS@CB.K.U-TOKYO.AC.JP**
Science Integration Program (Human), Department of Frontier Sciences and Science Integration, Division of Project Coordination, The University of Tokyo, 5-1-5 Kashiwa-No-Ha, Kashiwa-Shi, Chiba-Ken 277-8568, Japan
Steven B. Kraines **SK@CB.K.U-TOKYO.AC.JP**
Science Integration Program (Human), Department of Frontier Sciences and Science Integration, Division of Project Coordination, The University of Tokyo, 5-1-5 Kashiwa-No-Ha, Kashiwa-Shi, Chiba-Ken 277-8568, Japan
Keywords: semantic description, semantic pattern, semantic graph mining, ontology, description logic, reasoning

Abstract

The EKOSS system has been developed to let knowledge experts create computer-interpretable semantic descriptions based on description logics ontologies to describe their knowledge resources. A method for using graph mining to find common semantic patterns in the set of semantic descriptions is introduced. Special characteristics of semantic descriptions that make them more complicated than the labeled graph are analyzed, and a semantic graph mining algorithm that uses a description logic reasoner to check the common semantic patterns is described. Application of the semantic graph mining approach to the set of 203 semantic descriptions of engineering failures is presented.

1. Introduction

The Expert Knowledge Ontology-based Semantic Search (EKOSS – www.ekoss.org) system has been developed to support a form of computer-mediated sharing, discovery, and integration of expert knowledge not possible with simple key word indexing (Kraines et al., 2006). In EKOSS, each knowledge resource is represented by a computer-interpretable semantic description using a domain ontology grounded in a description logic (DL) (Baader et al., 2003). A semantic description is formulated as a DL assertion component or ABox populated by instances of the classes that are specified in the DL terminological component or TBox provided by the ontology. The DL ABox can be represented as a graph, where the instances form the graph vertices and the properties form the graph edges.

Common knowledge, such as general or reoccurring ideas, can be elucidated from a set of knowledge resources, such as a knowledge base of case histories of engineering failures, by mining the corresponding set of semantic descriptions to find common semantic patterns meeting a prescribed minimum support of occurrence. One important problem is that a DL ABox is more complicated than the normal labeled graph. For example, properties connecting instances in DL ABoxes can be transitive, symmetric, and/or inverses or subproperties of other properties. Existing graph mining algorithms cannot address these characteristics (Section 2).

This paper presents a new approach using a DL reasoner to solve these issues. In section 2 we review the existing graph mining algorithms that are related to the task we have framed. In section 3, we discuss the salient properties of semantic graph mining and give definitions of semantic descriptions and common semantic patterns in terms of graph mining. In section 4, we present an algorithm to mine a set of semantic descriptions. In section 5, we describe an application of this approach to semantic descriptions of engineering failures. Section 6 concludes the paper.

2. Related Work

Research in Knowledge Discovery and Data Mining has led to development of several algorithms that can find characteristic patterns and generalized knowledge from large sets of structured data, such as transaction data, sequences, vectors, time-series, geographical data, multi-relational data, graphs, and trees, and even semi-structured or unstructured data. Rajaraman describes a method for mining a kind of semantic networks for knowledge discovery from text that uses a concept frame graph (CFG) to represent a concept in the text (2003). First they construct CFG's from a set of documents. Then, they mine the CFG's to get frequent CFG's. However, this method does not use the *a priori* semantic structure of concept labels, and each CFG is a simple semantic network with one center concept and some other related concepts. The mining process extracts frequent concepts, not frequent semantic subgraphs.

Appearing in the 6th International Workshop on Mining and Learning with Graphs, Helsinki, Finland, 2008.

Inokuchi, Washio, and Motoda developed the AGM algorithm to mine frequent patterns from graphs (2000). This algorithm derives all frequent induced subgraphs from both directed and undirected graph structured data. The graphs can have loops (including self-loops) and labeled vertices and edges. An induced subgraph can be a connected or unconnected graph. Since in many applications, such as the one that we are studying, the patterns of interest are connected graphs, Inokuchi developed an extension of AGM called AcGM (2002). AcGM uses algebraic representations of graphs that make possible operations and well-organized constraints to limit the search space efficiently. AcGM can mine generalized patterns where vertices and/or edges have labels at any level of a taxonomy by extending the definition of "subgraph." However, the extended method outputs a massive set of patterns, most of which are over-generalized, which causes computation explosion.

In further work, Inokuchi presented an efficient method to discover all frequent patterns which are not over-generalized from labeled graphs when taxonomies on vertex and edge labels are available (2004). This method can mine the labeled graph data with taxonomies. However, there are important differences between a taxonomy and a DL ontology. To our knowledge, mining common semantic subgraphs from a given DL ontology and a set of semantic graphs representing ABoxes has not yet been addressed in the field of graph mining.

Mooney et al. (2002) applied inductive logic programming (ILP) to relational data for link discovery. ILP is the study of learning methods for data and rules that are represented in a logic such as first-order predicate logic. Given background knowledge and a set of positive and negative examples, ILP can infer a hypothesis in the form of a rule such as *daughter(X, Y) <= female(X), parent(Y, X)*. In our work, knowledge is represented in a set of semantic descriptions, and DL reasoning is used to determine what common semantic patterns appear in the set. Instead of background knowledge, we use a DL ontology, and instead of positive and negative examples, we use a set of semantic descriptions. While the goal of ILP is to define the target relation hypothesis, our goal is to find common semantic patterns that exist in the given set of semantic descriptions.

3. Semantic Graph Mining

A semantic description describes a knowledge resource in a computer-interpretable way using a DL ontology. A semantic description can be represented as a graph whose vertices are instances of classes in the ontology, and whose edges are properties that indicate specific semantic relations between the instances. This graph is similar to a labeled graph; however, the pre-existing structure behind a semantic description is different from that of a labeled graph: one is an ontology and the other is a taxonomy. A taxonomy is a classification of kinds of things into a tree, with only one relation operator between entities, the subsumption relation. DL ontologies use several relation types in addition to subsumption, such as partitive, participatory, and locational relations. A labeled graph can have two taxonomies: one for vertex labels that corresponds to the classes of the ontology and another for edge labels that corresponds to the properties of the ontology. However, a DL ontology is more than just the sum of a taxonomy of classes and a taxonomy of properties because classes can be used to specify characteristics of properties and properties can be used to specify characteristics of classes as described next.

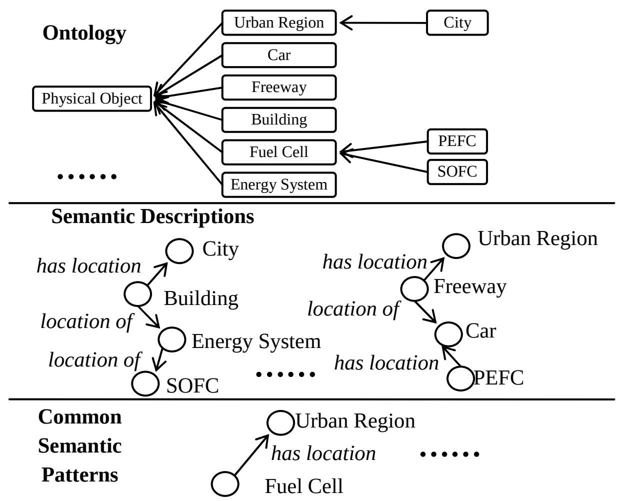

Figure 1. Top: subsumption relationships between ten classes in an ontology. Middle: two semantic descriptions to be mined. At the left is a Building that has location City. The Building is location of an Energy System that is location of a SOFC. At the right is a Car that has location Freeway that has location Urban Region. The Car is location of a PEFC. Bottom: a common semantic pattern showing that a Fuel Cell has location some Urban Region. The edges in the middle and bottom parts are object properties from the ontology whose property names are given in italic font. The vertices are instances of classes from the ontology whose class names are given in plain font. In the ontology, the object property "location of" is defined to be the inverse of "has location."

In our work, we use OWL-DL, which supports the *SHION (d)* DL, to represent the domain ontology. In this DL, object properties, which are relations between instances of two classes, can be defined with domain and range restrictions that must be fulfilled by the classes of the domain and range instances, respectively. Object properties can also be specified as transitive, symmetric, functional, and inverse functional, and they can be related to other properties through subsumption or the inverseOf relation. Properties can be used to specify classes through universal, existential, and cardinal restrictions on the usage of a particular property with the class.

We define semantic graph mining to mean mining common semantic patterns that satisfy the specified minimum support from a set of semantic descriptions by reasoning over a given ontology. A common semantic pattern is a semantic description that forms a subgraph matching with enough of the semantic descriptions in a given set to meet the given minimum support threshold. A semantic pattern matches a semantic description if the semantics specified in the pattern is included in the description. We can determine if a semantic pattern matches a specific semantic description by converting the semantic pattern to a semantic query and evaluating the query against the description using a reasoner, thereby elucidating embedded relations in addition to the explicit relations in the description graph. As a result, one difference between a common semantic pattern and a structural subgraph is that a structural subgraph must appear in the supporting graph but a common semantic pattern may not. This is because semantic pattern matching makes use of implicit embedded relationships both in the query and the matching description. We use the RacerPro DL reasoner to infer if a semantic pattern matches a semantic description (http://www.racer-systems.com).

Figure 1 shows a common semantic pattern found by semantic graph mining that does not appear in the two semantic descriptions at the structural level but matches with the two semantic descriptions in the semantic level.

4. Algorithm

We present an algorithm to handle the added complexity in semantic graph mining. First we select a set of classes and a set of properties from the ontology that are sufficiently general for the goals of the mining task and that appear at least once in the set of semantic descriptions. Second, we create element semantic patterns containing two vertices and one edge using all allowable combinations of the candidate classes and properties, subject to the domain-range restrictions on properties and the universal restrictions on classes. Third, we match the element semantic patterns with all of the semantic descriptions using the reasoner to get the set of common element semantic patterns. Fourth, we use the common element semantic patterns to create new level candidate semantic patterns, which are then matched with the set of semantic descriptions using the reasoner to get the new level common semantic patterns. The last step is repeated until no more common semantic patterns are found.

Each new level candidate semantic pattern is created based on the previous level common semantic patterns and the common element semantic patterns as follows. For each previous level common pattern, we try to add a new instance by finding a property that can connect that instance to one of the existing instances. We then add all combinations of additional allowable properties between the new instance and all combinations of the existing instances. This step must be done in reference to the DL structure of the ontology because the effects of restrictions like cardinality are not immediately obvious from the individual characteristics of properties and classes. Each allowable combination is a candidate pattern that must be evaluated by the reasoning engine. However, there may be some isomorphic patterns in the candidate patterns generated for a level, so we filter the redundant candidate patterns before matching them. The overall algorithm is summarized in Algorithm 1.

Algorithm 1 semantic graph mining

```
Input: descriptions ds, ontology o, minimum support m
  classes = set of selected upper classes from o in ds
  properties = set of selected upper properties from o in ds
  elements = set of all allowable triples (c1, p, c2) for c1
       and c2 in classes and p in properties
  Initialize commons_1 = empty.
  Initialize count = 0.
  for each e in elements
     for each d in ds
        if e matched d then
           count = count+1
     if count >= m * ds.size then
        commons_1 = commons_1 + e
  Initialize t = 2.
  while commons_{t-1}.size > 0 then
     Initialize candidates_t = empty.
     Initialize commons_t = empty.
     for each c_{t-1} in commons_{t-1}
        ps = createNewPatterns (c_{t-1}, common_1, o)
        if ps.size > 0 then
           candidates_t = candidates_t + ps
     candidates_t = Filter(candidates_t)
     Initialize count = 0.
     for each p in candidates_t
        for each d in ds
           if p matched d then
              count = count + 1
        if count >= m*ds.size then
           commons_t = commons_t + p
     t = t + 1
```

5. Application

We have applied our algorithm to a set of semantic descriptions developed for cases in the Shippai-Chishiki Database Project (http://shippai.jst.go.jp/en/Search). The Shippai-Chishiki Database contains descriptions of major accidents and failures in the fields of mechanical engineering, material science, chemical engineering and civil engineering. Reusing common knowledge elucidated from the failure cases can help to prevent the failures from reoccurring (Kraines et al., submitted). In order to elucidate common knowledge from the failure cases, we applied the approach described in this paper. First, we created semantic descriptions for 203 cases in the database using the SCINTENG ontology (http://157.82.238.34/ontologies/scinteng20060901.owl). There are 1264 classes, 236 properties and 567 restrictions in the SCINTENG ontology. We then mined the semantic descriptions using a minimum support of 15%. We found 38 candidate classes, 16 candidate properties, 96 common element semantic patterns, and common semantic patterns up to level 7 (8 instances). The number of common patterns for levels 2 to 7 were 367, 851, 1289, 945, 337, and 59. Figure 2 shows a 5th level common semantic pattern that corresponds to the natural language statement "an artificial activity characterized by a human failure ends in an event marking the start of a subsequent natural activity that ends in an event involving a specific physical object." This common pattern matches with 46 semantic descriptions in 203 semantic descriptions (failure cases).

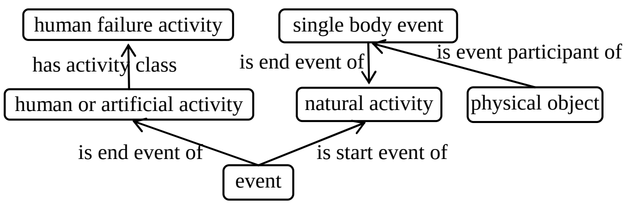

Figure 2. An example of a 5th level common semantic pattern mined from the Shippai-Chishiki Database Project.

Table 1. The pattern classes and matching description instances.

PATTERN	DESCRIPTION 1	DESCRIPTION 2
physical object	passengers and crew	workers
single body event	fatal accident	fatal accident
natural activity	falling of bus and passenger car	high temperature vapor eruption
event	break and fall	destruction of pipe
human or artificial activity	welding	not measuring the thickness
human failure activity	cutting corners	disregard of procedure

Table 1 shows how the common pattern matches with two semantic descriptions. The first description states that "a welding activity characterized by cutting corners ends in the break and fall of a bridge marking the start of a bus and passenger car falling and resulting in fatal accident involving the passengers and crew." The second description states that "an activity of not measuring the thickness characterized by disregard of procedure ends in destruction of a pipe marking the start of high temperature vapor eruption resulting in a fatal accident involving workers." The two descriptions describe situations that appear to be completely different. But table 2 shows that they share the same basic chain of activities and events as represented by the common pattern in figure 2.

6. Conclusions

Semantic descriptions created in the EKOSS system can be expressed as graphs, to which graph mining techniques can be applied as a method for knowledge discovery. However, an EKOSS semantic description has many characteristics that make it more complicated than a labeled graph. We have shown that common semantic patterns can be mined from a set of semantic descriptions through a graph mining approach using a description logic reasoner. The approach was applied to a set of 203 semantic descriptions created for the Shippai-Chishiki Database Project, and 59 common semantic patterns having as many as 8 vertices were found. Inspection of one of the common semantic patterns found demonstrates the potential of the approach for elucidating useful common knowledge from dissimilar case descriptions.

References

Baader, F., Calvanese, D., McGuinness, D., Nardi, D. and Schneider, P.P. (2003). The Description Logic Handbook: Theory, implementation and applications. *Cambridge University Press.*

Inokuchi, A., Washio, T., and Motoda, H. (2000). An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. *Proc. of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 13-23.

Inokuchi, A., Washio, T., and Nishimura, Y. (2002). *A Fast Algorithm for Mining Frequent Connected Subgraphs.* (IBM Research Report, RT0448 Feb., 2002).

Inokuchi, A. (2004). Mining Generalized Substructures from a Set of Labeled Graphs. *Proc. of the 4th IEEE International Conference on Data Mining*, 415-418.

Kraines, S.B., Guo, W., Kemper, B.E., and Nakamura, Y. (2006). EKOSS: A knowledge-user centered approach to knowledge sharing, discovery, and integration on the Semantic Web. *Lecture Notes in Computer Science (LNCS) 4273*, 833-846.

Kraines, S.B., Guo, W., Kemper, B.E., and Nakamura, Y. (2008). Reusing failure knowledge through EKOSS, the Expert Knowledge Ontology-based Semantic Search System. *7th International Semantic Web Conference,* Karlsruhe, Germany. Submitted.

Mooney, J.R., Melvile, P., Tang, L.R., Shavlik, J., Castro Dutra, I., Page, D., and Costa, V.S. (2002). Relational Data Mining with Inductive Logic Programming for Link Discovery. *Proc. of the National Science Foundation Workshop on Next Generation Data Mining*, Nov, 2002, Baltimore, MD.

Rajaraman, K., and Tan, A. H. (2003). Mining Semantic Networks for Knowledge Discovery. *Proc. of the Third IEEE International Conference on Data Mining.*