A GRAPH ANALYTICS FRAMEWORK FOR KNOWLEDGE DISCOVERY

A Dissertation
IN
Computer Science
and
Telecommunications and Computer Networking

Presented to the Faculty of the University
of Missouri–Kansas City in partial fulfillment of
the requirements for the degree

DOCTOR OF PHILOSOPHY

by
FEICHEN SHEN

M. S., University of Missouri - Kansas City, Missouri, USA, 2012
B. S., Nanjing University, Jiangsu, China, 2010

Kansas City, Missouri
2016

A GRAPH ANALYTICS FRAMEWORK FOR KNOWLEDGE DISCOVERY

Feichen Shen, Candidate for the Doctor of Philosophy Degree

University of Missouri–Kansas City, 2016

ABSTRACT

In the current data movement, numerous efforts have been made to convert and normalize a large number of traditionally structured and unstructured data to semi-structured data (e.g., RDF, OWL). With the increasing number of semi-structured data coming into the big data community, data integration and knowledge discovery from heterogeneous domains become important research problems. In the application level, detection of related concepts among ontologies shows a huge potential to do knowledge discovery with big data. In RDF graph, concepts represent entities and predicates indicate properties that connect different entities. It is more crucial to figure out how different concepts are related within a single ontology or across multiple ontologies by analyzing predicates in different knowledge bases. However, the world today is one of information explosion, and it is extremely difficult for researchers to find existing or potential predicates to perform linking among cross domains concepts without any support from schema pattern analysis. Therefore, there is a need for a mechanism to do predicate oriented pattern

analysis to partition heterogeneous ontologies into closer small topics and generate query to discover cross domains knowledge from each topic. In this work, we present such a model that conducts predicate oriented pattern analysis based on their close relationship and generates a similarity matrix. Based on this similarity matrix, we apply an innovative unsupervised learning algorithm to partition large data sets into smaller and closer topics that generate meaningful queries to fully discover knowledge over a set of interlinked data sources.

In this dissertation, we present a graph analytics framework that aims at providing semantic methods for analysis and pattern discovery from graph data with cross domains. Our contributions can be summarized as follows:

- The definition of predicate oriented neighborhood measures to determine the neighborhood relationships among different RDF predicates of linked data across domains;

- The design of the global and local optimization of clustering and retrieval algorithms to maximize the knowledge discovery from large linked data: i) top-down clustering, called the Hierarchical Predicate oriented K-means Clustering; ii) bottom-up clustering, called the Predicate oriented Hierarchical Agglomerative Clustering; iii) automatic topic discovery and query generation, context aware topic path finding for a given source and target pair;

- The implementation of an interactive tool and endpoints for knowledge discovery and visualization from integrated query design and query processing for cross domains;

- Experimental evaluations conducted to validate proposed methodologies of the framework using DBpedia, YAGO, and Bio2RDF datasets and comparison of the proposed methods with existing graph partition methods and topic discovery methods.

In this dissertation, we propose a framework called the GraphKDD. The GraphKDD is able to analyze and quantify close relationship among predicates based on Predicate Oriented Neighbor Pattern (PONP). Based on PONP, the GraphKDD conducts a Hierarchical Predicate oriented K-Means clustering (HPKM) algorithm and a Predicate oriented Hierarchical Agglomerative clustering (PHAL) algorithm to partition graphs into semantically related sub-graphs. In addition, in application level, the GraphKDD is capable of generating query dynamically from topic discovery results and testing reachability between source target nodes. We validate the proposed GraphKDD framework through comprehensive evaluations using DBPedia, Yago and Bio2RDF datasets.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled “A Graph Analytics Framework for Knowledge Discovery,” presented by Feichen Shen, candidate for the Doctor of Philosophy degree, and hereby certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Yugyung Lee, Ph.D., Committee Chair
Department of Computer Science & Electrical Engineering

Zhiqiang Chen, Ph.D.
Department of Civil & Mechanical Engineering

Baek-Young Choi, Ph.D.
Department of Computer Science & Electrical Engineering

Praveen Rao, Ph.D.
Department of Computer Science & Electrical Engineering

Cui Tao, Ph.D.
School of Biomedical Informatics
The University of Texas Health Science Center at Houston

Yuji Zhang, Ph.D.
Epidemiology & Public Health
University of Maryland School of Medicine

Yongjie Zheng, Ph.D.
Department of Computer Science & Electrical Engineering

CONTENTS

Chapter

ILLUSTRATIONS

TABLES

Tables Page

ACKNOWLEDGEMENTS

First of all, I'd like to give sincere thanks to my research advisor, Dr. Yugyung Lee. Dr. Yugyung Lee gave me the best guidance in doing my research and in helping me to solve problems as well as overcome difficulties. She always encouraged me to pursue the highest quality of research work in terms of writing paper and experimental evaluations.

I also want to thank my other Ph.D. committee members Drs. Baek-Young Choi, Cui Tao, Praveen Rao, Yongjie Zheng, Yuji Zhang and Zhiqiang Chen. As a co-displine committee member, Dr. Choi gave me a good insight in the network area and made me think more in many different aspects. Dr. Tao was my previous supervisor at the Mayo Clinic when I did my 2012 summer Ph.D. internship. She trained me wonderfully in ontology management and data normalization. Dr. Rao's team and our group shared the same lab room, which increased the opportunity of having conversations with him. His intelligence and valuable feedback always helped me in research, job searching, and my personal life. Dr. Zheng and I collaborated on a class entitled "Software Method and Tools". At that time, I was his teaching assistant. During his class, I learned how to become an outstanding lecturer and how to make students interested in the class. On research sites, Dr. Zheng's word always inspired me: "doing research is just like climbing a circling mountain. Sometimes we are facing sunshine, but most of the time we are facing darkness. When we are facing darkness, if we don't move forward, we can never see the sunshine again. But if we keep moving, we will definitely reach the top of the mountain."

Dr. Yuji Zhang was also my previous supervisor from the Mayo Clinic when I did my 2013 summer internship there. She trained me how to use CytoScape, a visualization tool that is commonly used in the biomedical domain. With the help of CytoScape, I was able to generate all visualizations throughout my research. Dr. Chen and I collaborated on mobile cloud computing research in the past. His dedication and great insight influenced me tremendously. I'd like to give thanks again to all the committee members for their support and help throughout the formation of my dissertation and academic endeavors.

Moreover, I'd like to give thanks to the UMKC School of Graduate Studies (SGS) for awarding me the research grant to support my academia research from 2014 to 2015 and training me as a leader in the Graduate Student Professional Development Program. I also want to thank School of Computing and Engineering (SCE) for supporting me with Research/Teaching Assistant Scholarship, Travel Grant and giving me the Outstanding Doctoral Student Award in 2014.

In addition, I want to give thanks to all members of UMKC Distributed Intelligent Computing (UDIC) Group for their help on my study and research.

Finally, I would like to thank my family. My wonderful wife, You Wan has always given me support during my Ph.D. work. My parents gave me not only financial support but they also encouraged my spirit even though they live in China. I would like to thank each of them for making this possible.

CHAPTER 1

INTRODUCTION

In this chapter, we first give a basic overview of current approaches and solutions on big graph knowledge discovery and analysis. We then elaborate the motivation for developing the GraphKDD framework and the contributions we have made.

1.1 Problem Statement

Today, the main challenge we are facing in knowledge discovery research is the big data problem associated with large, complex, and dynamic variations of formats. There is no capacity to carry out analysis of these datasets, because we do not have the appropriate tools and computational infrastructure that can be fully understood and utilized by involved personnel. As the demand for the integration and analysis of such data has been growing steadily, the first effort toward connecting scattered data materialized as a data movement by a different community, i.e., the Linked Open Data (LOD) [8].

In order to extract a cohesive structure and semantics, it is essential to know what information exists and what significant relationships are among the related domains. The Semantic Web is able to provide a platform of information exchanges for different knowledge bases. Increasingly, we are also seeing the emergence of cross domains among different datasets. Especially, in the biomedical informatics domain, data normalization plays an important role to integrate heterogeneous resources for further analysis (i.e., Bio2RDF [10], OBO [112], LinkedCT [50]). For example, the Semantic Web Health Care and Life Sciences Interest Group (HCLSIG [23]) was formed to "improve collaboration, research and development, and innovation in the information ecosystem of the health care and life science domains using Semantic Web technologies." Under this drive, the large amounts of data have been specified and shared via machine-readable formats, such as a Resource Description Framework (RDF) [66] and Ontology Web Language (OWL) [9]. The ontologies are developed to easily extend the work of others and share across different domains. The Semantic Web technologies make it easier and more practical to integrate, query, and analyze the full scale of relevant data from various domains.

1.2 Motivation

To make seamless interoperability and interchanges among heterogeneous datasets, significant difficulties still exist. There are some existing promising semantic approaches for linking different datasets; however, they are computationally expensive and impractical for large scale ontologies since these works may still require human intervention. Furthermore, as the size of data increases drastically, it is difficult to discover information from structured/unstructured data in a single domain or cross domains, especially for those researchers with expertise in a specific domain. Thus, we need to reduce human intervention with the help of process automation in the extraction and integration of semantics from structured or unstructured data.

For extraction of a cohesive structure and semantics from structured or unstructured data, identification of meaningful linking, either together within or across a large number of ontologies, is necessary. Especially in biomedical informatics domains, vSparQL [105] was introduced to enable application ontologies to be derived from these large, fragmented sources such as the FMA [104]. A series of queries might be generated using large ontologies like the NCI thesaurus by extracting relevant information that is desired for applications [83]. The GLEEN project aims to develop a useful service for simplified, materialized views of complex ontologies [30]. However, these works are limited due to the lack of the comprehensive semantic analysis of large sources and the usage of the knowledge for query processing. We need to connect related information through a reference ontology that becomes a platform to link together multiple ontologies that cover a broad range of related information. Advanced techniques are needed to analyze these larger reference ontologies, rather than simply getting a slice of a reference ontology and applying it for a query process or decision support [83]. There is also some related work on using K-Means and Fuzzy C-Means for clustering microarray data [114] [29], but neither of them are concerned about the semantics of data nor hierarchical clustering.

Today is the world of information explosion. With the increase of research in big data, more and more datasets from different domains have been added to the existing LOD (e.g., DBPedia [12]), which makes highly complex relationships and condensed interlinks among the large number of these knowledge bases. To some extent, the speed of data growing in terms of multiple domains is much faster than that of the large amount of knowledge people can acquire and consume in their daily lives. In other words, since different datasets are physically grouped instead of semantically clustered, it is extremely difficult for people without expertise to extract knowledge from various domains nowadays. Therefore, there is a big gap between human limited knowledge and the large amount of knowledge that can be discovered from this huge amount of data.

1.3 Contribution

In this work, we apply a RDF predicate oriented pattern analysis methodology and combine the advantages of Machine Learning with the added rigor of machine-readable semantics in extracting information and generating queries applicable for knowledge discovery. A pattern based predicate oriented similarity measurement gives a close relationship among predicates and generates a similarity matrix. An unsupervised learning algorithm works on the similarity matrix to build smaller topics that hold the closest domain and knowledge inside. In addition, queries are evaluated by measuring the content of information, identifying possible extensions or compositions of queries and making a comparison with an existing query benchmark. We develop a prototype of the GraphKDD system and evaluate the proposed query model based on predicate oriented clustering with different domains of datasets. More specifically, the contributions of this work are as follows:

- A Predicate Oriented Neighborhood Patterns (PONP) analysis model to quantify the close relationship among different RDF predicates with cross domains knowledge bases (Chapter 2);
- A Hierarchical Predicate oriented K-Means clustering (HPKM) and a Predicate oriented Hierarchical Agglomerative clustering (PHAL) approach to partition graphs

into small semantically related sub-graphs with different purposes (Chapter 3);

- A dynamic query generation algorithm from outputs of topic discovery (Chapter 4);
- A topic aware link discovery algorithm to efficiently find paths with a context between the source and target nodes (Chapter 4);
- An ontology learning framework is proposed to extract keywords from unstructured data with a natural language processing technique and build an ontology based on retrieved words for further analysis (Chapter 5);
- Comprehensive experiments and evaluations for the proposed framework using DBPedia [12], Yago [115] and Bio2RDF [10] datasets (Chapters 2-5).

The rest of this dissertation is organized as follows. We first define primary concepts that are the basic building blocks of the Predicate Oriented Neighborhood Patterns (PONP) and present evaluation results related to the similarity measurement and pattern analysis in Chapter 2. In addition, we elaborate Hierarchical Predicate oriented K-Means clustering (HPKM) and Predicate oriented Hierarchical Agglomerative clustering (PHAL) approaches in detail and demonstrate their different features and running purposes in Chapter 3. Moreover, we introduce automatic query generation and topic aware linking discovery tools with experiment results in Chapter 4. Furthermore, we extend the functionality of the GraphKDD framework to make it suitable for retrieving information from unstructured data and adopting ontology building and learning as a further analysis step in Chapter 5. Finally, we conclude in Chapter 6 with the summary and future work.

CHAPTER 2

PREDICATE ORIENTED NEIGHBORHOOD PATTERNS

2.1 Introduction

We develop a pattern-based approach, called Predicate Oriented Neighborhood Patterns (PONP) to measure the similarity of graph. Fig. 1 shows the framework of the GraphKDD. The fundamental technique of the GraphKDD framework is PONP and its association similarity. PONP measures predicates similarity and quantifies relationship among predicates. Basically, the further two predicates locate, the less similarity score they are assigned.

On top of this base, we apply a top-down and a bottom-up unsupervised learning approaches to conduct a cluster analysis with similarity matrices generated according to PONP. Specifically, the top-down approach focuses on global optimization while the bottom-up approach is suitable for finding local optimization. Based on this observation, the top-down approach is suitable for single domain dataset clustering and the bottom-up approach is suitable for cross domains dataset clustering. Both of these techniques play an important role in divide a predicate space into several topics with similar contexts.

The GraphKDD framework supports full-fledged features of knowledge discovery including topic discovery from ontology, query generation from topics, source target reachability testing and path finding with context awareness. The topic discovery tool helps users to find related topics with similar contexts. Queries automatically generated from topics can be used to guide users to discover interesting topics of domains. The path finding tool generates all possible paths between the source and the target nodes based on the results from topic discovery process. More useful information can be found through the analysis of the paths starting from the source to the target.

In the GraphKDD framework, the underlying representation of data is based on “RDF that is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed.” As Fig. 1 shows, data from heterogeneous data sources will be converted, integrated and represented in RDF in GraphKDD. In this chapter, we first conduct a thorough survey of previous work on graph analysis and compare them with our PONP approach. Furthermore, we give a formal definition of the PONP approach and explain how to measure the PONP association in details.

2.2 Related Work

In this section, we present the state-of-the-art in graph analytics and then compare them with our work. In addition, the design of the PONP pattern will be justified.

Among some graph partition algorithms, SEDGE [130] provided a complementary partition approach to eliminate cross domain edges to facilitate query performance. SEDGE also proposed an on-demand partition to handle unbalanced query workload. Unfortunately, the partition is mainly based on physical relationships rather than semantic relationships. Mizan [64] made improvements based on Pregel [79] that is built on

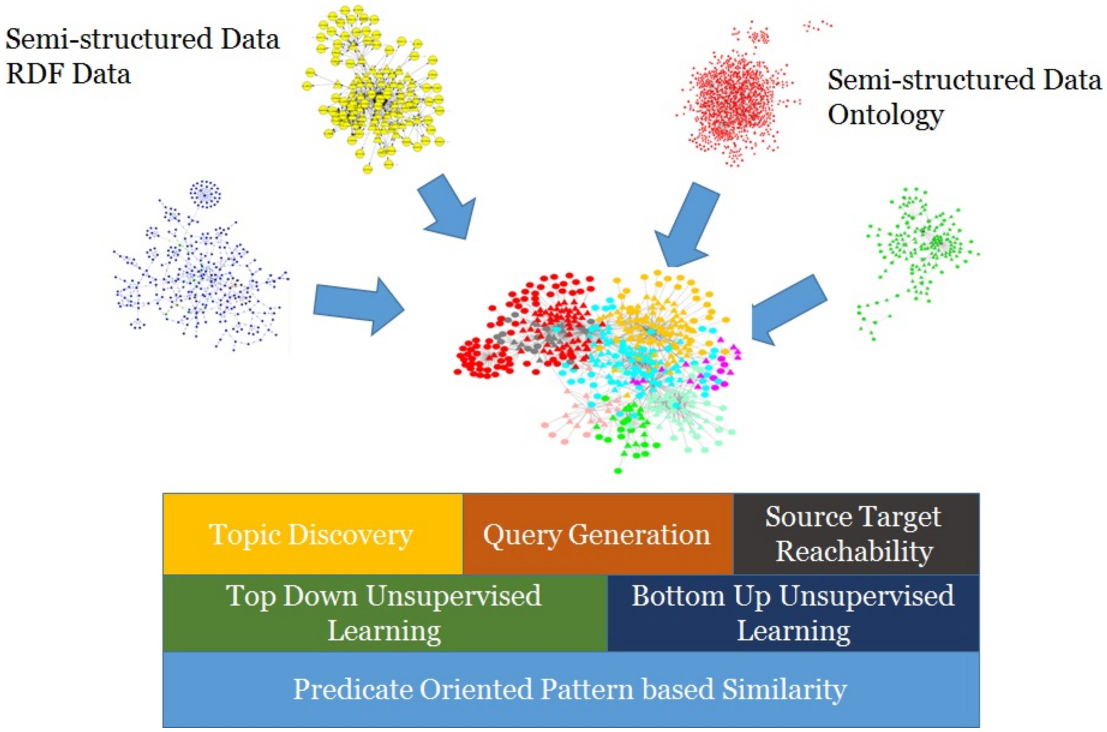

Semi-structured Data
RDF Data
Semi-structured Data
Ontology
Topic Discovery
Query Generation
Source Target Reachability
Top Down Unsupervised Learning
Bottom Up Unsupervised Learning
Predicate Oriented Pattern based Similarity

Figure 1: The GraphKDD Framework

Table 1: Comparison among Graph Analysis Approaches

Research	**Methods**	**Limitation**
SEDGE [130]	Complementary partition, On-demand partition ->query	Physical partition non-semantic
Mizan [64]	Dynamic graph partitioning strategies ->load balance	Physical partition non-semantic
GoFFish [111]	Subgraph centric. Connected component ->graph analytics	Physical partition non-semantic
Similarity-Driven Semantic Role Induction via Graph Partitioning [69]	Unsupervised method for semantic role induction. Vertex centric clustering to partition	No cross domain

Bulk Synchronous Parallel (BSP) [40] programming model. Mizan focused on dynamically efficient load balancing in terms of computation and communication among all worker nodes. It achieved load balancing by using fine-grained vertex migration in a distributed manner. In Mizan, a vertex centric model was designed, mainly focused on the size balance load of the graph rather than design of data migration by analyzing the contexts of work. Goffish [111] is a distributed approach that is sub-graph centric with connected components and furthermore makes abstraction for a large scale graph for an efficient graph analytics. This model combined the advantages of both the vertex centric approach and the shared-memory algorithms. However, Goffish does not support any context awareness. Similarity Driven Semantic Role Induction via Graph Partitioning [69] is a vertex centric unsupervised method for semantic role induction. But in this work, cross domains issues were not addressed. Table 1 gives a comparison among these approaches.

Compared to the existing approaches, the PONP pattern not only provide structure-based graph analysis but also support context awareness by analysis of topic information and description. In this sense, the PONP approach provides a better solution to partition both homogeneous and heterogeneous graphs in a different manner by grouping semantically related contents as well as similar contexts in sub-graphs.

Pattern based approaches were introduced for data mining and knowledge discovery. Specifically in biomedical informatics domain, Warrender and Lord [125] proposed an axiom based generalized and localized pattern driven approach in biomedical ontology engineering. Wang et al., [124] designed a biomedical pattern discovery algorithm based on a supervised learning approach. Rafiq et al., [91] developed an algorithm to discover temporal patterns in genomic databases. Van Leeuwen and Matthijs proposed an interactive way to do data mining by applying pattern mining in [121]. Gotz et al., [44] used electronic health record data as a use case to introduce an approach to perform data mining and visual analysis on clinical event pattern. Meanwhile, WHIDE [44] is a tool for colocation pattern mining in multivariate bioimages. Huang et al., [55] accomplished the goal of clinical pathway pattern discovery by using probabilistic topic models. Lasko et al., [70] introduced a computational phenotype pattern discovery with unsupervised learning on clinical data.

There are also many related work in general computer science research on pattern analysis and knowledge discovery. Trinity [132] performed graph mining on web scale RDF data by decomposing SPARQL queries into smaller subsets of triple patterns and then applied a sequence of graph explorations to come up with the combination of different triple patterns. k^2-Triples provided a compressed indexing approach to handle large RDF data in memory. It can also support triple pattern queries on such indexed RDF representation. RDFPeers [15] is a P2P based scalable distributed repository that supports disjunctive query pattern and conjunctive query pattern that were built based on atomic triple patterns. SPARQ2L [3] defined a formal syntax for path pattern expression to extract subgraph and find path in RDF databases.

However, our approach is different from these work. None of these techniques support measuring association among nodes. Moreover, none of these approaches handle both homogeneous and heterogeneous context. In this perspective, the PONP approach can find patterns within a defined boundary and support different strategies for discovering knowledge in a single domain as well as cross domains. What is more, the PONP focuses on a more general approach for graph structural pattern analysis and discovery. In addition, we have combined an unsupervised learning algorithm with a pattern discovery technique to provide a more dynamic way of knowledge discovery from large amount of ontologies.

Similar to our approach, there are also several research focus on predicate oriented approach to perform graph analysis. Shi, Baoxu, and Tim Weninger provided a predicate oriented path finding approach to do fact checking in large knowledge graph [110]. VEPathCluster [134] proposed a combination of vertex-centric and edge-centric approach for meta path graph analysis to enhance clustering quality of cross domains datasets. In addition to the path finding feature, the PONP approach formulates the relationship among

RDF predicates which form the prerequisite of topic discovery and federated query generation.

Besides predicate oriented graph analysis, there are also many work on concept based approach. Alani et al., [1] proposed a concept structure based ontology ranking system. Stuckenschmidt et al., [113] gave a graph partition solution by applying concept hierarchy. Formica and Anna [39] measured the formal concept analysis similarity by using ontology based approach. However, we decide to use predicate oriented approach for the reasons that i) predicate is easy to connect multiple domains which brings more information; ii) predicate is unique only referring to the unique context in a domain; iii) predicate maintains the same representation for both schema and data level, which leads to a easier way to apply schema level learning results to data level.

What is more, some researchers have conducted study on similarity measurements to do graph analysis and graph partition. Rouzbeh Meymandpour et al., [80] proposed a feature based semantic information content measurement for linked open data. Positive Matching Index (PMI) was given by Daniel et al., to measure similarity with optimal lists of attributes [33]. Other popular similarity measurement approaches such as SimRank [60] and [86] provided the idea of using neighborhood similarity to define node similarity, which is similar to PONP. However, our approach mainly focuses on a dynamic similarity assignment mechanism based on the boundary and distance of predicates' neighborhood.

Based on literature review and related work comparison, we propose Predicate Oriented Neighborhood Patterns (PONP). This approach specifies high connectivity on the RDF/OWL graph for information sharing and integration. A predicate P is representing a binary relation between two concepts (c1 and c2) in ontology. In RDF/OWL, P is represented as a property to express a kind of relationship (e.g., rdfs:subClassOf) between domain (subject) and range (object). The subject and object can be either from the same ontology or from different ontologies. In our study, relationships are defined by the empirical analysis of ontology data. We are particularly interested in predicates (relationships) that are different from existing approaches like PSPARQL [20] and SPARQLer [21].

Apart from being similar, predicates may share other aspects, e.g., sharing the same subjects or the same objects as well as the connectivity between predicates. This forces not only on concepts among graphs but also relationships of the concepts.

2.3 Formal Definition of Pattern and Topic

This research mainly focuses on doing knowledge discovery and ontology learning from the information network. Here we formally define some related terms that covered by this research.

- *Information Network:* The network with the ability to do content information exchange and holds complex linked relationship.
- *Homogeneous Information Network:* The information network shared the same context and resource. Referred as single domain datasets in this work.
- *Heterogeneous Information Network:* The information network with different contexts and resources. Referred as cross domains datasets in this work.

Table 2: RDF Notation

Shape	**Meaning**
Circle	RDF Concept
Triangle	RDF Predicate
Line	Relationship
Arrow	Direction

In the GraphKDD framework, the knowledge model is defined by levels of abstraction: (i) the smallest component is a predicate (relation) from a information network (RDF graphs), (ii) the intermediate component is a pattern that is defined by groups of predicates, (iii) at a higher abstraction level, a topic can be discovered from groups of patterns, and (iv) the highest level of abstraction that can be presented as an analytical view of multiple ontologies (cross domains). The relationships of ontologies (domains) can be determined from a comprehensive analysis of the discovered topics and patterns of predicates.

As the predicates define the relationships between subjects and objects, it is interesting to see that the relationships among subjects and objects are nicely defined through patterns and topics. In this research, we define the Predicate Oriented Neighborhood Patterns (PONP) that describes the association and collaboration among different predicates (relationships) and concepts in information networks. There are basically two types of the PONP patterns: *Share Pattern* and *Connectivity Pattern*.

For different patterns introduced below, graph visualization notation is shown in Table 2.

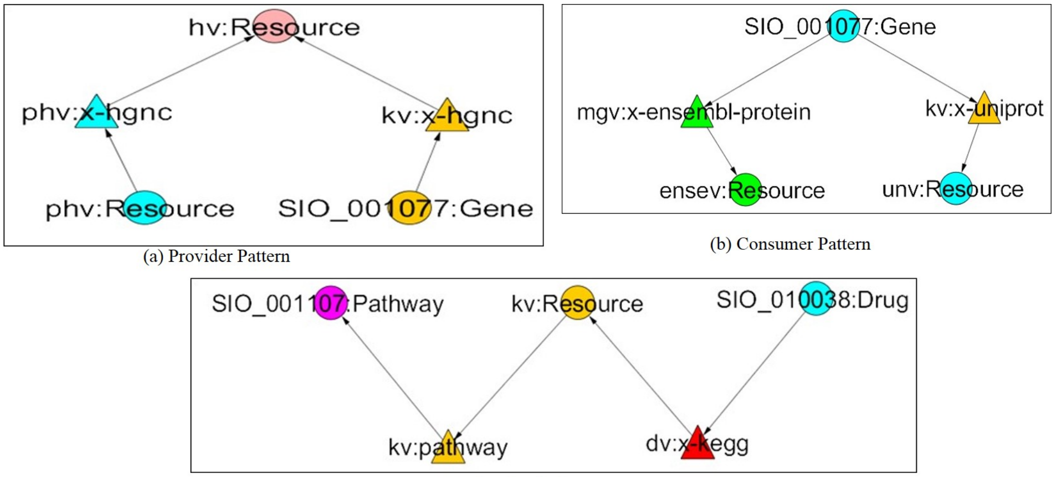
Definition 1: Share Pattern This pattern describes the resources sharing relationships between predicates where the resources are concepts from a information network (RDF graphs). Given two triples $\langle S_i, P_i, O_i \rangle, \langle S_j, P_j, O_j \rangle$, the conditions of the share pattern are defined as follows:

$$\forall S_i \in D_i, \forall P_i \in D_i, \forall O_i \in D_i \text{ and } \forall S_j \in D_j, \forall P_j \in D_j, \forall O_j \in D_j$$
$$(P_i \neq P_j)\&\&(S_i == S_j || O_i == O_j)\&\&(D_i \neq D_j).$$

where the logical OR operator ($||$) returns the Boolean value true if either or both operands is true and returns false otherwise, the logical AND operator ($\&\&$) returns the Boolean value true if both operands are true and returns false otherwise. For all (denoted by $\forall$) S_i, for all P_i and for all O_i are in a domain D_i and for all S_j, for all P_j, and for all O_j are in a domain D_j, but these two domains D_i and D_j are different.

There are three types of Share patterns are defined as follows:

- The *Provider* pattern describes the relationship with a pair of predicates sharing a common object, describes the provider role of entity giving information to Consumers. This role has more out-degree edges than in-degree edges.
- The *Consumer* pattern describes the relationship with a pair of predicates sharing a common subject, describes the role of entity receiving information from Providers. Consumer has more in-degree edges than out-degree edges.
- The *Reacher* pattern describes the relationship with a pair of predicates having a same concept as a subject and object, describes the role connecting the Provider role with the Consumer role.



(a) Provider Pattern

(b) Consumer Pattern

(c) Reacher Pattern

Figure 2: Share Patterns

Three share patterns (Provider, Consumer, Reacher) are shown as an example. In this diagram, the circle represents a concept and the triangle represents a predicate. Different colors indicate various domains they come from

Fig. 2 shows the share patterns from Bio2RDF datasets such that (a) Provider pattern: the object *hv:resource* is shared through two predicates *pv:x-hgnc* and *kv:x-hgnc* (b) Consumer pattern: the subject *SIO_001077:Gene* is shared with two predicates *mgv:x-ensembl-protein* and *kv:x-uniprot* (c) Reacher pattern: a concept *kv:Resource* is shared by two predicates *dv:x-kegg* and *kv:pathway*.

Definition 2: Connectivity Pattern This pattern describes the connectivity relationships at least three predicates in a information network. This Connectivity pattern is defined using the *Reacher* pattern from Definition 1. A subject (S_i) in a source domain (D_i) is connected to an object (O_i) in a target domain (D_j) through cross domains connectivity predicates $(P_i, P_j \in P_c$ and $D_i \neq D_j)$. The pattern of the source domain or the target domain is defined as a *Reacher* pattern. There are two types of the Connectivity pattern:

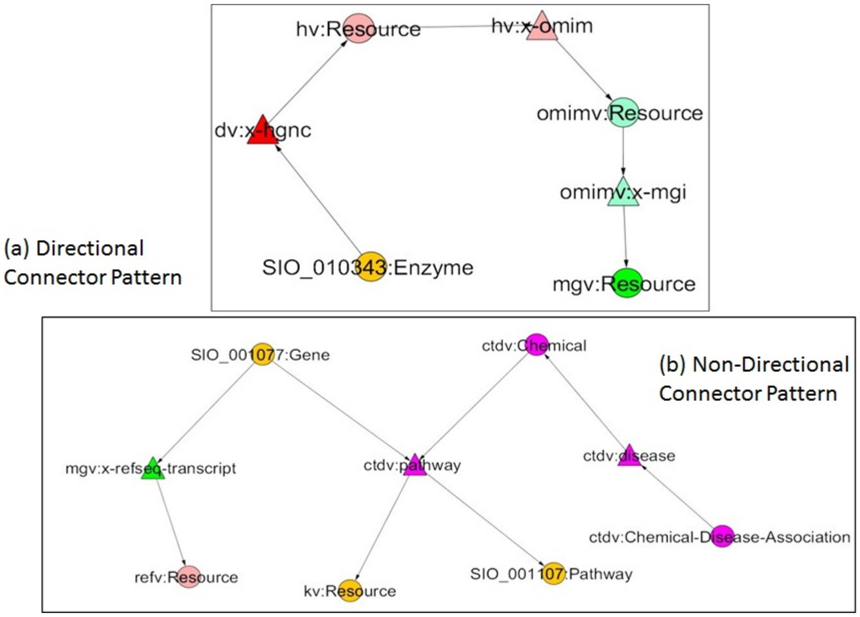


Figure 3: Connectivity Patterns

Two connectivity patterns (DC and NDC) are shown as an example. In this diagram, the circle represents a concept and the triangle represents a predicate. Different colors indicate various domains they come from

Directional Connector (DC) and *Non-Directional Connector* (NDC).

- The *DC* pattern describes the connectivity pattern considering the direction of the edges between predicates whose distance is higher than equal to 2.
- The *NDC* pattern is same with the DC pattern in terms of the predicate collaboration for indirect connectivity, however, the edge directions are not considered in this NDC pattern.

This Connectivity pattern is formally defined as follows: Given a *Reacher* pattern $\langle S_s, P_s, O_s \rangle$ and a new triple $\langle S_i, P_i, O_i \rangle$, the conditions of the connectivity pattern are as follows:

$$\forall S_s \in D_s, \forall P_s \in D_s, \forall O_s \in D_s \text{ and } \forall S_i \in D_i, \forall P_i \in D_i, \forall O_i \in D_i$$
$$(P_s \neq P_i)\&\&(O_s == S_i)\&\&(D_s \neq D_i).$$

where the logical AND operator (&&) returns the Boolean value true if both operands are true and returns false otherwise. For all (denoted by $\forall$) S_s, for all P_s and for all O_s are in a domain D_s and for all S_i, for all P_i, and for all O_i are in a domain D_i, but these two domains D_s and D_i are different.

Fig. 3 shows the Connectivity patterns in Bio2RDF datasets such that the subject and object are connected through three predicates: (a) Directional Connector (DC) among three predicates *dv:x-hgnc*, *hv:x-omim*, *ommimv:x-mgi* (b) Non-Directional Connector (NDC) among three predicates *mgv:x-refseq-transcript*, *ctdv:pathway*, and ctdv:disease.

We build topics based on different pattern predicates draw. We formally define topic and topic boundary in Definition 3 and 4.

Definition 3: Topic The *topic* describes bounded contexts through association patterns of both shared and connected predicates in a information network. Different topics may have completely different associations among any common predicates. In a graph to represent the topic (called the topic graph), a group of predicates collaborate each other to share and connect information through the predicates of the PONP patterns.

Definition 4: Topic Boundary The *topic boundary* (denoted as B) defines the scope of context in which the information can be associated and shared, and connected in a information network. The association and collaboration of information is described in terms of sets of concepts and relations within the given boundary on the information network.

Topic boundary can be depicted by topic radius and topic centrality defined by Definition 5 and 6.

Definition 5: Topic Radius The *topic radius* (denoted as R) defines the distance D between topic center to any other target nodes within one topic.

Definition 6: Topic Centrality The *topic centrality* (denoted as C) defines the center node for each topic. The center is calculated as the mean value of all predicate nodes within one topic. Boundary B can be determined by centrality C and radius R.

Boundaries between contexts (topics) can be determined by various factors. Usually the dominant one is strongly associated with others so that this can be measured by high in-degree/out-degree and distance in a information network. This boundary can be set differently depending on the domains of interest. Multiple contexts can be found within the same domain context and similarly a single context can be found across multiple domains.

Based on five basic patterns, we give Theorem 1 and 2 with proof to demonstrate the relationship among different patterns.

Theorem 1. For $\forall$ *Reacher* pattern $\{\mathrm{R}\}$ and *DC* pattern $\{\mathrm{D}\}$. if $\{\mathrm{R}\}$ and $\{\mathrm{D}\}$ share the same predicate P, then $\{\mathrm{R}\} \subsetneqq \{\mathrm{D}\}$.

Proof. Suppose $\forall$ concepts C, P_a and predicate P form a *Reacher* pattern, so that $P_a \rightarrow C \rightarrow P$ and $\{C,\ P,\ P_a\}$ = $\{\mathrm{R}\}$. Suppose $\forall$ concepts C, C_b and predicate P, P_a, P_b form a *DC* pattern, so that $P_b \rightarrow C_b \rightarrow P_a \rightarrow C \rightarrow P$ and $\{P_b,\ C_b,\ P_a,\ C,\ P\}$ = $\{\mathrm{D}\}$. Because $\{C,\ P,\ P_a\} \subset \{P_b,\ C_b,\ P_a,\ C,\ P\}$, therefore, $\{\mathrm{R}\} \subset \{\mathrm{D}\}$. Because the number of predicates in $\{\mathrm{R}\}$ is less than k, and the number of predicates in $\{\mathrm{D}\}$ is larger than or equal to k, so $\{\mathrm{R}\} \neq \{\mathrm{D}\}$. Therefore, $\{\mathrm{R}\} \subsetneqq \{\mathrm{D}\}$.

Theorem 2. For $\forall$ *Provider* pattern $\{\mathrm{V}\}$ and *Consumer* pattern $\{\mathrm{C}\}$, and *NDC* pattern $\{\mathrm{N}\}$, if they share the same predicate P, then $\{\mathrm{V}\} \subsetneqq \{\mathrm{N}\}$ and $\{\mathrm{C}\} \subsetneqq \{\mathrm{N}\}$.

Proof. Suppose $\forall$ concept C_a and predicates P, P_a form a *Provider* pattern, so that $P_a \rightarrow C_a \leftarrow P$ and $\{C_a,\ P,\ P_a\}$ = $\{\mathrm{V}\}$. Suppose $\forall$ concept C_b and predicates P, P_b form a *Consumer* pattern, so that $P \leftarrow C_b \rightarrow P_b$ and $\{C_b,\ P,\ P_b\}$ = $\{\mathrm{C}\}$. Suppose $\forall$ concepts C_a, C_b and predicates P_a, P_b, P form a *NDC* pattern, so that $P_a \rightarrow C_a \leftarrow P \leftarrow C_b \rightarrow P_b$ and $\{C_a,\ C_b,\ P,\ P_a,\ P_b\}$ = $\{\mathrm{N}\}$. Because $\{C_a,\ P,\ P_a\} \subset \{C_a,\ C_b,\ P,\ P_a,\ P_b\}$, $\{C_b,\ P,\ P_b\} \subset \{C_a,\ C_b,\ P,\ P_a,\ P_b\}$, so $\{\mathrm{V}\} \subset \{\mathrm{N}\}$ and $\{\mathrm{C}\} \subset \{\mathrm{N}\}$. Because the number of predicates in $\{\mathrm{V}\}$ and $\{\mathrm{C}\}$ is less than k, and the number of predicate in $\{\mathrm{N}\}$ is larger than or equal to k, so $\{\mathrm{V}\} \neq \{\mathrm{N}\}$ and $\{\mathrm{C}\} \neq \{\mathrm{N}\}$. Therefore, $\{\mathrm{V}\} \subsetneqq \{\mathrm{N}\}$ and $\{\mathrm{C}\} \subsetneqq \{\mathrm{N}\}$.

Based on Theorem 1 and 2, we also give Lemma 1, which indicates that DC pattern must be composed by Reacher pattern and NDC pattern must be composed by Provider or Consumer patterns.

Lemma 1. *DC* pattern must be composed by *Reacher* pattern and *NDC* pattern must be composed by *Provider* or *Consumer* patterns.

Proof. if $\exists$ a *DC* pattern, as Theorem 1 shows, there must be a *Reacher* pattern share a common predicate P with *DC* pattern. Therefore, *DC* pattern must be composed by *Reacher* pattern. Similarly, if $\exists$ a *NDC* pattern, as Theorem 2 shows, there must be a *Provider* pattern or *Consumer* pattern share a common predicate P with *NDC* pattern. Therefore, *NDC* pattern must be composed by *Provider* or *Consumer* pattern.

Based on Theorem 1, 2 and Lemma 1, we conclude that in any RDF graph, Predicate oriented Patterns can cover all the predicate neighborhood cases as shown in Theorem 3.

Theorem 3. In $\forall$ RDF graph, predicate oriented *Provider* pattern, *Consumer* pattern, *Reacher* pattern, *DC* pattern and *NDC* pattern cover all the predicate neighborhood cases.

Proof. In $\forall$ RDF graph G, the basic component is triplet. Suppose $\forall$ predicate neighborhood triplet in G as $\{P_a, C, P_b\}$, there are four cases with different combination of directions: 1) $P_a \rightarrow C \rightarrow P_b$;2) $P_a \rightarrow C \leftarrow P_b$;3) $P_a \leftarrow C \rightarrow P_b$;4) $P_a \leftarrow C \leftarrow P_b$. As Definition 1 and 2 shows, Case 2) belongs to *Provider* pattern, Case 3) belongs to *Consumer* pattern, Case 1) and 4) belong to *Reacher* pattern. Therefore, three shared patterns cover all the basic cases. In addition, as Lemma 1 shows, *DC* pattern is composed by *Reacher* pattern, NDC patterns can be composed by *Provider*, *Consumer* and *Reacher* pattern, therefore, five patterns are enough to cover all predicate neighborhood cases.

As a summary, Table 3 gives notations of each pattern. In this table, S denotes subject, P denotes predicate and O denotes object. In general, there are two patterns: Share pattern and Connectivity pattern. Specifically, general patterns involves Provider pattern, Consumer pattern and Reacher pattern. Triple properties examples for these three patterns are described in Table 3. Similarly, Connectivity pattern includes a Directional Connector pattern and a Non-directional Connector pattern. In Table 3, we use examples with three predicate notations to demonstrate them.

2.4 Association Measurements for Predicate Oriented Neighborhood Patterns

We now define the measurement for the Predicate Oriented Neighborhood Patterns (PONP) in terms of sets of concepts and relations (predicates) within a single domain or cross domains. For this purpose, we describe how to quantify associations between different predicates. It is based on the PONP pattern describing the relationships between predicates P_i and P_j through a concept C.

The association measurement for the PONP patterns varies based on different neighboring levels for each pair of predicates. Basically, we give a higher shared score to predicates with more shared concepts and lower scores to predicates with less shared concepts. Similarly, we give a higher connection similarity score to closer predicates and

Table 3: Summary of Patterns

General Pattern	Specific Patterns	Triple Properties
Share Pattern	Provider	Si->Pi->Oi & Sj->Pj->Oj & Oi = Oj
	Consumer	Si->Pi->Oi & Sj->Pj->Oj & Si = Sj
	Reacher	Si->Pi->Oi & Sj->Pj->Oj & Oi = Sj | Oj = Si
Connectivity Pattern	Directional Connector Pattern	Si->Pi->Oi & Sj->Pj->Oj & Sk->Pk->Ok & (Oi = Sj, Oj =Sk) | (Oj = Si, Oi = Sk) | (Oi = Sk, Ok = Sj) | (Oj = Sk, Ok = Si)
	Non-Directional Connector Pattern	Si->Pi<-Oi & Sj->Pj->Oj & Sk->Pk<-Ok & (Oi = Sj, Oj =Sk) | (Oj = Si, Oi = Sk) | (Oi = Sk, Ok = Sj) | (Oj = Sk, Ok = Si)

lower scores to further predicates.

The patterns are discovered with the bounded contexts which are a central concept in knowledge discovery. The clustering technique is applied to partition a large and complex network into multiple smaller topics in the same context in an optimal manner. The bounded contexts are specifically tailored for a set of cross domains patterns. The boundary B is determined based on the distance L (without considering direction) between any two predicates.

We formally define the association measurement and its related concepts in Definition 7-12.

Definition 7: Degree of Association The *degree of association* is defined to measure the degree of the association between predicates in a information network. The degree is defined with a weight assigned to links between predicates. The weight can differentiate the degree of the association between predicates. The rationale is to capture cross domains relations between predicates by giving a higher weight to the links across domains while giving a lower weight to links in a single domain. The degree is strongly related to the topic boundary B. As we are mainly interested in the relationships within the boundary, the weight strategy will be changed depending on the topic boundary.

In this research, we set the topic boundary B as 3 after heuristic evaluation and testing. In other words, the maximum distance between predicates (without considering the direction) in a topic is 3. Formal definition of ontology association and association distance are shown in Definition 8 and 9.

Definition 8: Ontology Association The *Ontology Association* defines the association among ontologies that depicts a high level of views on cross domains collaboration. Based on the predicate collaboration in the PONP patterns, the ontology association and collaboration model can be defined. For each pattern, the top *K* predicates are considered to build the ontology association model that represents the abstract relationships between these topics.

Definition 9: Topic Association The *Topic Association* indicates the distance between topics that describes the relationship and association among topics. It is calculated by measuring the distance or dissimilarity between center nodes for each topic.

Definition 10: Association Distance The *association distance* defines the distance between associated predicates in a information network. Given a directed graph $G(C,P)$, concepts C denote subject S and object O and P predicate in a RDF schema graph, respectively. Let $d(P_i,P_j)$ represent the number of concepts C between P_i and P_j. $r(P_i,P_j)$ determines if a predicate P_i is reachable from another predicate P_j where the domain D_i of P_i is not the same from the domain D_j of P_j, i.e., $D_i \neq D_j$, without considering the direction of links). $l(P_i,P_j)$ indicates the shortest distance between P_i and P_j.

$$l(P_i, P_j) = \begin{cases} 0 & P_i = P_j \\ 1 & d(P_i, P_j) = 1 \\ L_1 + L_2 & L_1 = d(P_i, P_k), L_2 = d(P_k, P_j) \\ & r(P_i, P_k) = true, r(P_k, P_j) = true, r(P_i, P_j) = true \end{cases} \tag{2.1}$$

The direct association describes the direct relationship between P_i and P_j in the distance L = 1 that is within the boundary B. The indirect association describes any relationship between P_i and P_j in distance L computed by Eq. (2.1) within the boundary B, i.e., $1 < L \leq B$. The share pattern is the directed association while the Connectivity pattern is the indirect association. We now define these two probability based similarity scores: i) $[SA](P_i, P_j)$ is defined a share pattern of any two predicates P_i and P_j ii) $[CA](P_i, P_j)$ for a Connectivity pattern of any two predicates.

Definition 11: Share Association Given predicates P_i and P_j in a directed RDF schema graph $G(C, P)$. Let $C(P_i)$ and $C(P_j)$ denote the entities (subjects or objects) that are directly connected to P_i and P_j. $l(P_i, P_j)$ is the reachability test for the given predicates P_i, P_j. $SA(P_i, P_j)$ indicates the probability-based association matrix for a share pattern between P_i and P_j.

$$SA(P_i, P_j) = \begin{cases} 1 & l(P_i, P_j) = 0 \\ 0 & l(P_i, P_j) \to \infty (nolink) \\ \frac{(|C(P_i)| \cap |C(P_j)|)^2}{|C(P_i)| * |C(P_j)|} & otherwise \end{cases} \tag{2.2}$$

Definition 12: Connectivity Association For a connectivity pattern of any two predicates P_i and P_j, $CA(P_i, P_j)$ defines the probability-based association for a connectivity pattern between P_i and P_j based on the share pattern. For the given share associations $SA(P_i, P_k)$ and $SA(P_k, P_j)$ and the distance between the predicates $l(P_i, P_j)$, the connectivity association can be computed as follows:

$$CA(P_i, P_j) = \begin{cases} SA(P_i, P_k).SA(P_k, P_j) & l(P_i, P_j) = 2 \\ \max_{1 \leq k < j} CA(P_i, P_k).CA(P_k, P_j)) & l(P_i, P_j) > 2 \end{cases} \tag{2.3}$$

The definition is influenced by the chain matrix multiplication problem (a kind of dynamic programming) of determining the optimal sequence for performing a series of operations. After we get the similarity score for all pairs of predicates, we use the formula in Eq. (2.2) and Eq. (2.3) to generate a predicate association matrix for clustering.

Definition 13: Predicate Association Matrix Given the total number of predicates n and the probability-based association score for share patterns $SA(P_i, P_j)$ and connectivity patterns $CA(P_i, P_j)$ between predicatesP_i and P_j, $PA[P_i, P_j]$ indicates an association matrix for all pairs of predicates P_i and P_j

$$PA[P_i, P_j] = \begin{cases} CA(P_i, P_j) & l(P_i, P_j) >= 2 \\ SA(P_i, P_j) & Otherwise \end{cases} \tag{2.4}$$

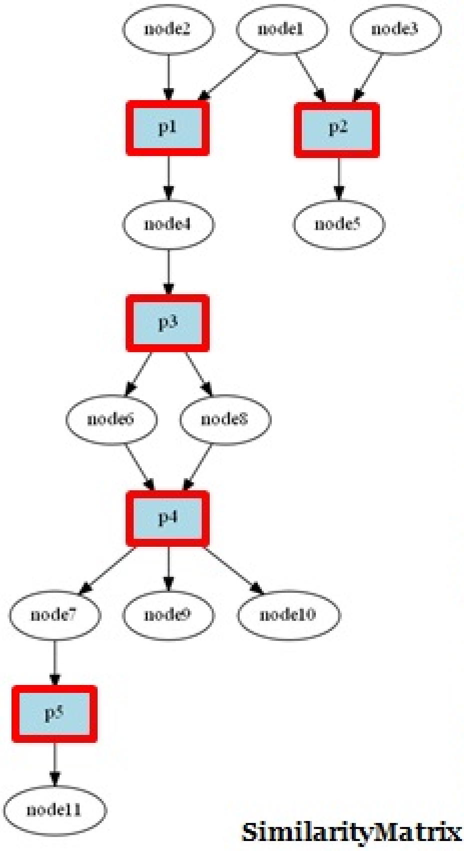
Fig. 4 shows an example of the predicate similarity computation for shared patterns and connection patterns with the consideration of direction. In this example, a shared pattern is identified between predicates P_1 and P_2 and connection patterns are identified between P_1 and P_3, P_1 and P_4, P_1 and P_5. Based on the PONP patterns, $PA[P_i, P_j]$ is computed as shown in Fig. 4.

Similarly, we apply the same strategy on concept to conduct the comparison study. In the same RDF graph, Fig. 5 gives a simple example to explain how to compute concept oriented similarity in level1, level2 and level3. In this example, we use notation N_k to indicate $node_k$. N_1 and N_4, N_4 and N_6 are located on the first level because there is only one predicate between a pair of concepts. Based on Eq. (2.2), the similarity score $PS_s(N_1, N_4)$ and $PS_s(N_4, N_6)$ are 0.5 and 0.25, respectively. N_1 and N_6 are in the second level, their similarity score is calculated by combine $PS_s(N_1, N_4)$ and $PS_s(N_4, N_6)$, which is 0.125. N_1 and N_7 are in the third level, their similarity score $PS_c(N_1, N_7)$ is the maximum score from $PS_s(N_1, N_4) * PS_c(N_4, N_7)$, $PS_c(N_1, N_6) * PS_s(N_6, N_7)$ and $PS_c(N_1, N_8) * PS_s(N_8, N_7)$, which is 0.03125.

Specifically, for cross domains datasets, we give weight optimization to predicate neighborhood association in order to make cross domains predicates relationship outstanding. Cross domains weight optimization is defined below.

Definition 13: Cross Domains Weight Optimization For $\forall$ topic T_i with average similarity association score $\overline{T_i}$, if predicates pair P_i,P_j forms a cross domains relationship with association score t_{i_j}, we define t'_{i_j} as an optimized association score between P_i and P_j, such that

$$t'_{i_j} = \begin{cases} \frac{t_{i_j}+\overline{T_i}}{2} & t_{i_j} < \frac{t_{i_j}+\overline{T_i}}{2} \\ t_{i_j} & t_{i_j} \geqslant \frac{t_{i_j}+\overline{T_i}}{2} \end{cases} \tag{2.5}$$



Level1: $(P_1, P_2), (P_1, P_3)$

$PS_s(P_1, P_2) = \frac{1}{3*3} = \frac{1}{9} = 0.11$

$PS_s(P_1, P_3) = \frac{1}{3*3} = \frac{1}{9} = 0.11$

$PS_s(P_3, P_4) = \frac{2^2}{3*5} = \frac{4}{15} = 0.27$

$PS_s(P_4, P_5) = \frac{1}{5*2} = \frac{1}{10} = 0.1$

Level2: (P_1, P_4)

$PS_c(P_1, P_4)$
$= PSs(P_1, P_3) * PSs(P_3, P_4)$
$= \frac{1}{9} * \frac{4}{15} = \frac{4}{135} = 0.03$

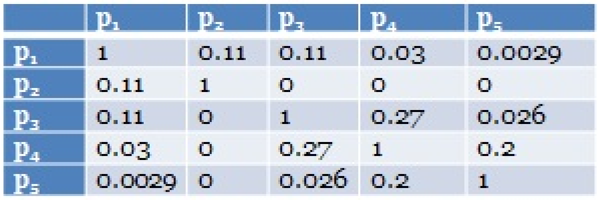
$PS_c(P_3, P_5)$
$= PSs(P_3, P_4) * PSs(P_4, P_5)$
$= \frac{4}{15} * \frac{1}{10} = \frac{4}{150} = 0.026$

Level3: (P_1, P_5)

$PS_c(P_1, P_5) =$
$Max(PSc(P_1, P_4) * PSs(P_4, P_5),$
$PS_s(P_1, P_3) * PSc(P_3, P_5)) =$
$Max(\frac{4}{135} * \frac{1}{10}, \frac{1}{9} * \frac{4}{150}) = 0.0029$

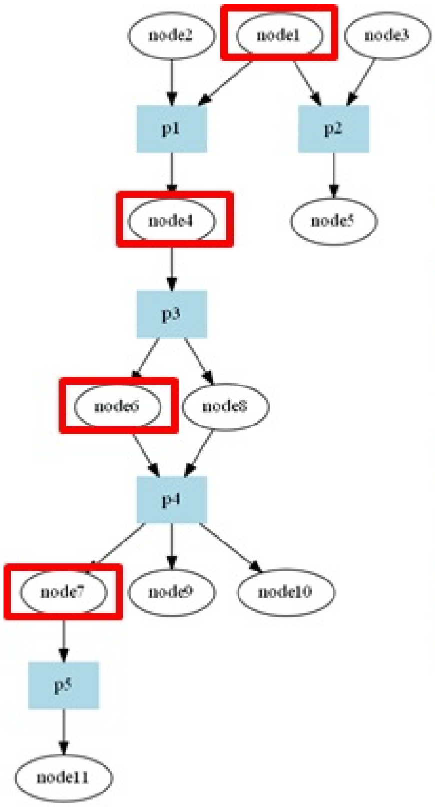
$PS_c(P_2, P_3) = 0$
$PS_c(P_2, P_4) = 0$
$PS_c(P_2, P_5)) = 0$

SimilarityMatrix(SM)



	p_1	p_2	p_3	p_4	p_5
p_1	1	0.11	0.11	0.03	0.0029
p_2	0.11	1	0	0	0
p_3	0.11	0	1	0.27	0.026
p_4	0.03	0	0.27	1	0.2
p_5	0.0029	0	0.026	0.2	1

Figure 4: PONP and Similarity Matrix (Directional Based)



Level1: (N_1, N_4), (N_4, N_6)

$$PS_s(N_1,N_4)=\frac{1}{1*2}=\frac{1}{2}=0.5$$

$$PS_s(N_4,N_6)=\frac{1}{2*2}=\frac{1}{4}=0.25$$

Level2: (N_1, N_6)

$$PS_c(N_1,N_6)=PS_s(N_1,N_4)*PS_s(N_4,N_6)=\frac{1}{2}*\frac{1}{4}=\frac{1}{8}=0.125$$

Level3: (N_1, N_7)

$$PS_c(N_1,N_7)=Max(PSs(N_1,N_4)*PS_c(N_4,N_7), PS_c(N_1,N_6)*PS_s(N_6,N_7)), PS_c(N_1,N_8)*PSs(N_8,N_7))=Max(0.03125, 0.03125, 0.03125)=0.03125$$

Figure 5: Concept based Neighborhood Similarity (Directional Based)

2.5 Evaluation and Results

In this evaluation, we first conduct a comparison experiment between predicate based similarity association and concept based similarity association. We then analyze the patterns we discover in the GraphKDD framework.

2.5.1 Data Specification

In this study, we use DBpedia, YAGO and Bio2RDF 9 domains datasets as use cases. All datasets schema are downloaded from the datahub repository, which is at the address *https://datahub.io/dataset/.* Format of each dataset is RDF N-triple. A summary of predicates, concepts, schema and instance for each dataset is given in Table 4. From the table, it is obvious that DBpedia has a dominant number of predicates but maintain a relative small number of concepts. YAGO also contains more predicates than concepts. Bio2RDF 9 domains dataset has more concepts than predicates.

2.5.1.1 Predicate oriented Similarity and Concept oriented Similarity

We compare the predicate association scores with the concept association scores for each dataset. Both the predicate oriented and the concept oriented approaches are evaluated in the GraphKDD framework. For each approach, we calculate the similarity association matrix. We compute the average similarity score for predicate and concept based approach as shown in Table 5. In general, we find that the predicate based similarity holds a higher association score than concept based approach for all three cases. However, for DBpedia and YAGO, the predicate based average similarity score is low. Compared

Table 4: Datasets Summary

	# Predicates	**# Concepts**	**# Schema Triple**	**# Instance Triple**
DBpedia	943	104	1837	3,000,000,000
YAGO	119	30	943	120,000,000
Bio2RDF 9 domains	330	374	126	795,329,244

Table 5: Average Similarity Measurement

	Average Predicate Similarity	**Average Concept Similarity**	**Average Predicate Similarity (SimRank)**	**Average Concept Similarity (SimRank)**
DBpedia	0.07	0.01	0.0075	0.0058
YAGO	0.12	0.04	0.0346	0.0285
Bio2RDF 9 domains	0.51	0.04	0.43	0.01

to the high similarity score with Bio2RDF 9 domains datasets, the reason is that DBpedia and YAGO are single domain dataset, so there are not much association among nodes. In addition, it is because both predicate and concept based similarity are not directly proportional to the size of predicates and concepts. From the evaluation, we found out that the large number of predicates or concepts does not mean there is a high level of interconnection among them.

The SimRank based experiment with predicate and concept oriented similarity measurements is also conducted. The results are shown in Table 5. Similar to the PONP outputs, the predicate oriented approach performs better than the concept based approach, and Bio2RDF 9 domains datasets show the highest similarity score. The GraphKDD framework performs a lot better than SimRank in all cases.

2.5.1.2 Ranking of Patterns and Topics in Cross Domains

We also conduct an analysis experiment to illustrate each specific pattern involved in cross domains ontology. In this study, we use Bio2RDF with 9 domains datasets to interpret the pattern based evaluation. First of all, we compute the rankings of predicates, patterns, and topics discovered from our knowledge discovery process and also summarize the relationships among ontologies based on the discovered patterns and topics.

Predicate and Concept Ranking: The predicates, the primary atomic component in GraphKDD, and their associated concepts are ranked based on their in-degree and out-degree. From this analysis, we find out the roles of ontologies for cross domains collaboration in a information networks. As shown in Fig. 6(b), among 330 predicates, top predicates such as *dv:source* and *dv:calculated.properties* are from three ontologies such as DrugBank, ClinicalTrials, and PharmGKB. Similarly, among 374 concepts, top concepts such as *clinv:Resource*, *kv:Resource*, *dv:Resource*, *phv:Resource* are shown in in Fig. 6(a). These predicates and concepts are mainly from the primary ontologies including ClinicalTrials, KEGG, DrugBank, and PharmGKB.

Cross Domains Predicate and Concept Ranking: The contents of cross domains are ranked based on the in-degree/out-degree of cross domains concepts and predicates. We observe the cross domains rankings with predicates and concepts is different from the previous ranking. However, the ontologies playing important roles are similar. Fig. 7 shows 40 cross domains concepts and predicates. Among them, SIO:Drug, kv:Resource and SIO:Gene are top 3 cross domains concepts of PharmGKB (SIO normalized), KEGG, and DrugBank (SIO normalized). *kv:pathway*, *clinv:arm.group* and *dv:x.kegg* are top 3

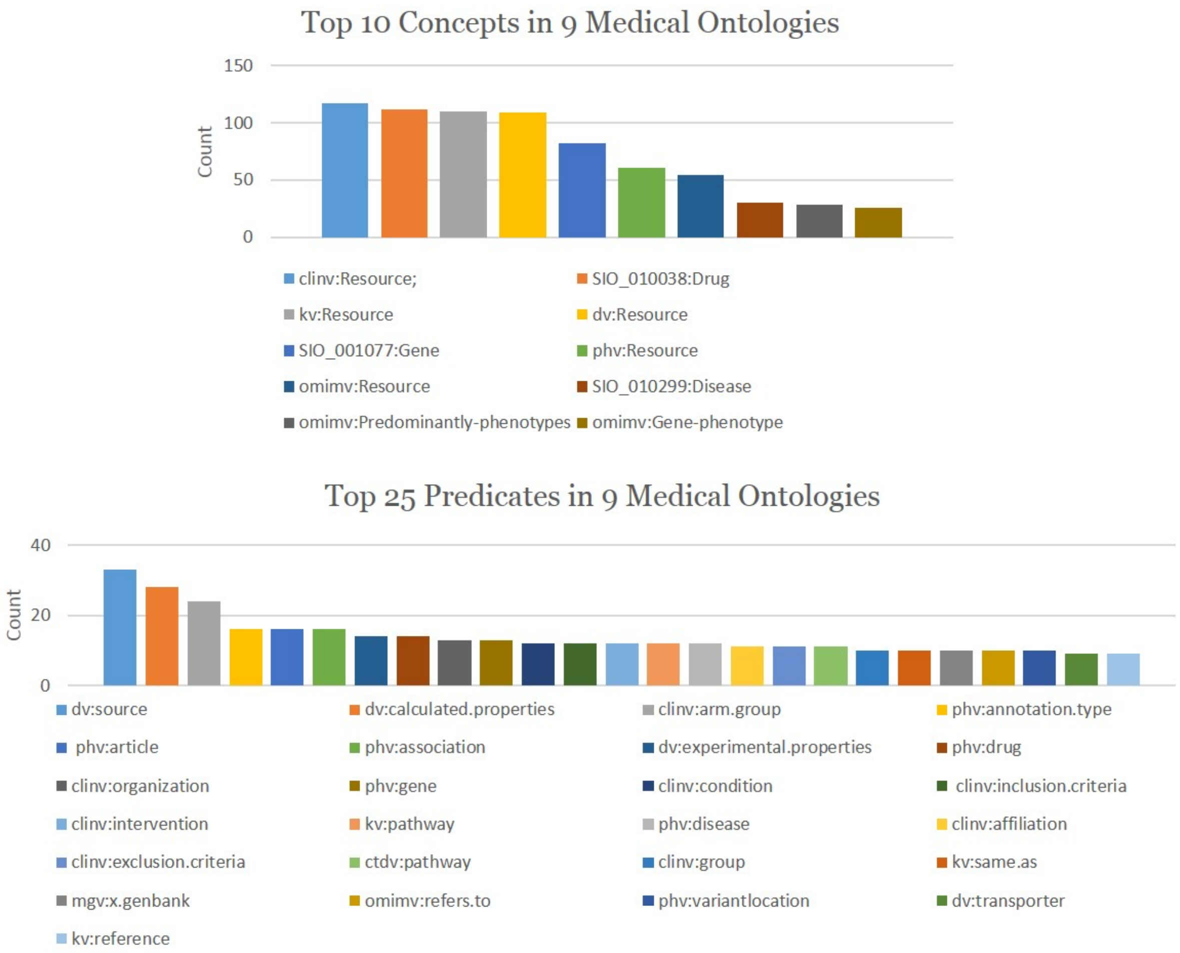


Figure 6: Top Concepts and Predicates: (a) Top 10 Concepts (b) Top 25 Predicates

The prefixes describe the domain of the concepts and predicates. clinv: http://bio2rdf.org/clinicaltrials_vocabulary dv: http://bio2rdf.org/drugbank_vocabulary kv: http://bio2rdf.org/kegg_vocabulary mgv: http://bio2rdf.org/mgi_vocabulary omimv: http://bio2rdf.org/omim_vocabulary phv: http://bio2rdf.org/pharmgkb_vocabulary

Table 6: Top 3 Predicates for 5 Top Topics with Bio2RDF 9 Domains

Num of Topic	Num of Predicate	Top 3 Predicates
Topic 16	119	dv:source; dv:calculated.properties; clinv:arm.group
Topic 25	72	dv:calculated.properties; clinv:arm.group; phv:annotation.type
Topic 23	39	dv:calculated.properties; clinv:arm.group; kv:pathway
Topic 22	36	dv:calculated.properties; clinv:arm.group; kv:pathway
Topic 26	24	clinv:arm.group; kv:pathway; mgv:x.genbank

Table 7: Top 2 Unique Predicates for 5 Top Topics Bio2RDF 9 Domains

Num of Topic	Num of Predicate	Top 2 Unique Predicates
Topic 16	119	phv:drug; phv:disease
Topic 25	72	phv:association; phv:article
Topic 23	39	clinv:group; kv:module
Topic 22	36	pathway; dv:x.uniprot
Topic 26	24	dv:transporter; dv:target

cross domains predicates of KEGG, ClinicalTrials, and DrugBank, respectively.

Topic Ranking with Cross Domain Features: These patterns are ranked according to primary features such as cross domains predicates, predicate popularity (in-degree/out-degree of the predicates), and domain verity (the number of ontologies in which the patterns are captured). Fig. 8(a) shows top 5 topics (Topic 16, Topic 25, Topic 23, Topic 22 and Topic 26) computed by the cross domains features. Table 6 and 7 show the top 3 predicates and top 2 unique predicates of these topics.

Topic Ranking with Cross Domain Neighborhood Patterns: Topics are ranked based on the PONP patterns. Fig. 8(b) shows top 5 topics (Topic 16, Topic 25, Topic 23, Topic

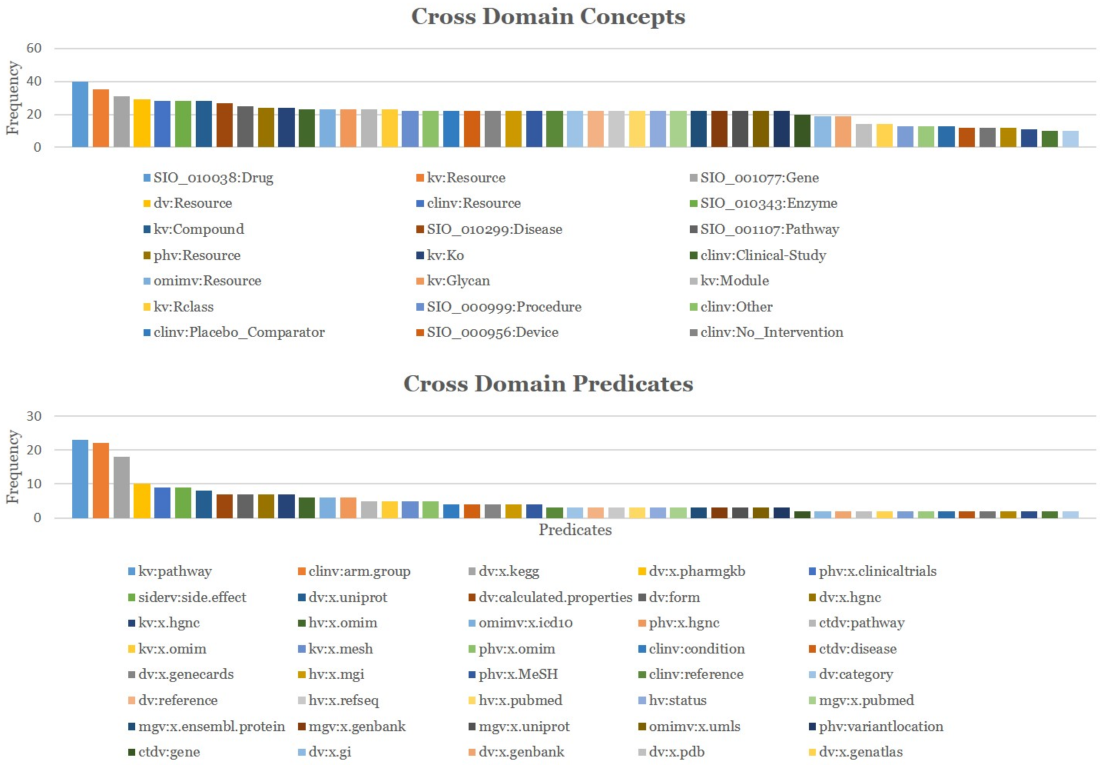
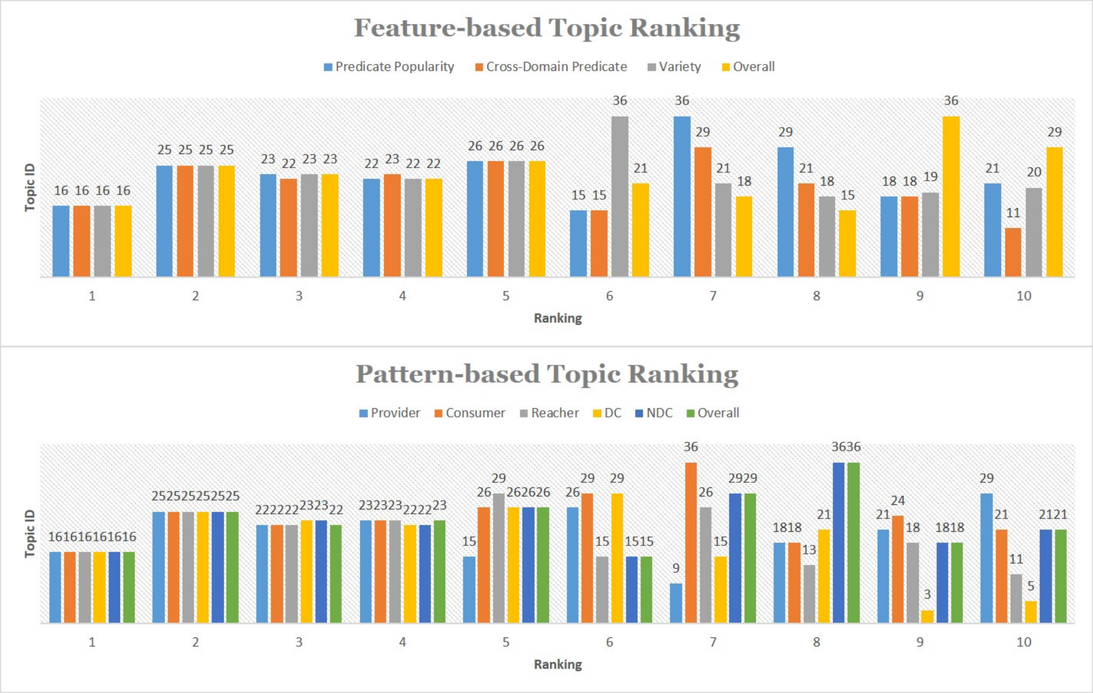


Figure 7: Cross Domain Concept and Predicate Ranking: (a) Top 40 Concepts (b) Top 40 Predicates

The prefixes describe the domain of the concepts and predicates. clinv: http://bio2rdf.org/clinicaltrials_vocabulary ctdv:http...bio2rdf.org.ctd_vocabulary dv: http://bio2rdf.org/drugbank_vocabulary hv: http://bio2rdf.org/hgnc_vocabulary kv: http://bio2rdf.org/kegg_vocabulary mgv: http://bio2rdf.org/mgi_vocabulary omimv: http://bio2rdf.org/omim_vocabulary phv: http://bio2rdf.org/pharmgkb_vocabulary sider: http://bio2rdf.org/sider_vocabulary


Feature-based Topic Ranking

Pattern-based Topic Ranking

Figure 8: Cross Domain Topic Ranking: (a) Feature-based Ranking (b) Pattern-Based Ranking

Popularity is measured by In-degree/Out-degree of predicates. Verity is measured by the number of ontologies involved. The numbers in the bar graph are the topic ID (ranged: 1 - 43).

22 and Topic 26). The ranking based on the counts of the PONP patterns (*Provider*, *Consumer*, *Reacher*, *CD* and *NCD* patterns) is very similar to the ranking computed by the predicate popularity, cross domains predicate, and variety shown in Fig. 8(a). This confirms that the proposed pattern-based approach reflects an excellent understanding of the important features of the network such as density, verity, and popularity.

2.5.1.3 Ontology Patterns in Cross Domains

Based on top five PONP patterns (Provider, Consumer, Reacher, Directional Connector, Non-Directional Connector) as case studies, we analyze the collaboration between ontologies as shown in Fig. 9, 11, 13, 15 and 17. Topic graphs are depicted in 10, 12, 14, 16 and 18. In each graph, a color is assigned to each ontology as follows: DrugBank: Red; HGNC: Pink; MGI: Green; PharmGKB: Cyan; ClinicalTrials: Yellow; OMIM: Sky Blue; SIDER: Gray; KEGG: Orange; CTD: Magenta. Semanticscience Integrated Ontology (SIO) represents the normalized name for integrated medical ontologies. For each case study, we now show its topic pattern graph of concepts and predicates and the instances of concepts/predicates in this topic graph.

Case 1: Provider Patterns in Ontology Collaboration Five ontologies (DrugBank, PharmGKB, ClinicalTrials, KEGG and CTD) are involved in the collaboration of the provider pattern. In this collaboration, we find that DrugBank and KEGG are a *Provider*, CTD is a *Balancer*, and PharmGKB is a *Consumer* as well as a *Bridger*. ClinicalTrials is its *Consumer*. Fig. 9 shows an ontology collaboration graph for the given provider

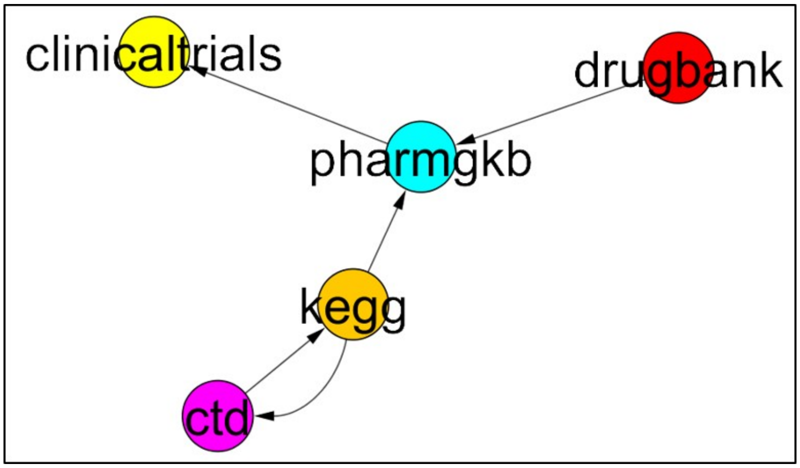


Figure 9: Ontology Provider Pattern

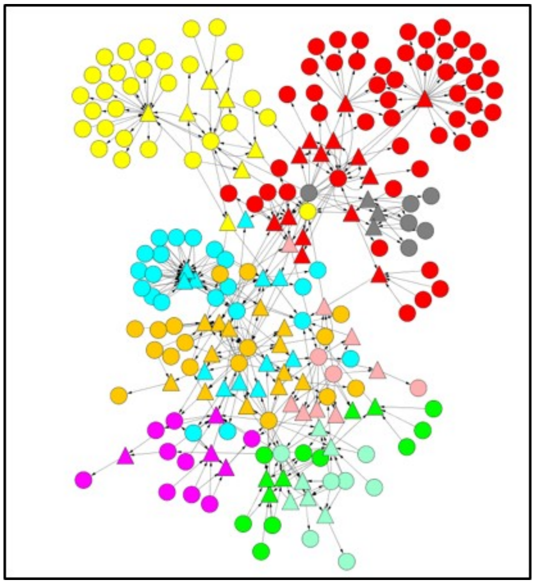


Figure 10: Topic 25: Provider Pattern Graph

Table 8: Provider Pattern in Topic 25

phv:Resource	hv:Resource	SIO_001077:Gene
epidermal growth	factor receptor	Gene Symbol for EGFR EGFR, ERBB, ERBB1, HER1, PIG61, mENA; epidermal growth factor receptor (EC:2.7.10.1); K04361 epidermal growth factor receptor [EC:2.7.10.1]
complement component 1, r subcomponent	Gene Symbol for C1R	C1R; complement component 1, r subcomponent (EC:3.4.21.41); K01330 complement component 1, r subcomponent [EC:3.4.21.41]
complement component 1, q subcomponent, B chain	Gene Symbol for C1QB	C1QB; complement component 1, q subcomponent, B chain; K03987 complement C1q subcomponent subunit B
complement component 1, s subcomponent	Gene Symbol for C1S	C1S; complement component 1, s subcomponent (EC:3.4.21.42); K01331 complement component 1, s subcomponent [EC:3.4.21.42]

interleukin 2 receptor, beta	Gene Symbol for IL2RB	IL2RB, CD122, IL15RB, P70-75;
		interleukin 2 receptor, beta;
		K05069 interleukin 2 receptor beta

pattern. Fig. 2(a) shows a provider pattern in Topic 25. This pattern describes the collaboration of two predicates, namely *phv:x-hgnc* and *kv:x-hgnc* to integrate information from three domains. Specifically, PharmGKB *Resource* links to KEGG *Gene* (SIO normalized) through HGNC *Gene symbol*. Table 8 shows 5 instances of the concepts in the provider pattern of Topic 25.

Case 2: Ontology Collaboration with Consumer Patterns Five ontology, namely KEGG, OMIM, DrugBank, CTD, and PharmGKB are involved. We find that CTD is a consumer

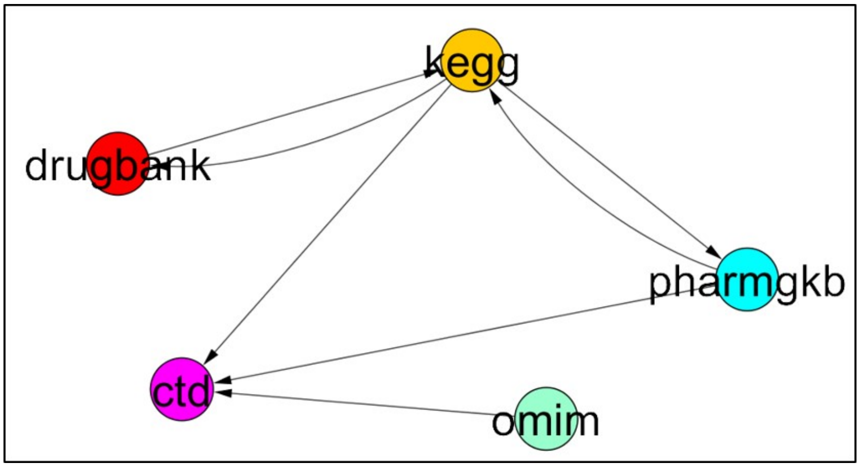


Figure 11: Ontology Consumer Pattern

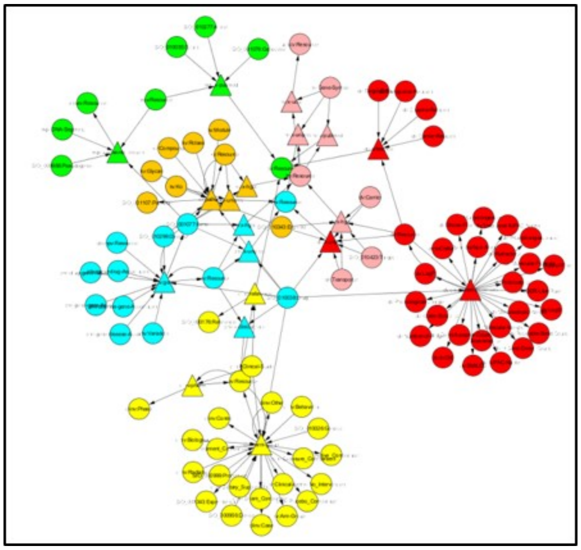


Figure 12: Topic 15: Consumer Pattern Graph

Table 9: Consumer Pattern in Topic 15

SIO_001077:Gene	**ensev:Resource**	**unv:Resource**
Brd7	ENSMUSP00000034085	Bromodomain-containing protein 7
C1qbp	ENSMUSP00000077612	Complement component 1 Q subcomponent-binding protein, mitochondrial
Ddx21	ENSMUSP00000042691	Nucleolar RNA helicase 2
Kcnab1	ENSMUSP00000047480	Voltage-gated potassium channel subunit beta-1
Nip7	ENSMUSP00000034392	60S ribosome subunit biogenesis protein NIP7 homolog

of KEGG, OMIM and PharmGKB. DrugBank are a balancer with KEGG. Fig. 11 shows an ontology collaboration graph for the consumer pattern, *CTD*. Fig. 2(b) shows a consumer pattern in Topic 15. This consumer pattern shows the collaboration between predicates *mgv:x-ensembl-protein* and *kv:x-uniprot* as a consumer of the PharmGKB concept (SIO normalized), *SIO_001077:Gene*. The collaboration is established across three domains such as KEGG, MGI and PharmGKB. In this pattern, due to the collaboration of these two consumer predicates, the Uniprot concept *resource* is linked to the Ensemble concept *Resource* through PharmGKB concept *Gene* (SIO normalized). Table 9 shows 5 instances of the concepts in the consumer pattern of Topic 15.

Case 3: Ontology Collaboration with Reacher Patterns Only two predicates from two ontology PhargGKB and ClinicalTrials are involved in the Reacher pattern. From this pattern analysis, we find that PharmGKB plays a *Provider* and ClinicalTrials a *Consumer* from this collaboration. Fig. 13 shows the ontology collaboration with the reacher pattern between PharmGKB and ClinicalTrials. Fig. 2(c) shows a reacher patterns in Topic 22.

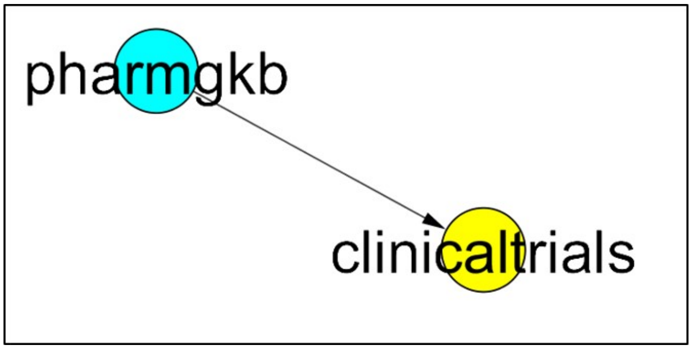


Figure 13: Ontology Reacher Pattern

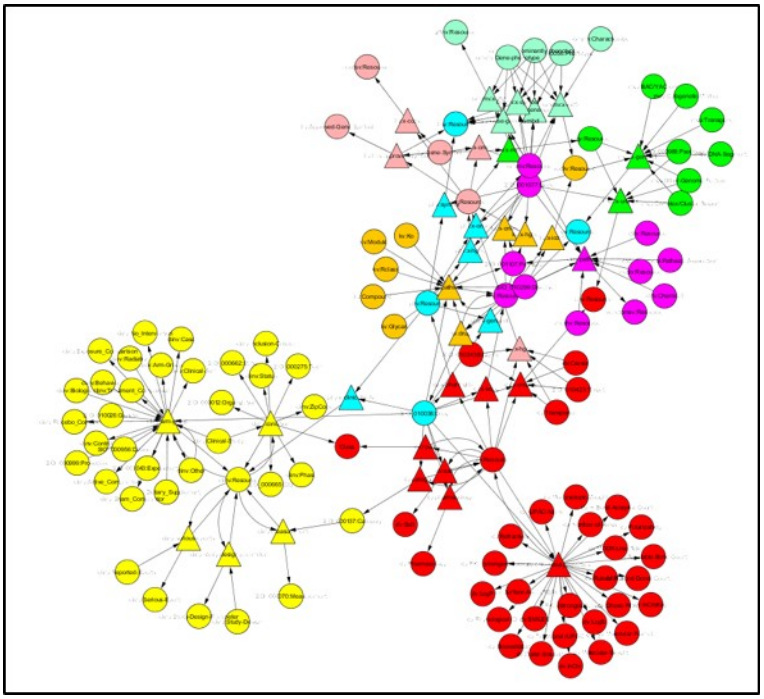


Figure 14: Topic 22: Reacher Pattern Graph

Table 10: Reacher Pattern in Topic 22

SIO 010038:Drug	**kv:Resource**	**SIO 001107:Pathway**
L-Lysine	L-Lysine; Lysine acid; 2,6-Diaminohexanoic acid	ABC transporters
Succinic acid	Succinate; Succinic acid; Butanedionic acid; Ethylenesuccinic acid	Citrate cycle (TCA cycle)
Glycine	Glycine; Aminoacetic acid; Gly	Biosynthesis of amino acids
Pyruvic acid	Pyruvate; Pyruvic acid; 2-Oxopropanoate; 2-Oxopropanoic acid; Pyroracemic acid	Pentose phosphate pathway
L-Glutamic Acid	L-Glutamate; L-Glutamic acid; L-Glutaminic acid; Glutamate	Biosynthesis of secondary metabolites

This reacher pattern is generated by predicates *kv:pathway* and *dv:x-kegg*, across four domains (PharmGKB, DrugBank, KEGG, CTD). Through the collaboration of these two predicates in this pattern, the PharmGKB concept *Drug* (SIO normalized) is linked to the KEGG concept *Resource* and the KEGG concept *Resource* is linked to the CTD concept *Pathway* (SIO normalized). Table 10 shows 5 instances of the concepts in the reacher pattern of Topic 22.

Case 4: Ontology Collaboration with Directional Connector Patterns From the pattern analysis with top 40 predicates, all nine ontologies have the Directional Connect (DC) patterns. Fig. 15 shows the ontology collaboration through the DC patterns with 54 links among these ontologies. We find that CliniclalTrials, DrugBank and SIDER play the role of *Provider* and CTD, HGNC, KEGG, MGI, OMIM, PharmGKB *Consumer*. Furthermore, KEGG, PharmGKB, SIDER, HGNC play the role of *Bridger*. The connection among the ontologies are established through the Bridger pattern. Fig. 3(a) shows a DC

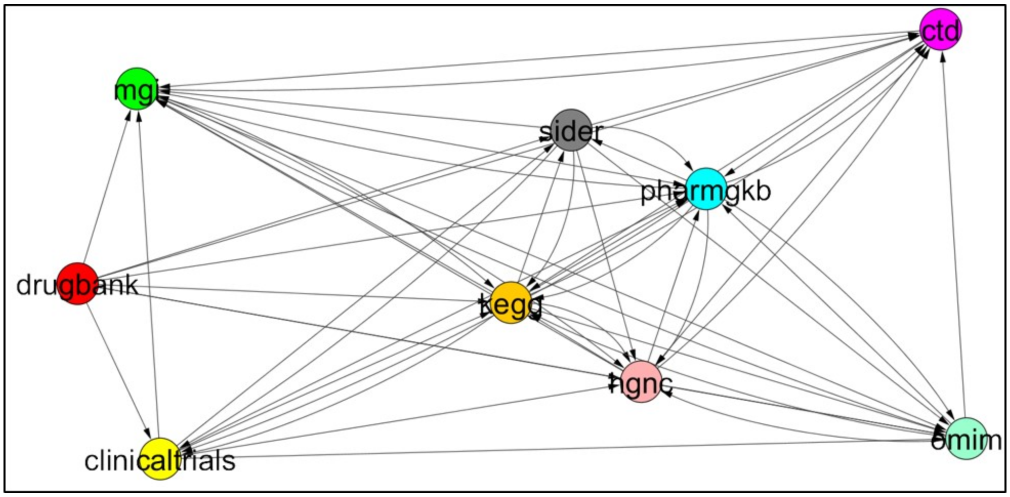


Figure 15: Ontology Directional Connector Pattern

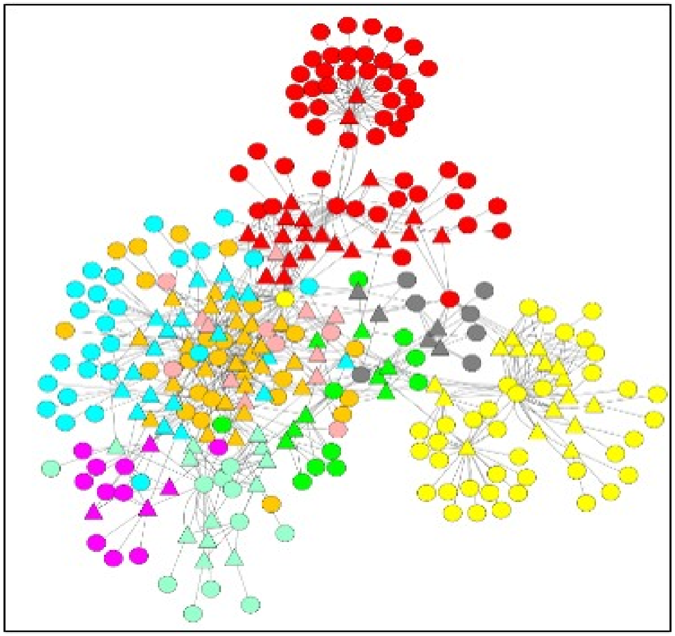


Figure 16: Topic 16: Directional Connector Pattern Graph

Table 11: Directional Connector Pattern in Topic 16

SIO 010343:Enzyme	hv:Resource	omimv:Resource	mgv:Resource
Prostaglandin G/H synthase 2	Gene Symbol for PTGS2	PROSTAGLANDIN-ENDOPEROXIDE SYNTHASE 2; PTGS2	Ptgs2
Vitamin K-dependent protein C	Gene Symbol for PROC	PROTEIN C; PROC	Proc
Cytochrome P450 2C9	Gene Symbol for CYP2C9	CYTOCHROME P450, SUBFAMILY IIC, POLYPEPTIDE 9; CYP2C9	Cyp2c65
CYP3A	Gene Symbol for CYP3A7	CYTOCHROME P450, SUBFAMILY IIIA, POLYPEPTIDE 7; CYP3A7	Cyp3a13
Cob(I)yrinic acid a, c-diamide adenosyltransferase, mitochondrial	Gene Symbol for MMAB	MMAB GENE; MMAB	Mmab

pattern in Topic 16. In this DC pattern of Topic 16, three predicates such as *dv:x-hgnc*, *hv:x-omim* and *omimv:x-mgi* are used to connect concepts across five domains (KEGG, DrugBank, HGNC, OMIM, MGI). In this pattern, the KEGG concept *Enzyme* (SIO normalized) links to the HGNC concept *Resource*. The HGNC concept *Resource* links to the OMIM concept *Resource*, and the OMIM concept *Resource* links to the MGI concept *Resource*. All the paths within the bounded context (the maximum distance between predicates, B = 3) can be determined by the DC patterns. From this pattern, we find many paths. One of them is the path ⟨*SIO_010343:Enzyme* → *dv:x-hgnc* → *hv:Resource* → *hv:x-omim* → *omimv:Resource* → *omimv:x-mgi* → *mgv:Resource*⟩. Table 11 shows 5 instances of the concepts in the DC pattern of Topic 16.

Case 5: Ontology Collaboration with Non-Directional Connector Patterns In the

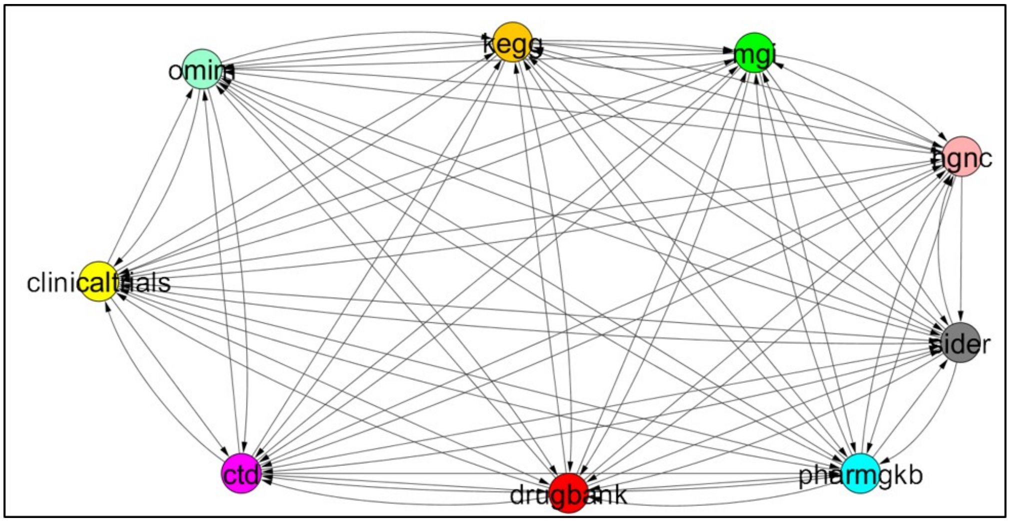


Figure 17: Ontology Non-Directional Connector Pattern

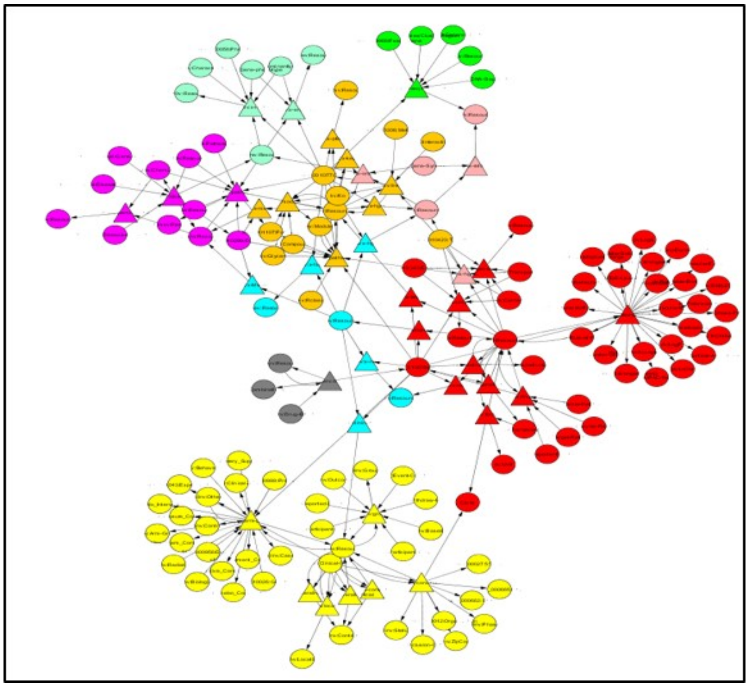


Figure 18: Topic 23: Non-Directional Connector Pattern Graph

Non-directional Connector (NDC) patterns discovery, all the 9 ontologies are fully participated. Fig. 17 shows the ontology collaboration through the NDC patterns. These 9 ontologies are connected with 72 links, which means all of them are fully connected. Interestingly, all of them have the same number of in-degree and out-degree, so that they are well balanced. Thus, no bridge pattern is required in this collaboration. Fig. 3(b) shows an ontology collaboration graph generated from the non-directional connector pattern (NDC) in Topic 23. This NDC pattern is composed with four predicates such as *mgv:x-refseq-transcript*, *ctdv:pathway* and *ctdv:disease* that are used to connect nine different domains (KEGG, DrugBank, MGI, HGNC, SIDER, PharmGKC, ClinicalTrials, OMIM, CTD). Specifically, in this pattern, those three predicates are used to connect six concepts such as *KEGG Gene* (SIO normalized), *Refseq resource*, *KEGG Resource*, *CTD Chemical*, *KEGG Pathway* (SIO normalized) and *CTD Chemical-disease-association*. Table 12 shows 5 instances of the NDC pattern in Topic 23.

2.6 Summary

We have defined five different patterns to elaborate the structural relationship for predicates in RDF graph based on basic Share pattern and Connectivity pattern. In addition, we have defined topic and topic boundary in information network based on patterns. Moreover, we have designed a Predicate Oriented Neighborhood Patterns (PONP) to measure the similarity among predicates. A dynamic programming based algorithm is designed to calculate the similarity association according to the distance and probability of shared concepts between each pair of predicates. Specifically, for cross domains datasets, we also developed an optimization solution to enhance the cross domains predicate similarity scores.

We conducted a comparative study to evaluate the predicate oriented and concept oriented approaches. The evaluation was conducted using DBpedia, YAGO and Bio2RDF 9 domains datasets. The results showed that PONP is better than the concept based approach. A similar experiment was conducted with the SimRank algorithm. The predicate oriented approach also performs better than SimRank for Bio2RDF 9 domains datasets. For the cross domains pattern analysis using Bio2RDF 9 domains datasets, these results showed some cross domain topics were discovered. Based on the discovered topics, interesting relationships among ontologies were discovered.

Table 12: Non-Directional Connector Pattern in Topic 23

SIO 001077, (Gene)	**refv:Resource**	**v:Resource**	**Chemical-Disease-Association**	**Chemical**	**SIO 001107, (Pathway)**
Fbxl12	NM 001002846	SDKD	1,10-phenanthroline (C025205) & Plasminogen Activator Inhibitor-1 Deficiency	Plasminogen Activator Inhibitor-1	p53 signaling pathway
Gjb6	NM 001010937	SDKD	2-nitro-4-phenylenediamine (C014706) & Interleukin 2 Receptor, Alpha, Deficiency of	Interleukin 2, Receptor Alpha	Cytokine-cytokine receptor interaction
Dclre1b	NM 001025312	SDKD	2-(methylamino) isobutyric acid (C017911) & Insulin-Like Growth Factor I Deficiency	Insulin-Like Growth Factor I	Oocyte meiosis
BC053393	NM 001025435	SDKD	2-methoxy-5-(2’,3’,4 -trimethoxyphenyl) tropone (C030370) & Combined Saposin Deficiency	Combined Saposin	Lysosome
Maf	NM 001025577	SDKD	2-methoxy-5-(2’,3’,4’ -trimethoxyphenyl) tropone (C030370) & Krabbe Disease, Atypical	Combined Saposin	Metabolism

CHAPTER 3

UNSUPERVISED LEARNING ON PONP ASSOCIATION MEASUREMENT

3.1 Introduction

In this chapter, we first give some background information of data clustering, graph partition and topic discovery. Then we conduct a comprehensive literature survey of previous work on knowledge discovery from different perspective of algorithms. In addition, we introduce innovated top-down and bottom-up unsupervised learning. In the end, we conduct evaluation and make result discussion on these two approaches.

We start to apply a systematic way of building topics from ontology after generating PONP association measurement. The workflow of the GraphKDD is shown as Fig. 19. As Chapter 2 introduces, first two steps are about applying predicate oriented pattern analysis on integrated RDF/OWL data and build predicate associate measurement matrix. Then the GraphKDD will apply clustering algorithms on such matrix. There are two different clustering algorithms, one is a bottom-up approach and another one follows top-down approach. The GraphKDD will apply different algorithms for different purposes of managing and extracting knowledge of data. We will discuss the detailed methodology and evaluation of clustering approaches in this chapter. In addition, knowledge discovery applications can be made based on clustering outputs, which will be covered in Chapter 4.

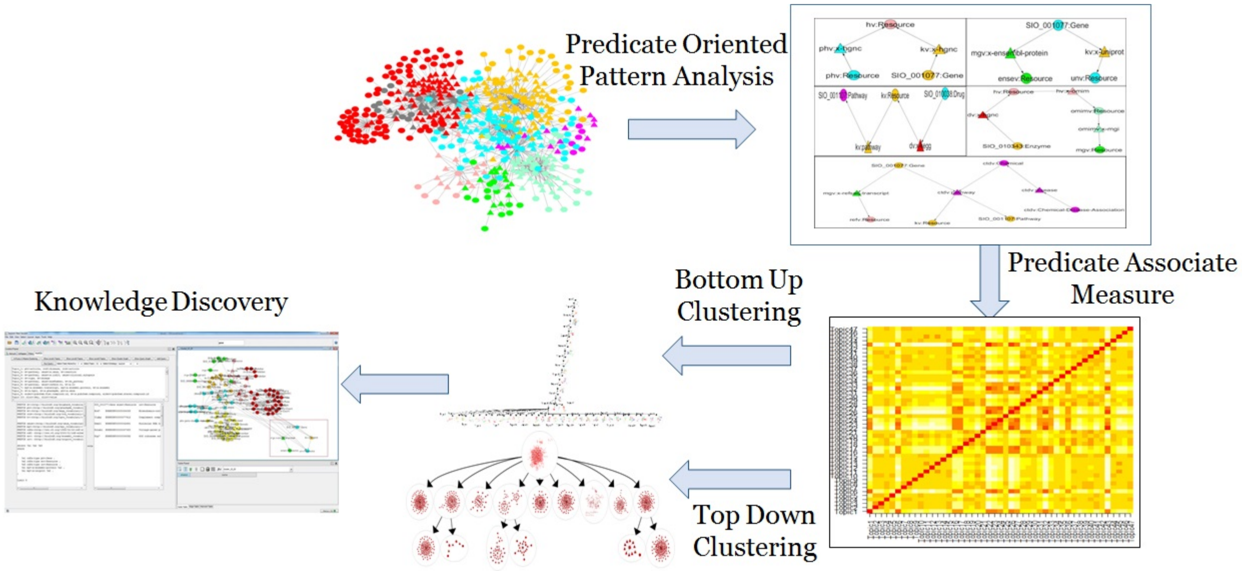


Figure 19: Workflow of the GraphKDD Framework

3.2 Background

We live in an era of information explosion. The manual process for discovering knowledge from such big data is not possible. Therefore, unsupervised learning algorithms are very effective for extracting useful information from big data. In this chapter, we present our clustering algorithms, named Hierarchical Predicate oriented K-Means (HPKM) and a Predicate oriented Hierarchical Agglomerative (PHAL) clustering, that are the extensions of existing unsupervised algorithms, i.e., K-Means clustering and hierarchical clustering algorithms. To validate these clustering algorithms, we compare them with K-Means [49], Pam [120], Clara [63], Hierarchical Clustering [62] in the evaluation part. The supervised and unsupervised learning are defined below.

- *Supervised Learning* provides a automatic way to infer knowledge based on training data.
- *Unsupervised Learning* is one of the machine learning approach that make clusters of datasets based on observed relationship that is not labeled explicitly.

In addition, here we introduce the definition of global similarity optimization and local diversity optimization at the beginning of this chapter. Specifically, HPKM is suitable for global similarity optimization and PHAL is suitable for local diversity optimization. These two concepts will be used to compare HPKM and PHAL.

- *Global Similarity Optimization:* The guiding principle is to minimize inter-cluster (inter-topic) similarity and maximize intra-cluster (intra-topic) similarity, based on the similarity measure for the PONP patterns. The PONP pattern-based similarity and the silhouette width (SW) are computed for achieving the objective of the clustering which is maximizing intra-cluster similarities and minimizing inter-cluster similarities. If the SW of a topic is higher than α, this topic will be clustered into K smaller topics. For each topic, we computed the average $sw(p_i)$ over all data of a topic as a measure of how tightly grouped all the predicates in the topic are. Thus the average $sw(p_i)$ over all predicates of the entire dataset is a measure of how appropriately the predicates have been clustered. The details on the global similarity optimization are available in [107].
- *Local Diversity Optimization:* The diversity optimization aims to determine the center and boundary of a topic. We address the diversity optimization as a local

approach considering local and diverse properties (i.e., cross domains properties and concepts) so that a center of topic needs to be determine.

Graph partition works directly on graph data and aims at partitioning graph into smaller components with specific properties. The focusing of graph partition is different from which of clustering algorithm. In this research, we map graph data into metric spaces and apply unsupervised clustering algorithm on it to find correlation among RDF predicates and generate small context topic of information. To validate the optimal branching factor of our unsupervised learning approach, we also include existing graph partition algorithm implemented by GraphX (Random Vertex Cut, Canonical Random Vertex Cut, Edge Partition 1D and Edge Partition 2D) [128]. Definition of branching factor and four random graph partition algorithms are shown below.

- *Branching Factor* defines the number of out-degree children successor nodes at each predecessor in a hierarchical manner.

- *Random Vertex Cut* follows vertex cut approach and considers source for applying hash function. In this algorithm, directions are considered.

- *Canonical Random Vertex Cut* follows vertex cut approach. It is similar to $RandomVertexCut$, but directions are not considered.

- *Edge Partition 1D* follows edge cut approach. Hash function is applied on source vertex ID. Edges are assigned to to the partitions according to the source vertices.

- *Edge Partition 2D* follows edge cut approach. Hash function is applied on both source vertex id and destination vertex ID.

Knowledge discovery is the unique feature of the GraphKDD framework. Based on PONP and unsupervised HPKM and PHAL approaches, the GraphKDD is able to find context awareness topic with predicates and concepts inside. To validate the knowledge discovery results coming out from the GraphKDD framework, we use Latent Dirichlet allocation (LDA) [14] to generate topics with the same datasets and compare the outputs for both approaches.

- *Latent Dirichlet Allocation* is able to find similar group of information by analyzing a set of observations in document. Let α represents the per-document topic distribution, β indicates the per-topic word distribution, θ_i represents the topic distribution for document i, φ_k indicates the word distribution for topic k, Z_{ij} points out the topic for the jth word in document i, and W_{ij} indicates the specific word, the equation of LDA is shown as below:

$$P(W, Z, \theta, \varphi, ; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{jt}|\theta_j) P(W_{jt}|\varphi Z_{jt}) \tag{3.1}$$

To validate the outputs among topics, we also analyze similarity among them. Specifically, we use Cosine [116], Jaccard [57] and Probabilistic Similarity [71] to make the comparison. Formal definition and equation for each of the similarity measurement is shown below.

- *Cosine Similarity* (CS) is used as a measure of similarity between two vectors of an inner product topic space that measures the cosine of the angle between topics. Given two vectors of topics, T_a and T_b, the cosine similarity is computed using a dot product and magnitude using the formula defined as follows:

$$\cos(\theta) = \frac{\sum\limits_{i=1}^{n} a_i b_i}{\sqrt{\sum\limits_{i=1}^{n} a_i^2}\sqrt{\sum\limits_{i=1}^{n} b_i^2}} \tag{3.2}$$

where a_i and b_i are components of vector T_a and T_b, respectively.

- *Jaccard Similarity Coefficient* (JSC) is defined as the size of the intersection divided by the size of the union of the sample sets as follows:

$$JSC(T_a, T_b) = \frac{|Pred(T_a) \cap Pred(T_b)|}{|Pred(T_a) \cup Pred(T_b)|} \tag{3.3}$$

where Pred(T) returns the list of predicates in given topic T and $0 \leq JSC(T_a, T_b) \leq 1$. If T_a and T_b are both empty, we define $JSC(T_a, T_b) = 1$.

- *Probabilistic Similarity* (PS) defines the degree of association between topics T_a and T_b as follows [107]:

$$PPS(T_a, T_b) = \frac{(|Pred(T_a) \cap Pred(T_b)|)^2}{|Pred(T_a) \cdot Pred(T_b)|} \tag{3.4}$$

where Pred(T) returns the list of predicates in given topic T.

3.3 Related Work

We compare the GraphKDD clustering approach with existing similarity and random based partition algorithm as shown in Table 13. In addition to neighbor measurement, structural analysis, data locality and dynamic/static discovery, the GraphKDD has an unique strength to do latent pattern discovery that other four partition algorithms are

Table 13: The GraphKDD with Graph Partition Algorithms

	Neighbor Measurement	**Structural Analysis**	**Data Locality**	**Dynamic/ Static Discovery**	**Latent Pattern Discovery**
Neighbor Matching [86]	Yes	Yes	No	Yes	No
SimRank [60]	Yes	Yes	Yes	Yes	No
Rand Vertex Cut [128]	No	Yes	Yes	Yes	No
Rand Edge Cut [128]	No	Yes	Yes	Yes	No
GraphKDD	Yes	Yes	Yes	Yes	Yes

Table 14: The GraphKDD with Clustering Algorithms

	Top Down	**Bottom Up**	**Optimization**	**Fuzzy**	**Context Aware**
K-Means [49]	Yes	No	No	No	No
FCM [11]	Yes	No	No	Yes	No
Pam [120]	Yes	No	No	No	No
Clara [63]	Yes	No	No	No	No
Hierarchical Clustering [62]	Yes	Yes	No	No	No
GraphKDD	Yes	Yes	Yes	Yes	Yes

not capable of doing.

Because the GraphKDD adopts unsupervised learning approach, we compare the GraphKDD with some of the famous existing clustering algorithms as Table 14 shows. Each one can perform top-down approach to clustering graph. However, only the GraphKDD and Hierarchical Clustering [62] are able to apply a bottom-up solution as well. Moreover, only the GraphKDD and Fuzzy C-Means clustering (FCM) [11] are able to make fuzziness output for each cluster. The GraphKDD has the unique features of optimization support and context awareness.

3.4 Hierarchical Predicate Oriented K-Means Clustering

The clustering approach we propose here is based on the similarity association measurement of the Predicate Oriented Neighborhood Patterns (PONP) inherent in the ontologies. We posit that predicate oriented clustering is a required step for efficient query processing involving the alignment and integration of ontologies. Given that predicates are more closely related to some predicates than others, predicates can be clustered for efficient query processing - the task of classifying a collection of predicates into clusters (or topics). The guiding principle is to minimize inter-cluster (inter-topic) similarity and maximize intra-cluster (intra-topic) similarity, based on the similarity measure for the PONP.

The *degree of diversity* is defined to measure the degree of the association between predicates from different domains in a heterogeneous information network. The diversity degree is defined with an optimal weight assigned to links between predicates from different domains. The weight can optimize the degree of the diverse association between diverse predicates from different domains. The rationale is to capture diverse relations between predicates from multiple domains by giving a higher weight to the links across domains while giving a lower weight to links in a single domain. The details on the local and diverse weight optimization are available in [106].

We first present our top-down clustering algorithm, called the Hierarchical Predicate oriented K-Means clustering (HPKM) that is designed by combining the divisive hierarchical clustering algorithm [62] and K-Means algorithm [49] for generating K topics level-by-level in an optimal manner. Similar to the K-Means algorithm, the HPKM is an unsupervised learning approach partitioning ontologies into k topics by clustering each predicate in the ontologies with the nearest mean. Similar to the divisive hierarchical clustering algorithm [62], the HPKM clusters ontologies into smaller topics in a hierarchical manner. The PONP similarity association score and the silhouette width (SW) are computed for achieving the objective of the clustering which is maximizing intra-cluster similarities and minimizing inter-cluster similarities [22]. If the SW of a topic is higher than α, this topic will be clustered into K smaller topics. The value of silhouette $sw(p_i)$ can be ranged between -1 and 1. For each predicate p_i, we compute the following two similarity: inter-cluster similarity and intra-cluster similarity.

Intra-cluster similarity $a(p_i)$: This measure refers to the similarity of data in a single cluster. Let $a(p_i)$ be the average dissimilarity of p_i (taking the inverse of the SM matrix computed from the PONP algorithm) with all other data within the same cluster. It can be validated how well p_i is assigned to its cluster according to $a(p_i)$ such as the smaller the value, the better the assignment. We then define the average dissimilarity of predicate p_i to a cluster C as the average of the distance from p_i to predicates in C_i.

Inter-cluster similarity $b(p_i)$: This measure refers to the similarity between clusters. Let $b(p_i)$ be the lowest average similarity of p_i to the sibling clusters C_j that has the same parent cluster with C_i of which p_i is not a member. The cluster with this lowest average similarity is said to be the "sibling (neighboring) cluster", C_j, of $p_(i)$ because it is the next best fit cluster for predicate p_i.

A silhouette width can be computed as follows:

$$sw(p_i) = \frac{b(p_i) - a(p_i)}{max(a(p_i), b(p_i))} \tag{3.5}$$

More specifically, it can be defined as follows: There are three possible cases about the silhouette width: (i) If the silhouette width $sw(p_i)$ is close to one, this means that the predicate p_i is appropriately clustered. (ii) If $sw(p_i)$ is close to a negative one, then the predicate p would be not appropriate here but would be more appropriate if it is clustered in its neighboring cluster C_j. (iii) If $sw(p_i)$ is near zero then this means that the predicate p_i is on the border of two natural clusters, namely C_i and C_j.

$$sw(p_i) = \begin{cases} 1 - \frac{a(p_i)}{b(p_i)} & if a(p_i) < b(p_i) \\ 0 & if a(p_i) = b(p_i) \\ \frac{b(p_i)}{a(p_i)} - 1 & if a(p_i) > b(p_i) \end{cases} \tag{3.6}$$

For each topic, we compute the average $sw(p_i)$ over all data of a topic as a measure of how tightly grouped all the predicates in the topic are. Thus the average $sw(p_i)$ over all predicates of the entire dataset is a measure of how appropriately the predicates have been clustered.

With the Bio2RDF Drugbank dataset as an example, Fig. 20 shows the average $sw(p_i)$ over all predicates of each topic at each level. For example, at level 1, K= 2 is computed using the SW. Furthermore, after partitioning into two topics, the silhouette widths, 0.89 (for 20 predicates) and 0.71 (for 43 predicates) are computed for each topic. At level 2, for the left topic, K= 5 and for the right topic, K = 2 are computed, respectively. After clustering, silhouette widths, 0.52 (for 4 predicates) and 0.7 (for 6 predicates), 0.59 (for 3 predicates), 0.92 (for 4 predicates), and 0.38 (for 3 predicates) and

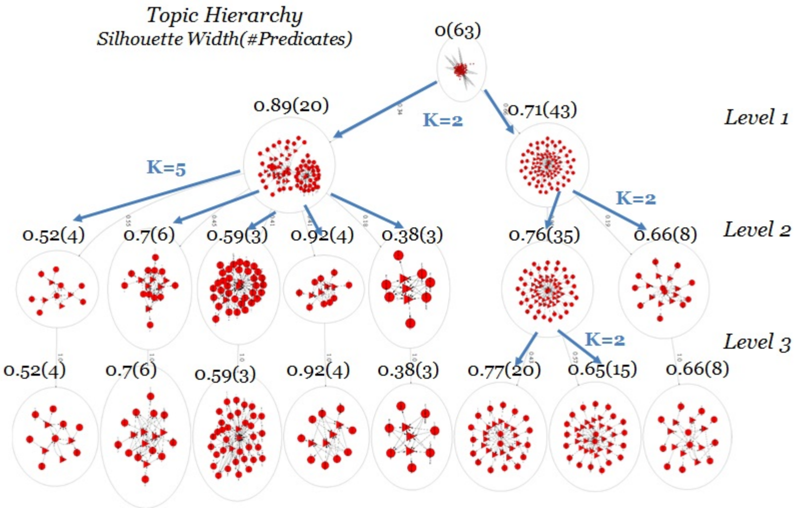


Figure 20: Silhouette Width and Number of Topics in Topic Hierarchy

two silhouette widths, 0.76 (for 35 predicates) and 0.66 (for 8 predicates) are computed for each topic. At level 3, one of the topics are partitioned into two (K= 2). Two silhouette widths, 0.77 (for 20 predicates) and 0.65 (for 15 predicates) are computed for each topic. If there are too many or too few topics, as may occur when a poor choice of k in each level is used in the hierarchical K-Means algorithm, some of the topics will typically display much narrower silhouettes than the rest. Thus silhouette averages are used to determine the number of topics within a dataset. In the HPKM, a topic of interest is further clustered into K subtopics (the optimal K subtopics) using a heuristic algorithm, Neighborhood Silhouette Width (NSW). NSW is similar to the silhouette method that validates the consistency checking by examining how well each predicate fits some uniformity criterion in its cluster, whereas Neighborhood Silhouette Width (NSW) is the average of the weighted SW for the (neighbored) topics at a specific level that have the same parents. The Neighborhood Silhouette Width (NSW) is computed by the sum of the multiplication of silhouette width and the number of predicates in a particular topic, $NumP(T_i)$, divided by the total number of predicates in the neighboring topics. The optimal k for a topic T_l at level l will be determined based on the highest Neighborhood Silhouette Width $nsw(T_l)$

$$sw(p_i) = \frac{\sum_{i=1}^{k} sw(T_i) * NumP(T_i)}{\sum_{i=1}^{k} NumP(T_i)} \tag{3.7}$$

For example, for the given $sw(T1\_1)$ is 0.89 and $NumP(T1\_1)$ is 20 and $sw(T1\_2)$ is 0.71 and $NumP(T1\_1)$ is 43, the first level's Neighborhood Silhouette Width $nsw(T1\_1)$ is computed as follows

$$nsw(T1\_1) = \frac{(0.89*20+0.71*43)}{(20+43)} = 0.77$$

Therefore, at level 1, the highest NSW value is 0.77 and the optimal K is determined as 2. Similarly, the second level's Neighborhood Silhouette Width $nsw(T2\_1)$ is computed as follows

$$nsw(T2\_1) = \frac{(0.52*4+0.7*6+0.59*3+0.92*4+0.38*3)}{(4+6+3+4+3)} = 0.64$$

$$nsw(T2\_2) = \frac{(0.76*35+0.66*8)}{(35+8)} = 0.74$$

Therefore, the highest NSW for $T2\_1$ and $T2\_2$ at level 2 is $T2\_1$ =0.64, $T2\_1$ =0.74 and the optimal K is determined as 5 and 2, respectively.

According to the optimal k determined by $nsw(T_l)$, the level of the hierarchy that can represent topics at multiple tasks will be constructed at different levels until there is no further change in the hierarchy. At the first level partition, we get an highest initial silhouette width s, which can be used as a global silhouette width threshold. For each of the m topics in the second level, we start doing second level partition again. If the highest silhouette width for each topic $s_m \geq$ s, we continue doing clustering on this specific branch. Here we name s_m as local silhouette width optimization threshold. Otherwise such topic is not able to be split any more. In addition, for any level n $>$ 2, to determine whether split a topic or not, we no longer compare the highest silhouette s_{nm} with global threshold s, but only compare s_{nm} with their own local optimization s_m. Similarly, if s_{nm} $\geq s_m$, clustering could be continued. We do a heuristic test for topics at all levels until there is no new topics been generated any more. In this way, we can achieve the HPKM objective of maximizing intra-cluster similarities and minimizing inter-cluster similarities. The algorithm of Hierarchical Predicate oriented K-Means clustering (HPKM) in terms of global and local optimization are given in Algorithm 1 and 2 respectively.

Algorithm 1 Hierarchical Predicate-based K-Means Clustering (HPKM) Part 1 with Global Optimization

```
/* P is an n * n predicate similarity matrix, n is the number of predicates in
   ontologies                                                              */
/* $\delta$ is the global silhouette width threshold $C_{ij}$              */
Input: P, $\delta$
Output: C={$C_{11}, C_{12} \ldots, C_{ij}$}
i=1
 while Change1==true and $sw_2 >= \delta$ do
    $sw_1$= $nsw(c_i)$
     k= OptimalK($C_i$, $sw_1$)
     Change1=false
     if $k > 1$ then
        Change1=true
    end
    for j = 1 to k do
        $\mu_{ij}$= RM($p_{j1}$,$p_{j2} \ldots, p_{jm}$)
    end
    for each $p_{ij}$ $\in P_i$ do
        $\mu_{ij}$ = $Argmin$(p$_{ij}$, $\mu_{ij}$), j $\in 1 \ldots k$
    end
    Go to Algorithm 2
end
```

Algorithm 2 Hierarchical Predicate-based K-Means Clustering (HPKM) Part 2 with Local Optimization

```
/* P is an n * n predicate similarity matrix, n is the number of predicates in
   ontologies                                                              */
/* $\lambda$ is the local silhouette width threshold $C_{ij}$              */
Input: P, $\lambda$
Output: C=$\{C_{11}, C_{12} \ldots, C_{ij}\}$
while Change2==true and and $sw_2 >= \lambda$ do
    for each $\mu_{ij}$ $\in U_i$ do
        UpdateCluster($\mu_{ij}$)
    end
    for each $p_{ij}$ $\in P_i$ do
        $NCen = Argmin($p$_{ij}$, $\mu_{ij}$), j $\in 1 \ldots k$ if NCen $\neq$mu$_{ij}$ then
            $\mu_{ij}$=NCen $C_{ij}$=$C_{ij}$ $\cup$p$_{ij}$
            changed2=true
        end
        $sw_2$ = SilhouetteWidth($C_{ij}$)
    end
end
```

Algorithm 1 describes the situation when the first level clustering happens. We first calculate and pick the highest silhouette width and make it as the global optimization threshold δ. In Algorithm 2, we then consider each branch individually and make local optimization threshold λ to determine the stop point for each topic specifically. Both algorithms stop running when there is no more new topics are generated.

3.5 Predicate Oriented Hierarchical Agglomerative Clustering

There are various different approaches in clustering information networks. As we discuss in last section, one solution is Hierarchical Predicate oriented K-Means clustering (HPKM) algorithm for discovery of relevant topics from integrated multiple sources and forms a topic hierarchy. The HPKM algorithm is an excellent way to summarize a integrated view of multiple datasets. For example, Fig. 21 presents a clustering solution provided by HPKM on 9 Bio2RDF cross domains datasets. However, we observe that HPKM is not suitable for cross domains knowledge discovery from heterogeneous information network. The reason is that the HPKM's top-down approach focuses on global clustering based on homogeneous perspectives, however, ignoring the diverse and local perspectives of the network.

In this research, to handle heterogeneous datasets, we design a new algorithm, called the Predicate oriented Hierarchical Agglomerative Clustering (PHAL), for topic discovery from the heterogeneous information network of the multiple ontologies. PHAL is a hierarchical bottom-up clustering algorithm by applying Hierarchical Agglomerative clustering (HAC) [62] to the heterogeneous information network of cross domains ontologies. PHAL starts with each predicate as a singleton cluster and then successively merge pairs of clusters while traversing up through its ancestors in the hierarchy. To find better cross domains patterns, we use a approach which described in Definition 13 to make cross domains predicates relationship outstanding.

Fig. 22 shows a topic hierarchy generated from the PHAL algorithm based on Bio2RDF nine domains data. The PHAL algorithm has four phases as shown below and

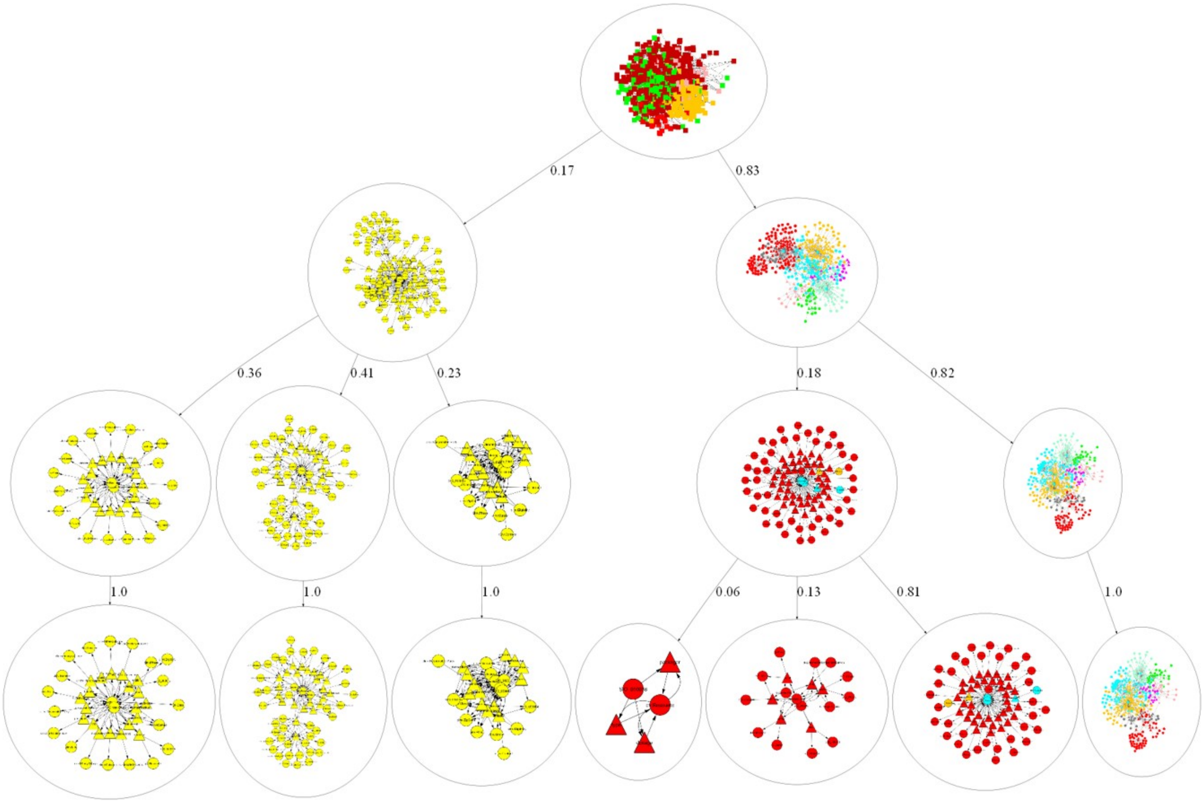


Figure 21: Top-Down Topic Hierarchy

top-down topic hierarchy with three levels and 7 topics at the third level. The number assigned to the edges indicates the distribution of predicates to its child node. The sum of the numbers should be one (e.g., 0.17+0.83 at the top level). A color is assigned to each ontology as follows: DrugBank: Red; HGNC: Pink; MGI: Green; PharmGKB: Cyan; ClinicalTrials: Yellow; OMIM: Sky Blue; SIDER: Gray; KEGG: Orange; CTD: Magenta.

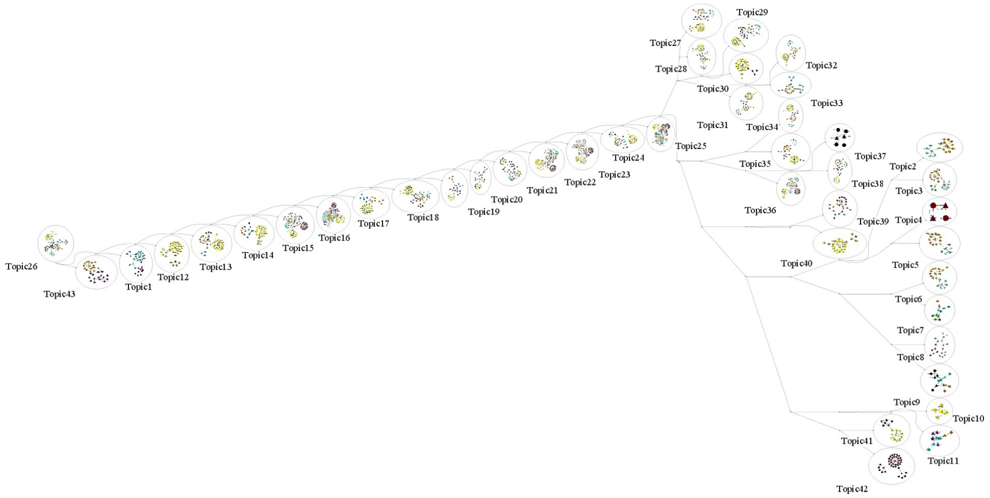


Figure 22: Bottom-Up Topic Hierarchy

Bottom-up topic hierarchy with 43 topics. Topic ID is assigned to each cluster in this hierarchy. A color is assigned to each ontology as follows: DrugBank: Red; HGNC: Pink; MGI: Green; PharmGKB: Cyan; ClinicalTrials: Yellow; OMIM: Sky Blue; SIDER: Gray; KEGG: Orange; CTD: Magenta.

the pseudo codes are shown in Algorithms 3, 4, 5 and 6.

Phase 1: This phase focuses on clustering predicates from the heterogeneous information network of the given ontologies using Hierarchical Agglomerative Clustering [62]. This algorithm is an incremental and interactive bottom-up approach to build a hierarchy of topics based on the PONP until all predicates in the network belong to a topic group. The results from this learning process are a set of *topics* (*InitialMap*) in a hierarchical tree structure (similar to the topics shown in Fig. 22).

Phase 2: Given the tree from Phase 1, we first compute the mid-level of the tree (i.e., *Mid* = $H/2$, where H is the height of the hierarchy generated from Phase 1). The topics at the mid-level *Mid* are assigned to *InitialTopicSet*. If there is no topic at the level *Mid*, then go upward until find any topic groups on the subsequent level of the *Mid* (i.e., *Mid*-1) in the hierarchy. Among 43 topics shown in Fig. 22, Topics 2-11 are the topic groups captured at the level *Mid*.

Phase 3: This phase illustrates the constructing of topics for the left-over topic groups (called the 'disjoint' topics), which do not belong to the topic groups *InitialMap*. Starting from the level *Mid* - 1, we start traversing the tree upward to construct a new topic group with each topic at the the subsequent level of the *Mid* level (i.e., *Mid* - 1) and assign it to *FinalTopicSet*. Repeat this step at *Mid* - 2 until reaching to the root. Topics 12-43 in Fig. 22 are newly constructed during this phase. In addition, we have made a special topic group (i.e., $Topic_1$) that is a collection of the singleton topics whose size is 1.

Phase 4: There are some cases such that relevant concepts are disconnected. This is due to the hard partition in which a predicate is not allowed to join more than one topic. To handle the issue, a refinement process is conducted to construct a more complete topic model with the respective predicates and their neighborhood. More precisely, for any two pairs of predicates, if they form a Connectivity pattern and then we include their intermediate predicates to the topic and update those topics in *FinalTopicSet*. From this refinement process, a predicate may join more than one topic group that results into fuzzy clustering.

Algorithm 3 Hierarchical Heterogeneous Clustering

```
Input: $X$ ={$x_1, \dots x_n$}
Output: Topic Set $T = \{t_1, \dots, t_k\}$
/* Phase 1: Hierarchical agglomerative clustering */
Define level $L = 0$
 Consider each element in $X$ as a topic, save them in InitialMap with level $L = 0$
 Put pair $\langle L, X \rangle$ to InitialMap
 while true do
     if all objects belong to one topic then
         break
     else
         In current level $L$, extract all topics from InitialMap
           Calculate the minimum average distance for any two topics $p$ and $q$ with formula
           $\frac{1}{|p|*|q|} \sum_{m \in p} \sum_{n \in q} d(m, n)$
           Save all pairs to set $M$
           $L = L + 1$
           for each pair of topic $p$ and $q$ in set $M$ do
               merge $p$ and $q$ into a new topic $u$
                 put $u$ to set $Y$
                 put $\langle L, Y \rangle$ to InitialMap
         end
     end
end
```

Algorithm 4 Initial Topic Groups at Level Mid

```
Input: $X$ =$\{x_1, \ldots x_n\}$
Output: Topic Set $T = \{t_1, \ldots, t_k\}$
/* Phase 2: Initial Topic Groups at Level Mid */
Define the height of the tree $H$ = $L$ ;    // $H$ is the height of the tree
 from Phase 1
Get the middle level Mid = Roundup $(L/2)$ while true do
    if InitialMap has any topic at level Mid then
        extract $Y$ at level Mid ; // by checking ⟨Mid, Y⟩ from InitialMap
        InitialTopicSet = $Y$
         reak
    else
        Mid = Mid-1
    end
end
```

Algorithm 5 Disjoint Topic Construction

```
Input: $X$ ={$x_1, \dots x_n$}
Output: Topic Set $T = \{t_1, \dots, t_k\}$
/* Phase 3: Disjoint Topic Construction */
Define FinalTopicSet = InitialTopicSet
 Define set $Z$ that contains all the other disjoint topics
 Initialization of topic index = 2 ;    // excluding the special topic $T_1$
for each element in InitialTopicSet do
    for each topic $z_i$ in $Z$ do
        if $z_i$.size=1 then
            Add $z_i$ to the special topic $Topic_1$
             Update the special topic $Topic_1$ in FinalTopicSet
        else
            Add $z_i$ to $Topic_{index}$
             Add $Topic_{index}$ to FinalTopicSet
             index++
        end
    end
end
return FinalTopicSet
```

```
Algorithm 6 Hierarchical Topic Refinement
Input: FinalTopicSet =$\{t_1, \ldots, t_k\}$
Output: FinalTopicSet =$\{t'_1, \ldots, t'_k\}$ ;          // refined topics with new
        predicates
/* Phase 4: Topic Refinement                                                  */
for each topic $t$ in FinalTopicSet do
    for any two predicates $p_i$ and $p_j$ in topic $t$ do
        if $p_i$ and $p_j$ are connected through a Connectivity pattern & d($p_i$,$p_j$) = 2 then
            find the intermediate predicate $p_t$ between $p_i$ and $p_j$
            add predicate $p_t$ to topic $t$
        end
        if $p_i$ and $p_j$ are connected through a Connectivity pattern & d($p_i$,$p_j$) = 3 then
            find the two intermediate predicates $p_m$ and $p_n$ between $p_i$ and $p_j$
            add predicate $p_m$ to topic $t$
            add predicate $p_n$ to topic $t$
        end
    end
end
```

Theorem 4 shows that both HPKM and PHAL don't miss any predicates after clustering the original graph.

Theorem 4. Assume $\{PR\}$ contains all predicates in RDF graph G, $\{PR_{T1}\}$, $\{PR_{T2}\}$, . . ., $\{PR_{Tk}\}$ represent predicates maintained in topic T_1, T_2, . . ., T_k generated by HPKM or PHAL. so that:

$$\{PR_{T1}\} \cup \{PR_{T2}\} \cup \ldots \cup \{PR_{Tk}\} = \{PR\}$$

Proof. For each level of HPKM, the algorithm is $\arg\min_S \sum_{i=1}^{k} \sum_{x \in S_i} |x - \mu_i|^2$, where S represents all topics so that S = $\{S_1, S_2, \ldots, S_k\}$, k indicates number of topics, x indicates predicates $(x_1,\ x_2,\ \ldots,\ x_n)$, μ_i indicates mean of points in each cluster. Because this algorithm runs recursively for each predicate x and cluster them into k topics, therefore, predicates $(x_1, x_2, \ldots, x_n)$ are divided into topics $\{S_1, S_2, \ldots, S_k\}$ without any missing. For PHAL, the algorithm is $\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$, where A and B maintain any two topics of predicates and d(a,b) indicates the distance between predicate a and b. Because this algorithm runs recursively until all the predicate $a \in A$ and $b \in B$ are visited, so there is no missing of predicates. Therefore, $\{PR_{T1}\} \cup \{PR_{T2}\} \cup \ldots \cup \{PR_{Tk}\}$ = $\{PR\}$

Based on Theorem 4, we also give Lemma 2 to demonstrate that the union result of topics make the original graph.

Lemma 2. Assume $\{G\}$ is a RDF graph, $\{G_{T1}\}$, $\{G_{T2}\}$, ..., $\{G_{Tk}\}$ represent graph maintained in Topic $T_1, T_2, \ldots, T_k$ generated by HPKM or PHAL. so that:

$$\{G_{T1}\} \cup \{G_{T2}\} \cup \ldots \cup \{G_{Tk}\} = \{G\}$$

Proof. In $\{PR_{Tk}\}$, $\forall$ predicates P, it links to subjects and objects and form graph $\{G_{Tk}\}$ so that $\{PR_{Tk}\} \subsetneqq \{G_{Tk}\}$. Based on Theorem 4, because $\{PR_{T1}\} \cup \{PR_{T2}\} \cup \ldots \cup \{PR_{Tk}\}$ = $\{PR\}$, therefore, $\{G_{T1}\} \cup \{G_{T2}\} \cup \ldots \cup \{G_{Tk}\}$ = $\{G\}$

3.6 Evaluation and Results

In this section, we conduct a comprehensive evaluation on GraphKDD framework with PONP, HPKM and PHAL approaches. We use different datasets and use cases to validate the processing results.

Table 15: Data Statistics

Domain	Dataset	#Concepts	#Predicates	#Triples (Schema)	#Triples (Instance)
Bio2RDF	PharmGKB	60	48	386	278,049,209
	DrugBank	92	63	728	3,672,531
	KEGG	61	72	457	50,197,150
	CTD	19	14	141	326,720,894
	HGNC	16	14	84	3,628,205
	OMIM	30	35	253	8,750,774
	SGD	83	33	792	12,494,945
	Sider	14	15	126	17,627,864
	Affymetrix	33	20	190	86,942,371
	Irefindex	74	15	1478	48,781,511
	Biomodel	54	12	472	2,380,009
	GO	9	12	67	97,520,151
	MGI	20	13	132	8,206,813
Web	DBpedia	168	943	943	3,000,000,000
	YAGO	44	119	126	120,000,000

3.6.1 Data Specification

We introduce different datasets from different domain for the purpose of evaluation. Specifically, we include computer science datasets DBpedia and YAGO for the purpose of single domain datasets evaluation and imported 13 datasets from linked life science datasets Bio2RDF for cross domains validation. Table 15 gives the detailed statistics for each dataset.

3.6.2 Single Domain Analysis

In this section, we introduce various case studies on three different datasets: DrugBank, DBpedia and YAGO. We give results and evaluation for each case to validate HPKM approach.

Table 16: DrugBank Ontology

Features	Num
Num of Total Concepts	116
Sum of in-degree and out-degree ($\|E\|$)	519
Num of Unique Concepts in DrugBank (C)	93
Num of Triples	737
Num of Total Predicates	68
Num of Domain Specific Triplets (T)	401
Num of Unique Domain Specific Predicates (P)	63
Density (D)	0.043

3.6.2.1 DrugBank Case Study

For DrugBank case study, we demonstrate topic hierarchy, statistics of top predicates and concepts, validation of Hierarchical K-Means Clustering (HPKM), topic generation output and topic discovery and query generation results.

Topic Hierarchy Generated using HPKM Approach

In this case study, we demonstrate the details of knowledge discovery as well as query generation in the proposed framework. We are particularly interested in generating interesting queries using the proposed PONP model and HPKM algorithms. In addition, the experiments have been conducted to validate the correctness of our approach. Table 16 shows the details of the DrugBank Ontology. In this case study, the unique concepts (C) of DrugBank ontology, excluding the duplicates, are considered. Only the domain specific predicates (P) excluding built-in predicates are considered. The number of edges in the graph ($|E|$) is computed as the sum of in-degree and out-degree. The overall density is computed based on the vertices (P+C) and the edges (E).

Table 17: Short Notation for DrugBank and Related Domains

prefix	**Domain URL**
ahv:	http://bio2rdf.org/ahfs_vocabulary:
av:	http://bio2rdf.org/atc_vocabulary:
bv:	http://bio2rdf.org/bindingdb_vocabulary:
cv:	http://bio2rdf.org/chemspider_vocabulary:
dv:	http://bio2rdf.org/drugbank_vocabulary:
dpv:	http://bio2rdf.org/dpd_vocabulary:
gv:	http://bio2rdf.org/genbank_vocabulary:
gav:	http://bio2rdf.org/genatlas_vocabulary:
gcv:	http://bio2rdf.org/genecards_vocabulary:
giv:	http://bio2rdf.org/gi_vocabulary:
gtv:	http://bio2rdf.org/gtp_vocabulary:
hv:	http://bio2rdf.org/hgnc_vocabulary:
iv:	http://bio2rdf.org/iuphar_vocabulary:
kv:	http://bio2rdf.org/kegg_vocabulary:
owl:	http://www.w3.org/2002/07/owl#
pcv:	http://bio2rdf.org/pubchem.compound_vocabulary:
pdv:	http://bio2rdf.org/pdb_vocabulary:
psv:	http://bio2rdf.org/pubchem.substance_vocabulary:
pv:	http://bio2rdf.org/pubmed_vocabulary:
uv:	http://bio2rdf.org/uspto_vocabulary:
chv:	http://bio2rdf.org/chebi_vocabulary:
nv:	http://bio2rdf.org/ndc_vocabulary:
phv:	http://bio2rdf.org/pharmgkb_vocabulary:
unv:	http://bio2rdf.org/uniprot_vocabulary:

The base URL of predicates is $http : //bio2rdf.org/drugbank\_vocabulary$. However, the concepts are from 24 different domains as shown in Table 17. Interestingly, all predicates are from the same domain and that gives us a good basis for linking concepts together either from same or different domains. This is one of the reasons we propose a predicate oriented approach. The concepts' domain URLs and their short notations are shown in Table 17. From the HPKM algorithm for each domain ontology, the topic hierarchy is generated. Fig. 23 shows the topic hierarchy generated for a single domain ontology, DrugBank. As seen in Fig. 23, DrugBank has the number of topics (2:7:8) with 2 topics at the first level, 7 topics at the second level, and 8 topics at the third level. K-Means clustering is performed in a top-down manner until both global and local clusters' silhouette width optimization are not reached. The number on each edge in the topic hierarchy represents the percentage of predicates that the upper level topic graph contributes to the lower level graph. For example, for the two topics in the first level of DrugBank, 66% of predicates of the DrugBank ontology are contributed to Topic 1 (T1_1) while 34% to Topic 2 (T1_2). The contribution rate is ranged between 0 and 1. Interestingly, predicates are unique to their topic graph, however, some concepts in a topic may appear in more than one topics. Moreover, for each topic at 3rd level, top 2 ranked predicates (computed based on in-degree/out-degree) are selected as a representative term for each topic.

Top Predicates and Concepts of DrugBank

Table 18 shows the ranks for Top 20 predicates and Top 20 concepts that are computed in terms of the sum of their in-degree and out-degree. These predicates and concepts are shown in terms of Predicate Rank (PR), Predicates, Predicate IO (PIO) and Predicate Topic ID (PIO), corresponding Concepts, Concept Rank (CR), and together with the description of predicates specified by DrugBank. From this list, many top predicates are from Topic 3_6, Topic 3_7, and Topic 3_1. Many top concepts are from Topic 3_7, Topic 3_2, and Topic 3_3. The prefix dv: of these concepts indicates the domain http://bio2rdf.org/drugbank_vocabulary. Some of the Top 20 Concepts are not directly

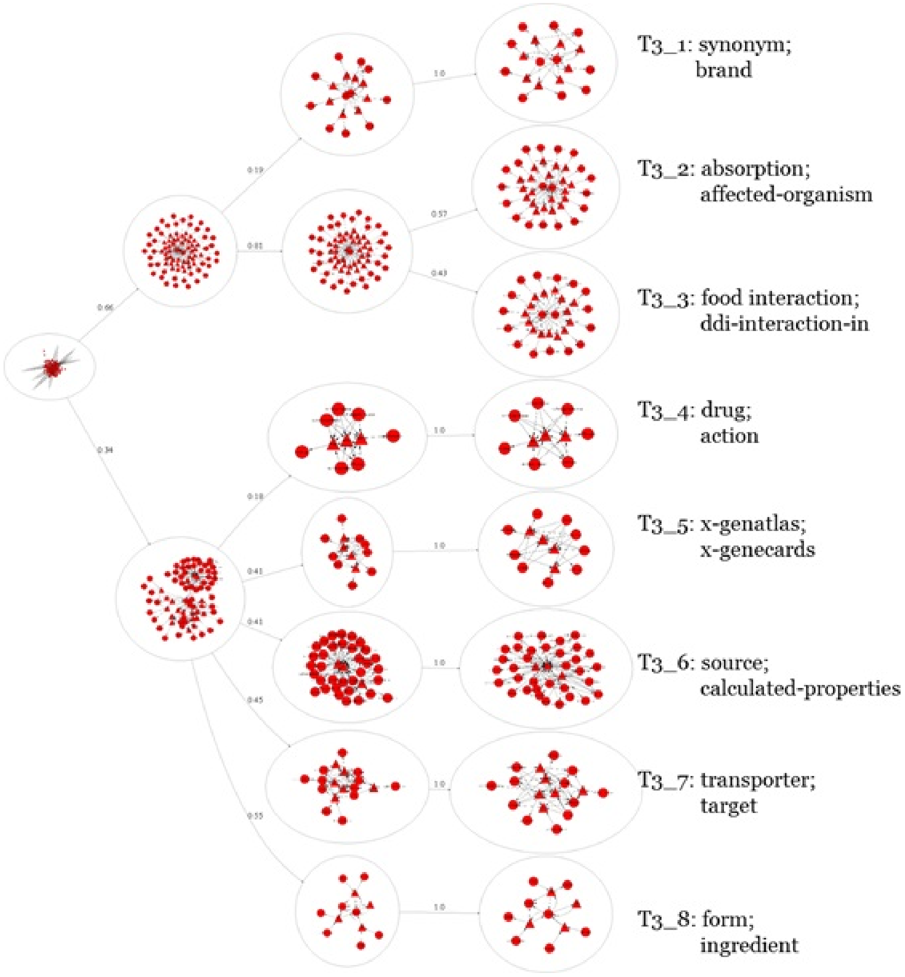


Figure 23: DrugBank Topic Hierarchy.

Table 18: Top 20 Predicates and Concepts in DrugBank Ontology

PR	Predicates	PIO	TID	CR	Concepts
1	source	66	T3_6	46	dv:Source
2	calculated-properties	56	T3_6	N/A	N/A
3	experimental-properties	28	T3_6	N/A	N/A
4	transporter	17	T3_7	7	dv:Transporter
5	target	16	T3_7	6	dv:Target
6	drug	14	T3_4	2	dv:Drug
7	enzyme	13	T3_7	5	dv:Enzyme
8	carrier	12	T3_7	4	dv:Carrier
9	action	11	T3_4	N/A	N/A
10	synonym	10	T3_1	21	dv:Synonym
11	brand	9	T3_1	51	dv:Brand
12	category	8	T3_1	21	dv:Category
13	form	8	T3_8	N/A	N/A
14	ingredient	8	T3_8	N/A	N/A
15	x-genbank	7	T3_7	N/A	N/A
16	x-uniprot	7	T3_7	N/A	N/A
17	manufacturer	6	T3_1	51	dv:Manufacturer
18	mixture	6	T3_1	45	dv:Mixture
19	toxicity	6	T3_1	51	dv:Toxicity
20	absorption	6	T3_2	51	dv:Absorption

mapped with the predicates in the Top 20 Predicates. These concepts are dv:Enzyme-Relation, dv:Target-Relation, dv:Carrier-Relation, dv:LogP, dv:LogS, dv:Molecular-Formula, dv:Molecular-Weight, dv:Transporter-Relation, dv:Water-Solubility, dv:Bioavailability, dv:Boiling-Point, dv:Caco2-Permeability. These results show that the predicates rankings are not always the same with the concept rankings.

Table 19 shows the duplicated concepts among topics. The total number of instances is 40 and the number of duplicates is 23. dv:Resource and dv:Drug appear in almost all the topics. According to this analysis, the sets of the topic groups T3_1 and

Table 19: Duplicated Concepts and their Topic ID in DrugBank Ontology

Concepts	**Freq**	**Topics**
dv:Resource	8	T3_1, T3_2, T3_3, T3_4, T3_5, T3_6, T3_7, T3_8
dv:Drug	6	T3_1, T3_2, T3_3, T3_4, T3_6, T3_7
uv:Resource	2	T3_1, T3_8
dv:Mixture	2	T3_1, T3_8
dv:Patent	2	T3_1, T3_8
dv:Pharmaceutical	2	T3_2, T3_8
dv:Carrier-Relation	2	T3_4, T3_7
dv:Target-Relation	2	T3_4, T3_7
dv:Transporter-Relation	2	T3_4, T3_7
dv:Enzyme-Relation	2	T3_4, T3_7
dv:Carrier	2	T3_5, T3_7
dv:Target	2	T3_5, T3_7
dv:Enzyme	2	T3_5, T3_7
dv:Transporter	2	T3_5, T3_7
uv:Resource	2	T3_1, T3_8
Total	23	

T3_8, T3_4 and T3_7, and T3_5 and T3_7 are similar. However, these are quite different from the outcomes from the predicated-oriented clustering algorithm.

Validation for Hierarchical K-Means Clustering

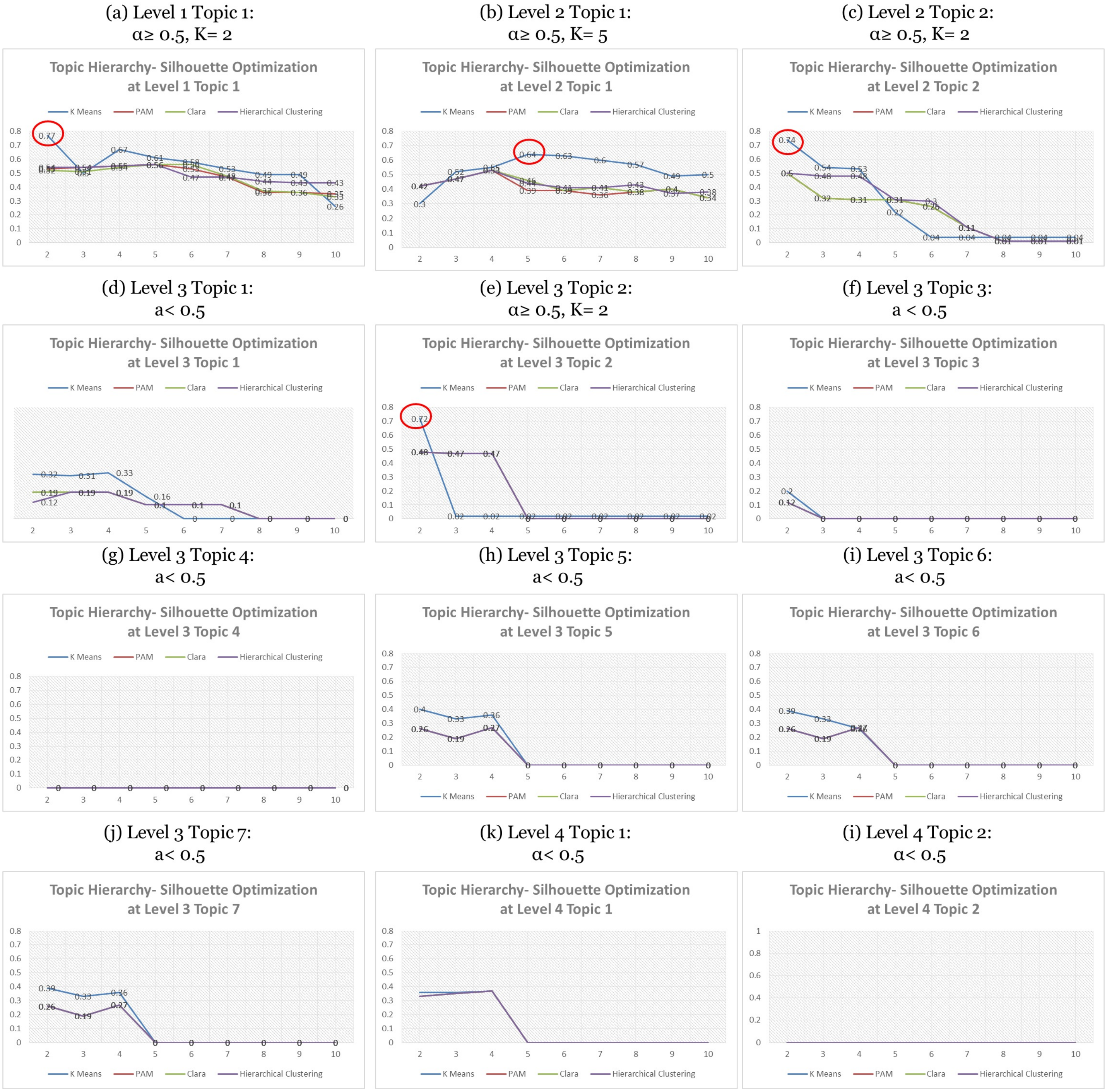
An experiment has been conducted to find an optimal number of the clusters using the four different clustering algorithms, K-Means [49], Clara [63], Pam [120], and Hierarchical Clustering [62]. Fig. 24 shows the results of the optimal K validation algorithm based on the clustering outcomes by the four different algorithms. As a result, Clara, Pam and Hierarchical clustering algorithms are not a good approach to find an optimal cluster number since they show a relative stable silhouette width for varying the number of clusters. The proposed HPKM algorithm determines the most significant number of clusters at each level such as K = 2 with SW = 0.77 at level 1 and K = 5 with SW = 0.64 and K=2 with SW = 0.74 at level 2 and K = 2 with SW = 0.72 at level 3. The HPKM algorithm is validated and compared against other algorithms in terms of the cluster number and the silhouette width.

Results for Topic Generation

For the DrugBank ontology, we've considered 63 concepts, 116 predicates. Figure 25 shows relevance scales of five different rankings and an overall ranking. The overall rank is computed in terms of the following 5 criteria: i) Top 20 Concepts, ii) Top 20 Predicates, iii) Similarity, iv) Silhouette Width, v) Density. Eight topics ranked from best to worst as follows: Topic 3_4, Topic 3_7, Topic 3_6, Topic 3_2, Topic 3_1, Topic 3_3, Topic 3_8, and Topic 3_5. Specifically, Topic 3_4 shows the best ranking for all three criteria such as Similarity, Silhouette Width and Density. However, Topic 3_7's Top 20 Concept Ranking, Top 20 Property Ranking, and Similarity Ranking are relatively good. From the ranking results, we have observed that the proposed ranking system correctly captured Topic 3_4 and Topic 3_7 as the core topics of DrugBank. Topic 3_5 is ranked the worst among the eight topics. Since Topic 3_5 is a connector topic whose predicates are mainly used to connect DrugBank with other domains. It is relatively less important from a single domain (DrugBank) perspective. However, Topic 3_5 would be very useful from a cross domains perspective.

3.6.2.2 DBpedia and YAGO Case Study

In this section, we apply the GraphKDD framework on DBpedia and YAGO datasets respectively to evaluate the single domain analysis with data in computer science


(a) Level 1 Topic 1:
α≥ 0.5, K= 2

(b) Level 2 Topic 1:
α≥ 0.5, K= 5

(c) Level 2 Topic 2:
α≥ 0.5, K= 2

(d) Level 3 Topic 1:
a< 0.5

(e) Level 3 Topic 2:
α≥ 0.5, K= 2

(f) Level 3 Topic 3:
a < 0.5

(g) Level 3 Topic 4:
a< 0.5

(h) Level 3 Topic 5:
a< 0.5

(i) Level 3 Topic 6:
a< 0.5

(j) Level 3 Topic 7:
a< 0.5

(k) Level 4 Topic 1:
α< 0.5

(i) Level 4 Topic 2:
α< 0.5

Figure 24: Optimal K Validation using Multiple Clustering Techniques.

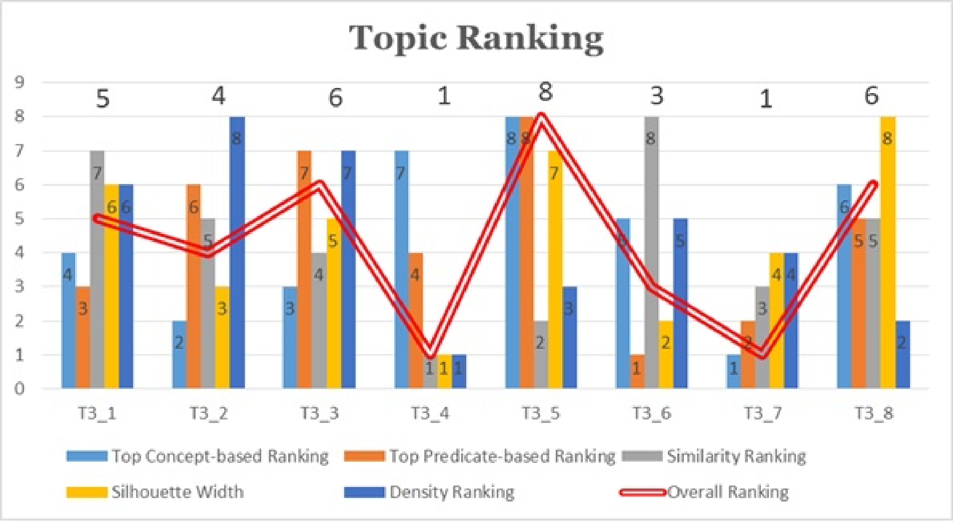


Figure 25: Topic Rankings for DrugBank.

domain. We conduct experiments to validate the optimal radius boundary and clustering partition point. In addition, we conduct comparison evaluations on the following specific aspects to validate the better performance of our predicated oriented HPKM approach: 1) topic ranking in terms of in-degree/out-degree, page ranking, topic size and Entropy; 2) predicate oriented approach with entity oriented solution; 3) HPKM with random graph partition approach; 4) Topic generation with LDA algorithm.

Optimal Predicate Neighbourhood Radius Boundary

In this research, we refer predicate neighbourhood radius boundary as the number of neighbour we considered for each predicate. We evaluate the optimal radius boundary for two datasets based on silhouette width, topic number and topic size. In each case, we apply a heuristic approach on radius varying from 1 to 5 and try to find at which point we can get the optimal silhouette width, topic number and topic size. Fig. 26 shows the

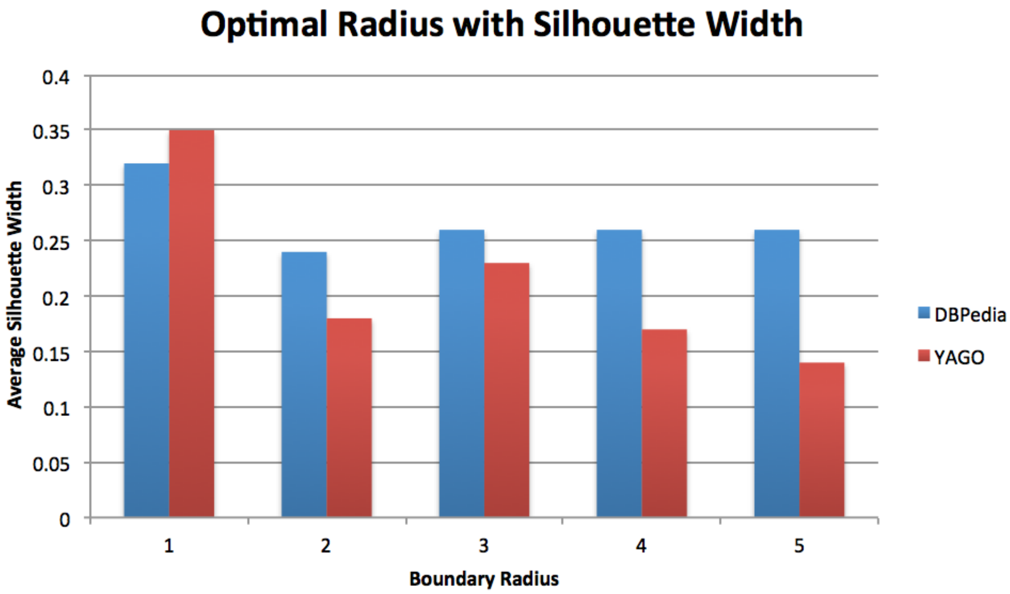


Figure 26: Optimal Radius with Silhouette Width

relationship between radius and silhouette width. Both datasets got the highest silhouette width when radius at 1. However, with the increasing of radius, silhouette width doesn't increase. Especially for YAGO, silhouette width even dropped when radius is larger than 3. Although radius at 1 gave the best silhouette width, we'd like to find more than one neighbourhood to expand predicates' relationship network, that is why we choose radius at 3 as the optimal one. Fig. 27 gives the association between boundary radius with number of topic. DBpedia got its maximum number of topic when radius is 2 while YAGO got its biggest number of topic when radius is 3. For both datasets, number of topic doesn't increase even if we increase the radius. From this evaluation, we concluded that radius at 3 is the best for both datasets. Similarly, as Fig. 28 shows, radius at 3 gives the optimal size of topic for both datasets.

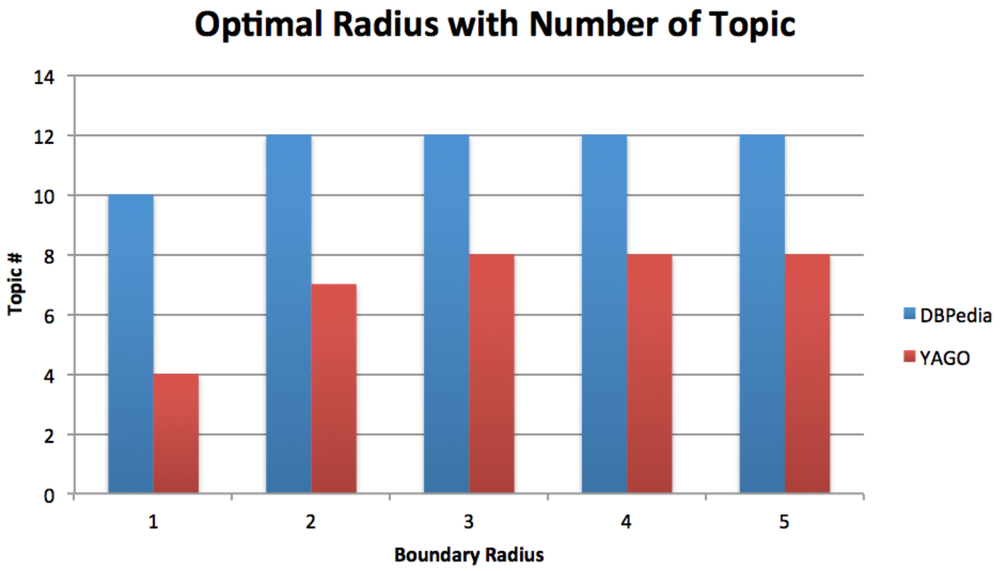


Figure 27: Optimal Radius with Number of Topic

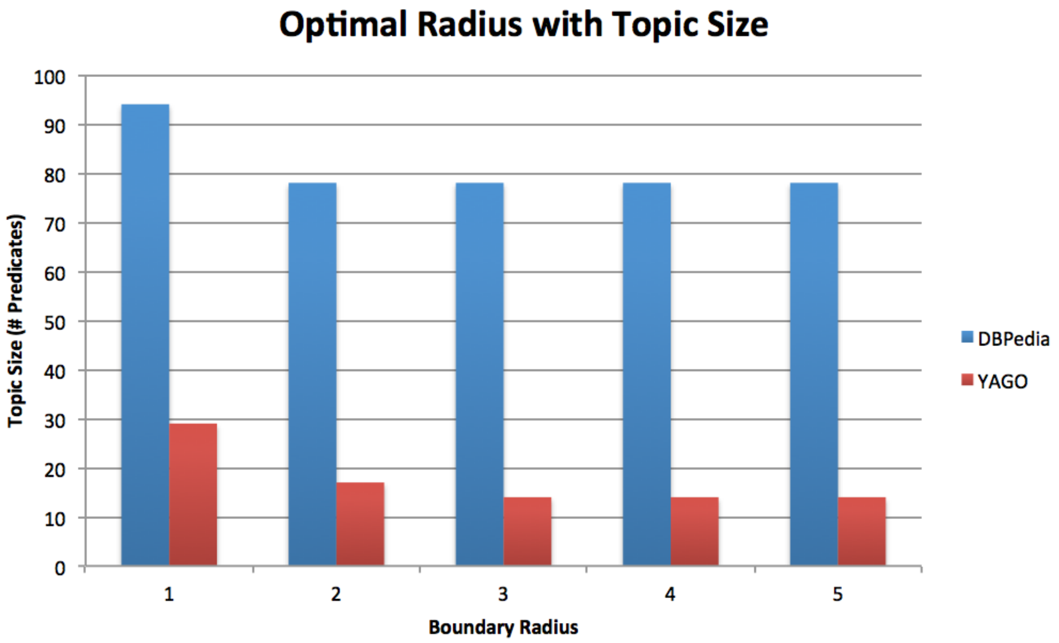


Figure 28: Optimal Radius with Topic Size

Table 20: Clustering Comparison

Clustering Algorithms
HPKM
Fuzzy C-means
Pam
Clara
Hierarchical Clustering

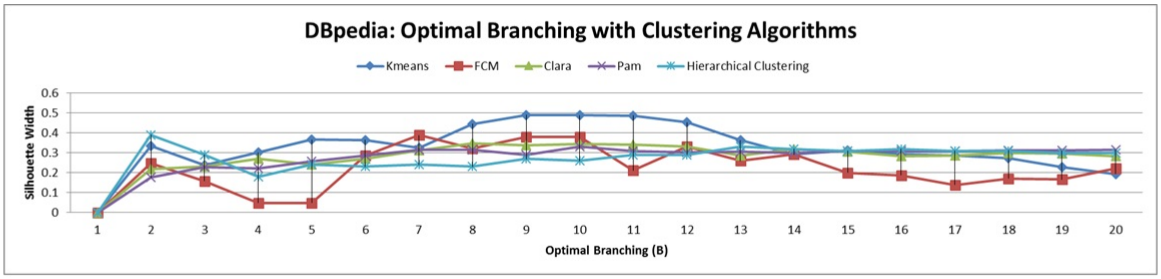


Figure 29: DBpedia:Optimal Branching with Clustering Algorithms

Optimal HPKM Branching Factor

To validate the optimal solution provided by HPKM algorithm, we compare HPKM with several clustering methods listed in Table 20. In addition, to decide the optimal branching factor for each algorithm, we conduct a heuristic approach by varying the number of branching factor from 1-20. Fig. 29 shows the optimal first level branching factor evaluation for DBpedia dataset. HPKM gets the highest silhouette width than other approaches when branching factor is 9. Similarly, as Fig. 30 represents, the optimal first level branching factor for YAGO dataset is 7. Therefore, GraphKDD divides DBpedia and YAGO as 9 clusters and 7 clusters at first level respectively. We apply the same strategy for each level until it meets the stop criteria.

As a result, the hierarchical topic visualization for DBpedia and YAGO is shown

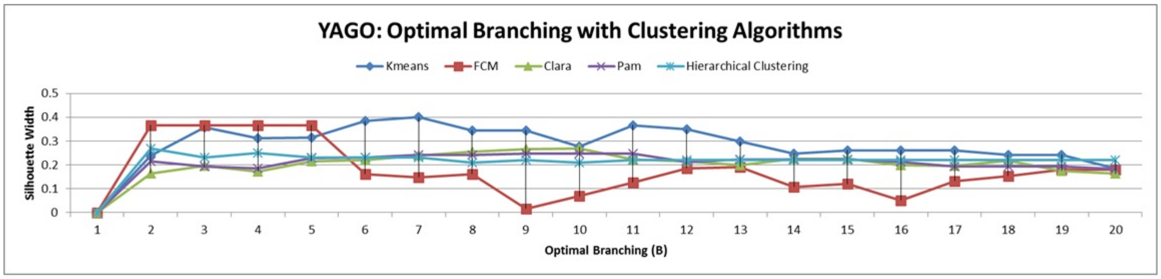


Figure 30: YAGO:Optimal Branching with Clustering Algorithms

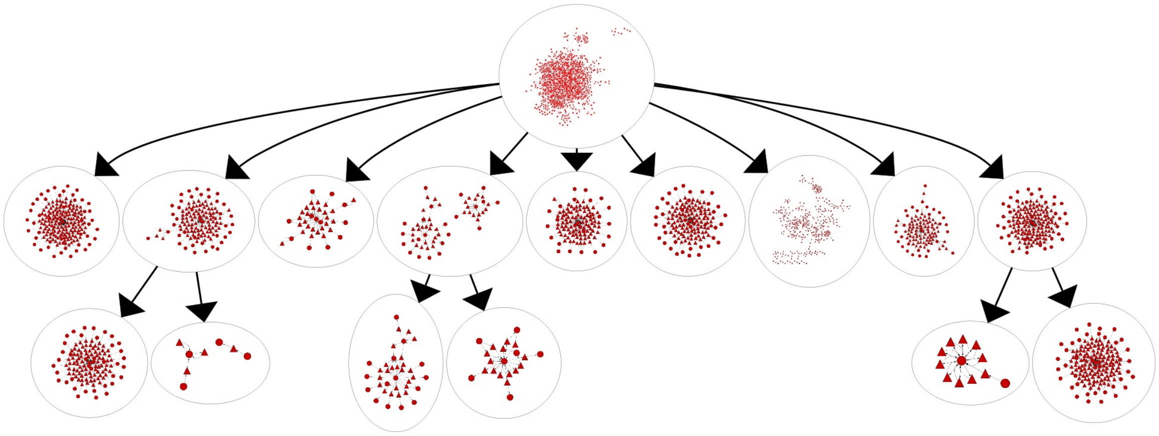


Figure 31: DBpedia Topic Hierarchy

as Fig. 31 and 32 respectively. DBpedia has three level with 9 topics in second level and 12 topics in third level. YAGO has three level in total with 7 topics in second level and 8 topics in third level.

Topic Ranking

For each topic, we first summarize the total number of predicates and concepts for DBpedia and YAGO as shown in Fig. 33 and 34 respectively. We then use different

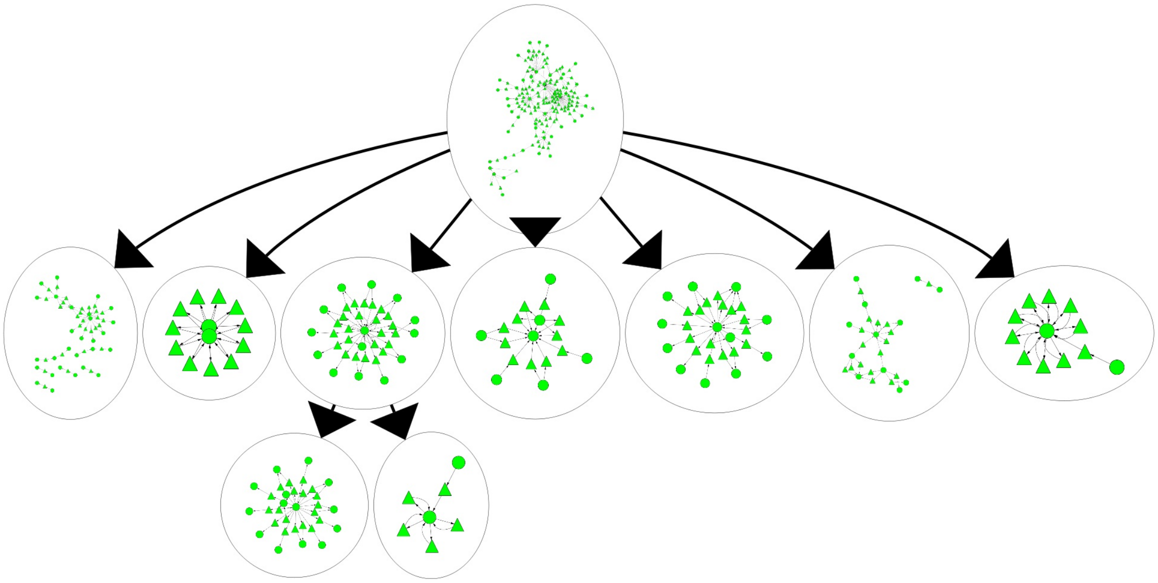


Figure 32: YAGO Topic Hierarchy

measurements to quantify the importance of different topics for both datasets. Summation of in-degree and out-degree (I/O) captures how many entities directed connected to each predicates for each topic. Page rank describes the significant predicates for each topic by counting the number of predecessor entities that directly connected. Size of topic illustrates how many predicates are involved in each topic. Entropy explains the expected value of the information contained in each topic. Ranking detailed for DBpedia and YAGO are shown in Fig. 35 and 36 respectively. From Fig. 35 we conclude that for DBpedia, Topic 9 is ranked as top 1 for each criteria, which is consistent with the maximum number of predicates and concepts shown in Fig. 33. From this perspective, we find that with a bigger number of predicates and concepts, a topic is more likely to have higher I/O, page rank, topic size and entropy. We get the same conclusion for YAGO, of which Topic 1 has the most predicates and concepts as well as highest ranking.

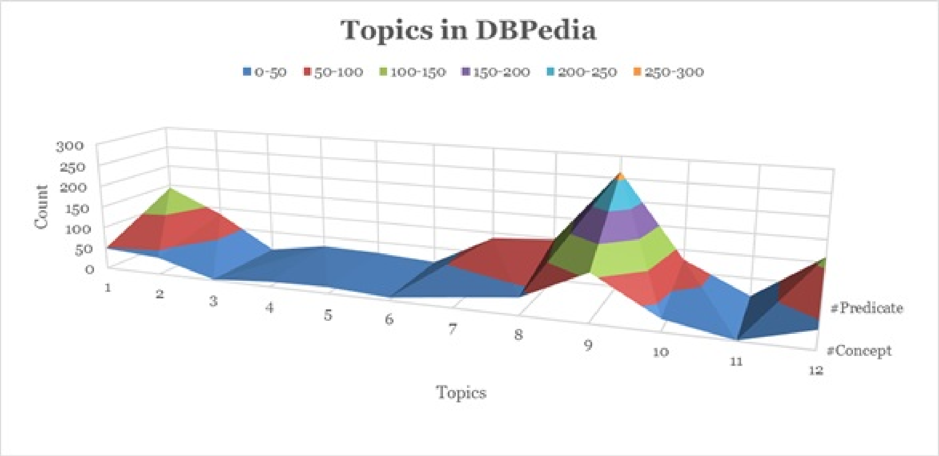


Figure 33: DBpedia Topic Summary

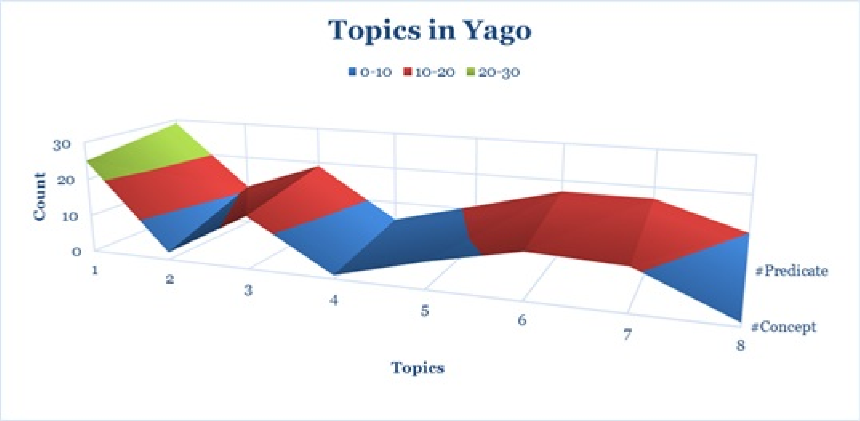


Figure 34: YAGO Topic Summary

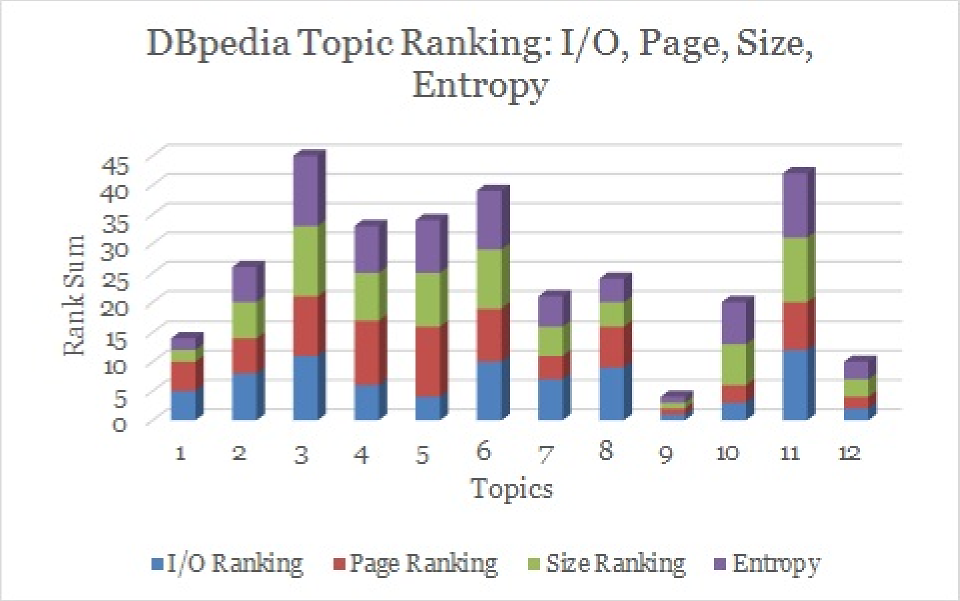


Figure 35: DBpedia Topic Ranking

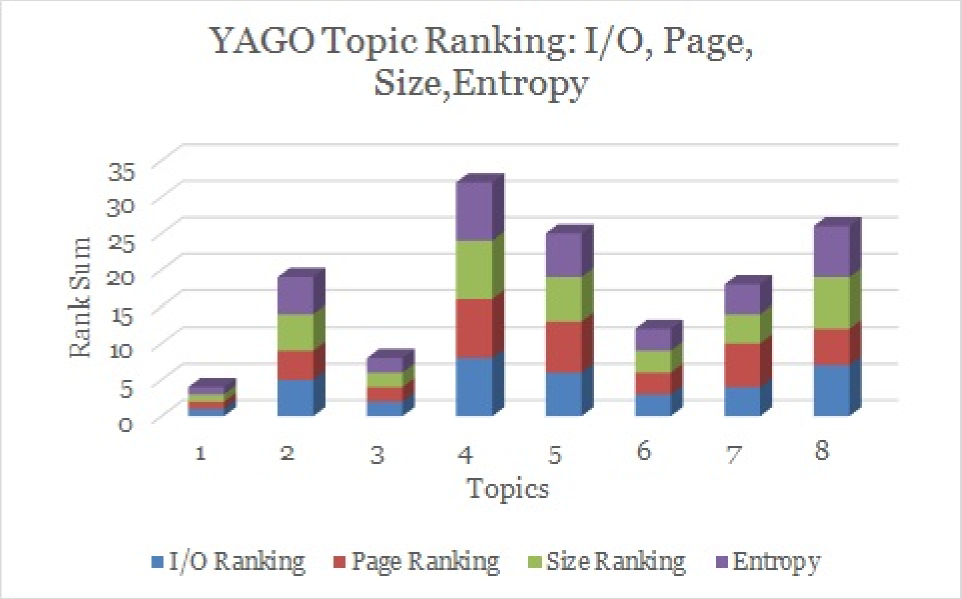


Figure 36: YAGO Topic Ranking

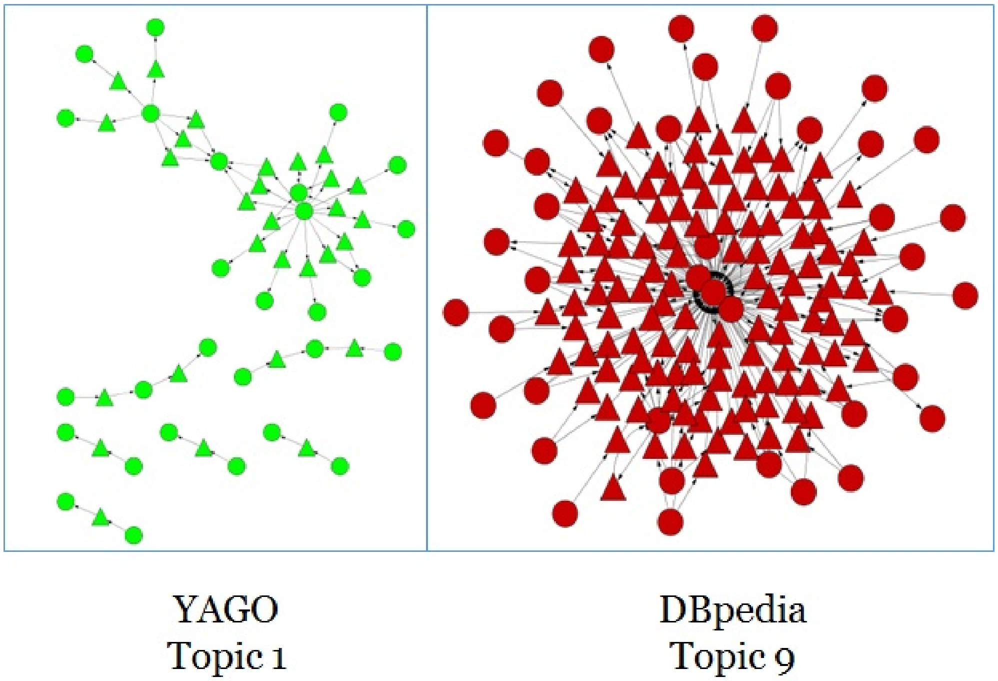
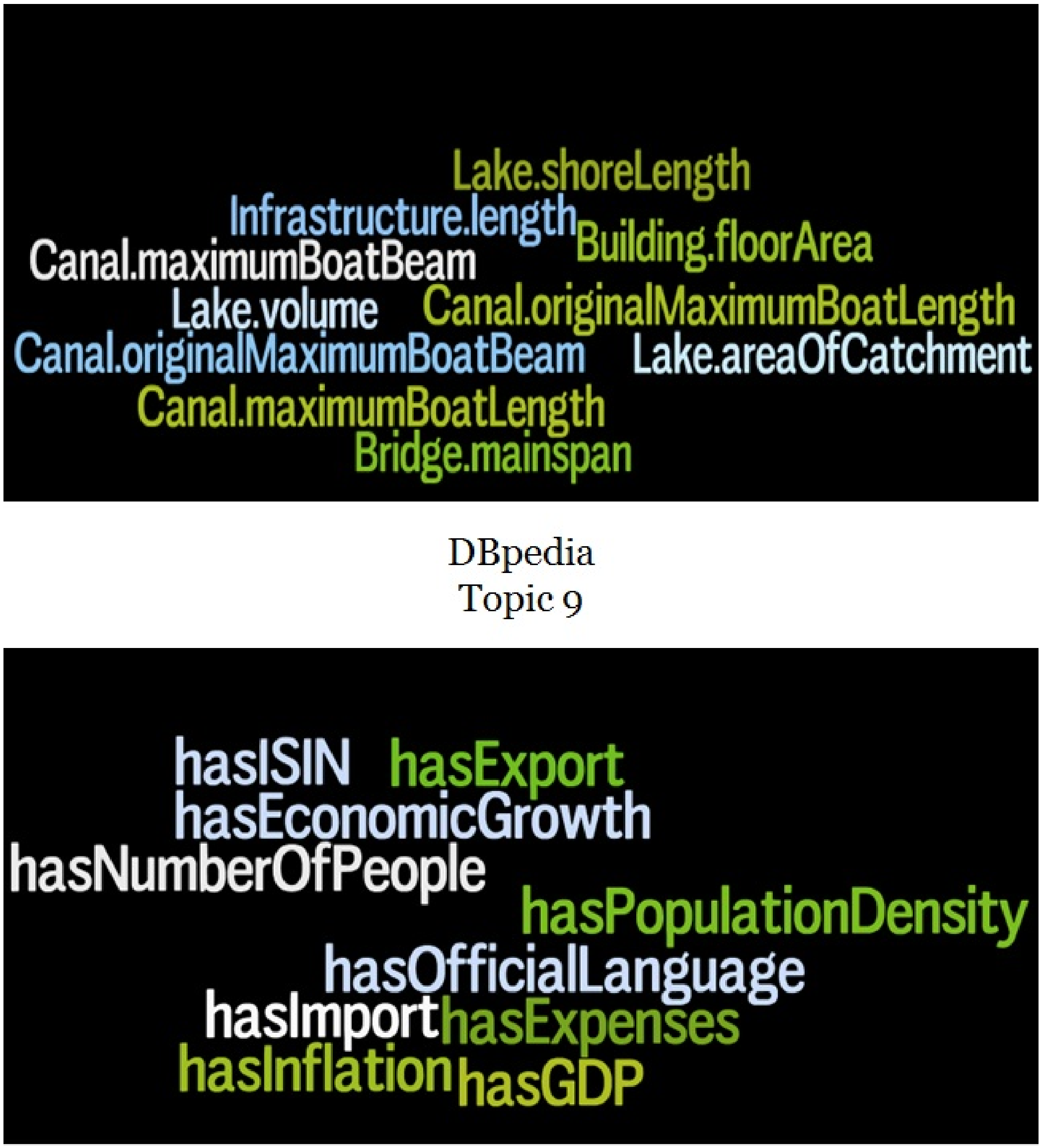


Figure 37: Visualization of DBpedia Topic 9 and YAGO Topic 1

We visualize Topic 9 of DBpedia and Topic 1 of YAGO in Fig. 37. We also list detailed predicates in these 2 topics in Fig. 38. Topic 9 of DBpedia talks about civil engineering and Topic 1 of YAGO introduces country related information.

Evaluation between Predicate Oriented and Concept Oriented Approach

In this section, we conduct several evaluations to compare predicate and concept oriented neighbour pattern analysis. Different from predicate oriented approach, concept oriented one considers each concept as a node and counts in-degree/out-degree predicate to calculate the similarity among concepts and applies HPKM algorithm on concepts based similarity matrix. For topic outputs generated by predicate oriented and concept



DBpedia
Topic 9

YAGO
Topic 1

Figure 38: Contents of DBpedia Topic 9 and YAGO Topic 1

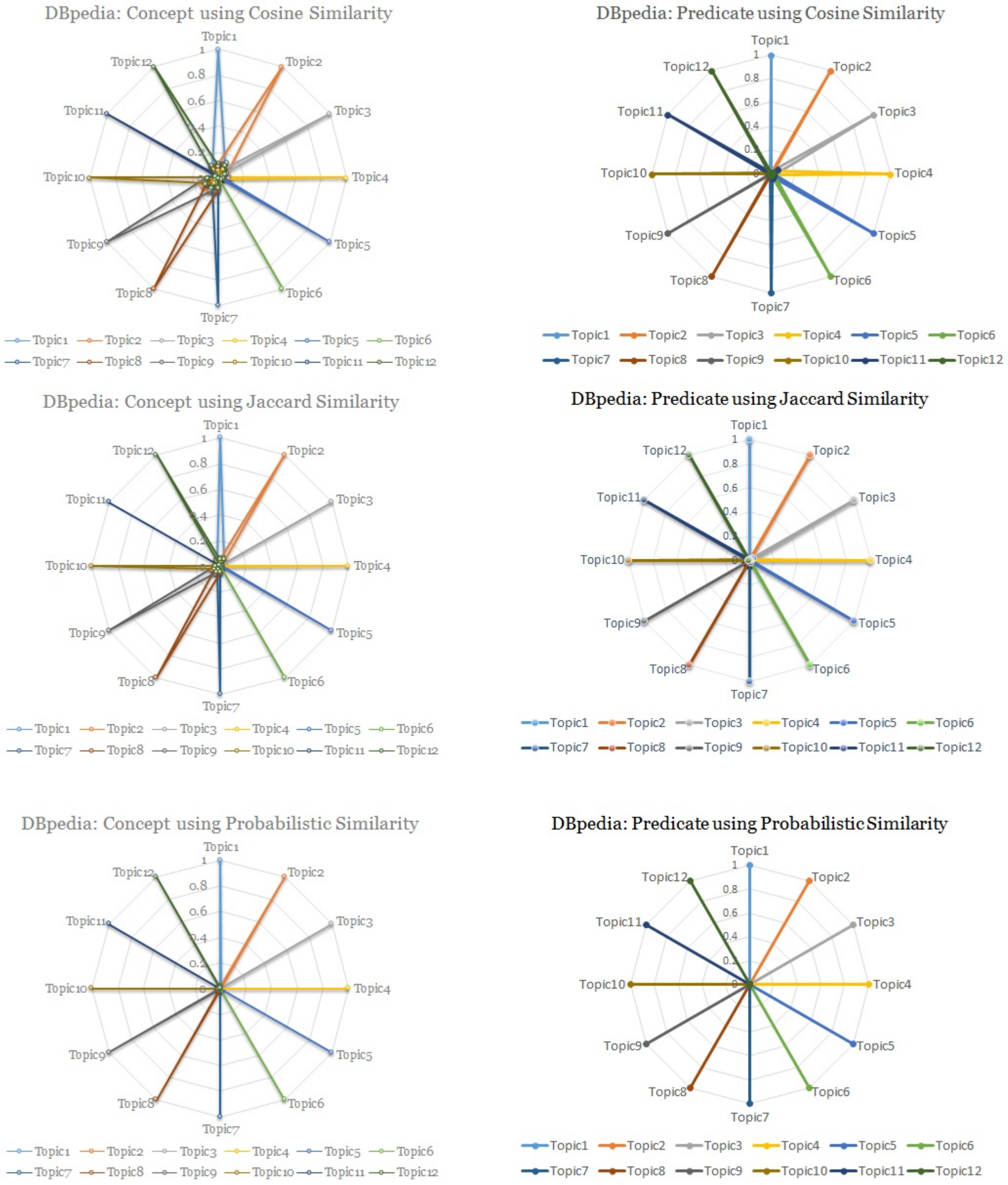
oriented approaches, we evaluate similarity score among topics with three different similarity measurements (Cosine [116], Jaccard [57] and Probabilistic Similarity [71]), respectively.

Specifically, for concept oriented approach, concepts play an important role to partition graph into smaller topics. To calculate the similarity among topics, we not only count concepts for each topic but also include their connected predicates. Therefore, in Eq. (3.2), concept oriented approach considers a_i or b_i as both concepts and predicates; in Eq. (3.3) and (3.4), concept based approach takes $Pred(T_a)$ or $Pred(T_b)$ as both concepts and predicates. However, concept has a more dominant influence on the topic generation results. Similarly, for predicate oriented approach, predicate means more to each topic, but we also include their connected concepts to calculate the similarity score. As a result, in Eq. (3.2), a_i or b_i is considered as predicates and concepts; in Eq. (3.3) and (3.4), $Pred(T_a)$ or $Pred(T_b)$ is considered as predicates and concepts as well. However, in predicate based solution, predicates have a stronger ability to determine clustering results than concepts.

Fig. 39 gives similarity measurements for DBpedia data. The left side shows concept oriented approach with cosine, jaccard and probabilistic similarity while the right side gives predicate oriented approach for each measurement. From Fig. 39, we find that topics generated from concepts oriented approach have more overlapped contents than topics produced from predicates oriented one, which shows the confusion between topic and topic. For predicate oriented similarity measurements, cosine similarity gives some overlaps between topics. For example, Topic 3 and Topic 4 are overlapped a little bit as the angle for two topics shown in Fig 39. Jaccard similarity gives a more clear topic partition. However, as Fig. 39 shows, the style of line for Topic 3, Topic 4, Topic 10, Topic 11 are still thick, which shows the unclear for these topics. Probabilistic similarity gives the best output. From Fig. 39, we find that the style for each line is thinner than other two similarity measurements, which means probabilistic similarity is easier to generate topics with clear contents and boundary. This result shows that predicate leads a better way to categorize and group similar contents into one groups than concept.

Similarly, for YAGO topic, Fig. 40 shows similarity measurement for predicate and concept oriented approaches. Concept oriented similarity shows the confusion among topics. But among all three similarity measurements, probabilistic similarity creates minimum overlap while cosine similarity generates maximum overlap. It shows that probabilistic similarity sets a high criteria to define similarity. That is why we choose probabilistic similarity to generate the PONP association matrix, because it can filter out lots of unnecessary relationships but promote outstanding associations. Same as DBpedia result, predicate oriented approach for YAGO also gives strictly separation for each topic for each similarity measurement. It shows the uniqueness of predicate and it proves predicate performs better to find topics for single domain datasets.

As a summary, for DBpedia and YAGO with the GraphKDD, average similarity scores for different similarity measurements are shown in Table 21. The smaller similarity score among topics, the more clear each topic is, and the less confusion among topics. It gives another evidence that for each dataset, concept based approach gives more confusion than predicate approach. Furthermore, probabilistic similarity gives optimized topic


DBpedia: Concept using Cosine Similarity

DBpedia: Predicate using Cosine Similarity

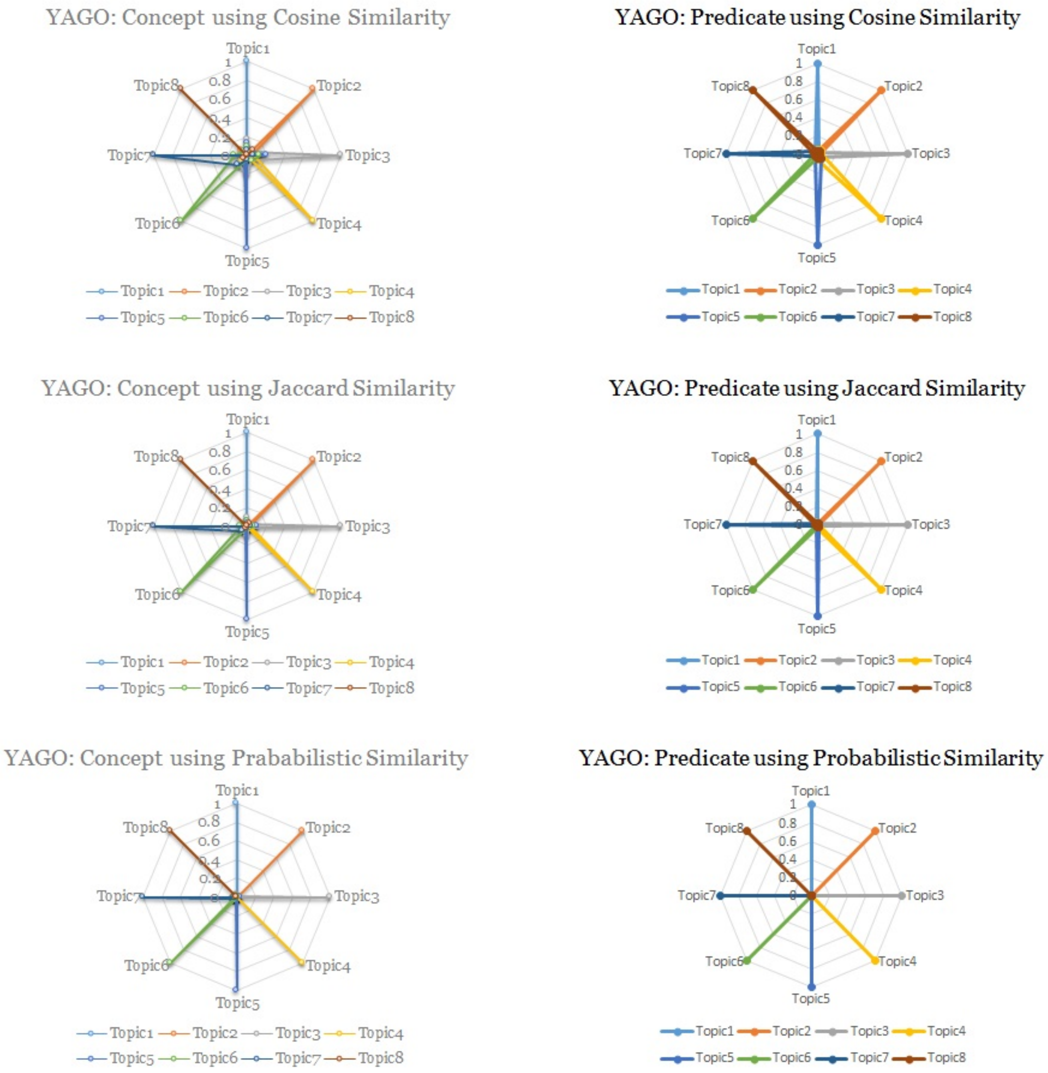
DBpedia: Concept using Jaccard Similarity

DBpedia: Predicate using Jaccard Similarity

DBpedia: Concept using Probabilistic Similarity

DBpedia: Predicate using Probabilistic Similarity

Figure 39: DBpedia Similarity Measurement with GraphKDD



YAGO: Concept using Cosine Similarity

YAGO: Predicate using Cosine Similarity

YAGO: Concept using Jaccard Similarity

YAGO: Predicate using Jaccard Similarity

YAGO: Concept using Prababilistic Similarity

YAGO: Predicate using Probabilistic Similarity

Figure 40: YAGO Similarity Measurement with GraphKDD

Table 21: Average Topic Similarity Comparison for GraphKDD

	Cosine Concept	**Jaccard Concept**	**Prob Concept**	**Cosine Predicate**	**Jaccard Predicate**	**Prob Predicate**
DBpedia	0.23	0.17	0.13	0.1	0.088	0.083
YAGO	0.34	0.27	0.19	0.158	0.141	0.126

relationship for both concept and predicate approach.

In addition, to perform a comparison study with our approach, we also use SimRank measurement to build the predicate and concept oriented similarity matrix for DBpedia and YAGO respectively. We apply the same HPKM algorithm on the generated similarity matrix. For DBpedia, we get 2 topics for both predicate and concept oriented approach. For YAGO, we get 5 topics for predicated oriented approach and 3 topics for concept oriented approach.

Fig. 41 and Fig. 42 show the topic similarity measurement results in terms of predicate/concept based approach respectively. As a summary, for DBpedia and YAGO with SimRank, corresponding average similarity scores for different similarity measurements are shown in Table 22. It gives another evidence that for each dataset with SimRank, concept based approach gives more confusion than predicate approach. Furthermore, probabilistic similarity gives optimized topic relationship for both concept and predicate approach.

Similar to the GraphKDD approach, for DBpedia and YAGO, SimRank concept oriented approach gives fuzzier results than predicates one, which shows the advantage of using predicates to make good categorization of topics. Moreover, even for less optimized SimRank concept oriented approach, probabilistic similarity gives less overlap

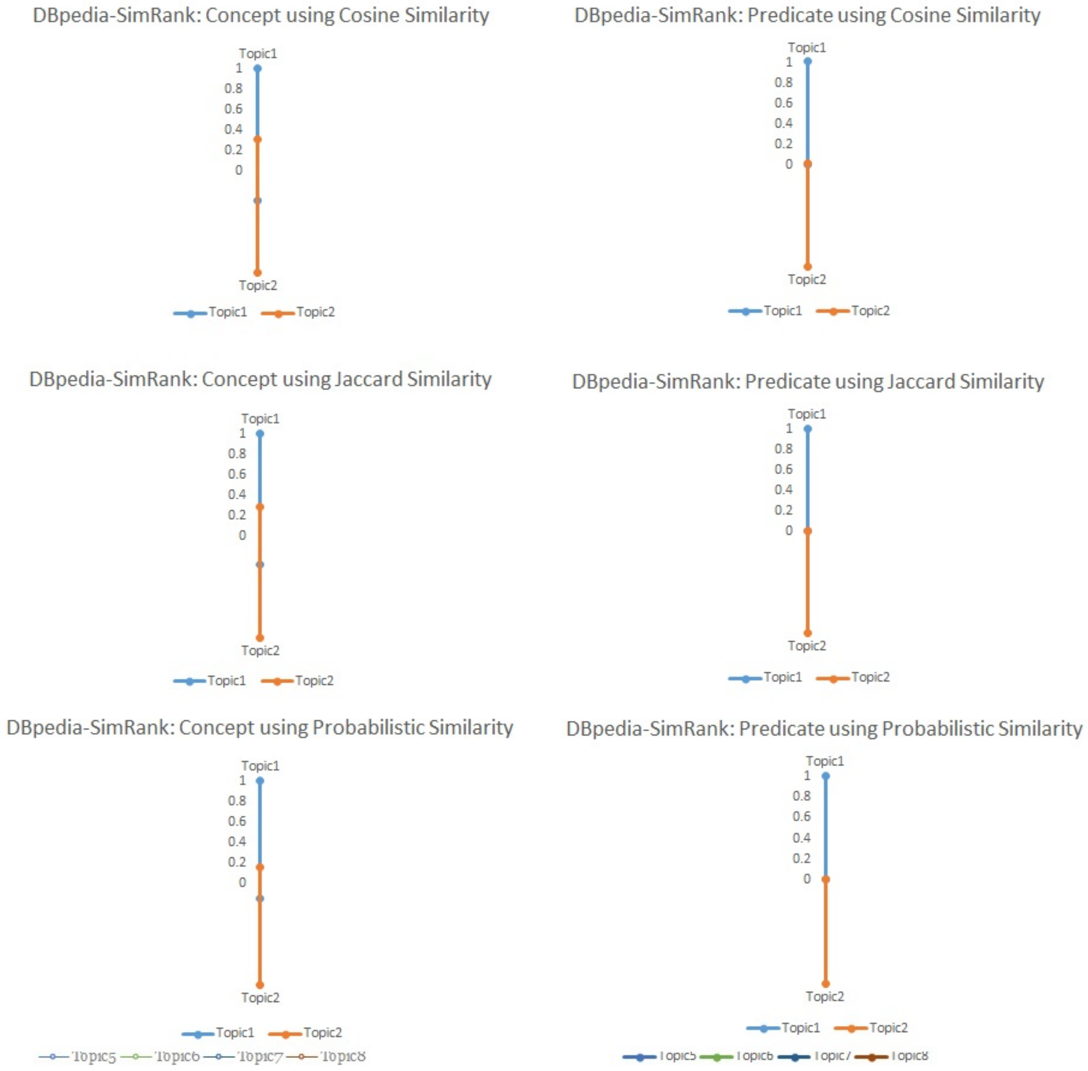
Table 22: Average Topic Similarity Comparison for SimRank

	Cosine Concept	**Jaccard Concept**	**Prob Concept**	**Cosine Predicate**	**Jaccard Predicate**	**Prob Predicate**
DBpedia	0.013	0.001	0.0009	0.006	0.001	0.000035
YAGO	0.38	0.34	0.31	0.26	0.23	0.21

among topics, the results prove that probabilistic similarity is capable of finding "real close" relationships among predicates. However, in terms of the number of topics, the GraphKDD gives more number of topics (12 for DBpedia, 8 for YAGO) than SimRank (2 for DBpedia, 5 for YAGO). It shows that the GraphKDD is able to give more specific group of information.

Evaluation between GraphKDD and Graph Partition Algorithms

We also conduct an experiment to compare the GraphKDD algorithm with four existing random graph partition algorithm: Random Vertex Cut, Canonical Random Vertex Cut, Edge Partition 1D and Edge Partition 2D implemented by Apache Graphx [128]. We use the same number of topic generated by GraphKDD (e.g., 12 for DBpedia and 8 for YAGO) to run four random graph partition algorithm and built topic distribution visualization for each case as shown in Fig. 43 and 44. In both figures, we use different colors to represent different topics, so there are 12 color in each case of Fig. 43 and 8 color in each case of Figure 42. In Fig. 44. We find that topics generated by GraphKDD are distributed in an organized manner except yellow topic (Topic 9) is quite separated. The reason is that Topic 9 has the highest ranking in terms of in-degree/out-degree, page rank, topic size and entropy. Therefore, concepts in Topic 9 has a higher chance to overlap with which in other topics. While for four random graph partition algorithm, topic distribution


DBpedia-SimRank: Concept using Cosine Similarity

DBpedia-SimRank: Predicate using Cosine Similarity

DBpedia-SimRank: Concept using Jaccard Similarity

DBpedia-SimRank: Predicate using Jaccard Similarity

DBpedia-SimRank: Concept using Probabilistic Similarity

DBpedia-SimRank: Predicate using Probabilistic Similarity

Figure 41: DBpedia SimRank Similarity Measurement

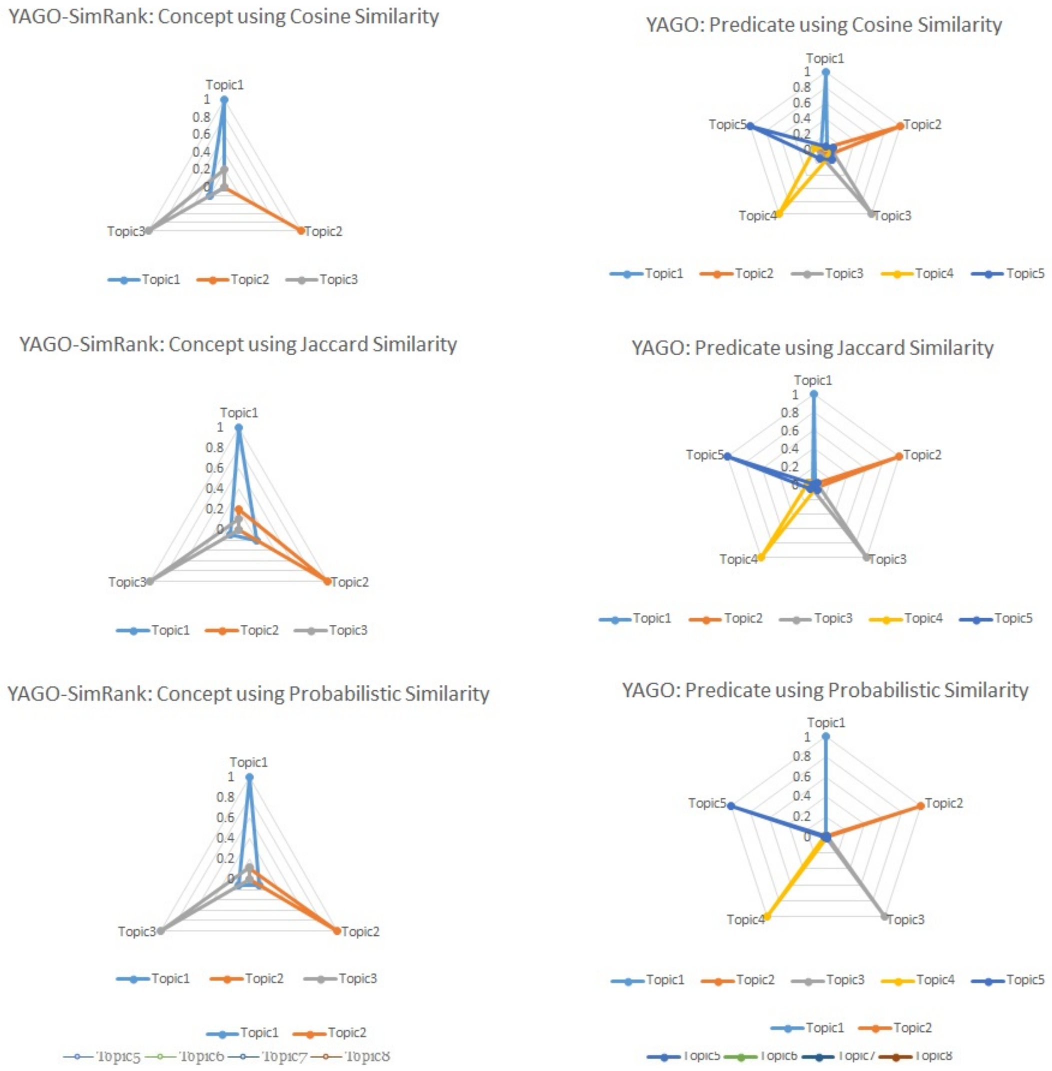


Figure 42: YAGO SimRank Similarity Measurement

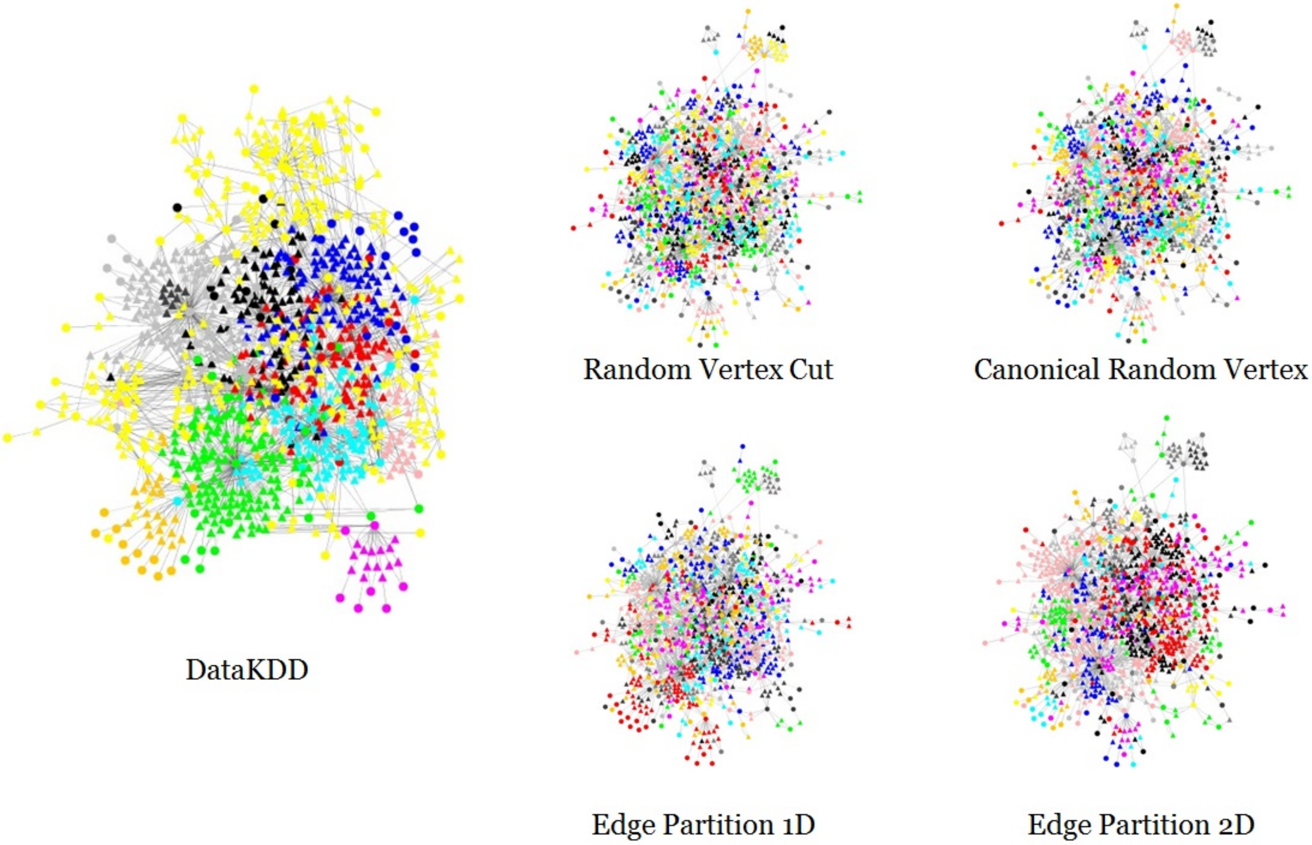


Figure 43: GraphKDD vs. Graph Partition on DBpedia

are quite mixed, which shows the random cut result without consider context awareness. Similarly, in Fig. 44, YAGO topics generated by GraphKDD is more organized compared to other four random approaches. In general, GraphKDD provides a better context awareness topic generation feature and knowledge discovery output than random graph partition approaches.

Evaluation between GraphKDD and LDA

We also conduct a comparison study between GraphKDD and Latent Dirichlet allocation (LDA) [14]. LDA is a statistical topic modeling algorithm that is able to find

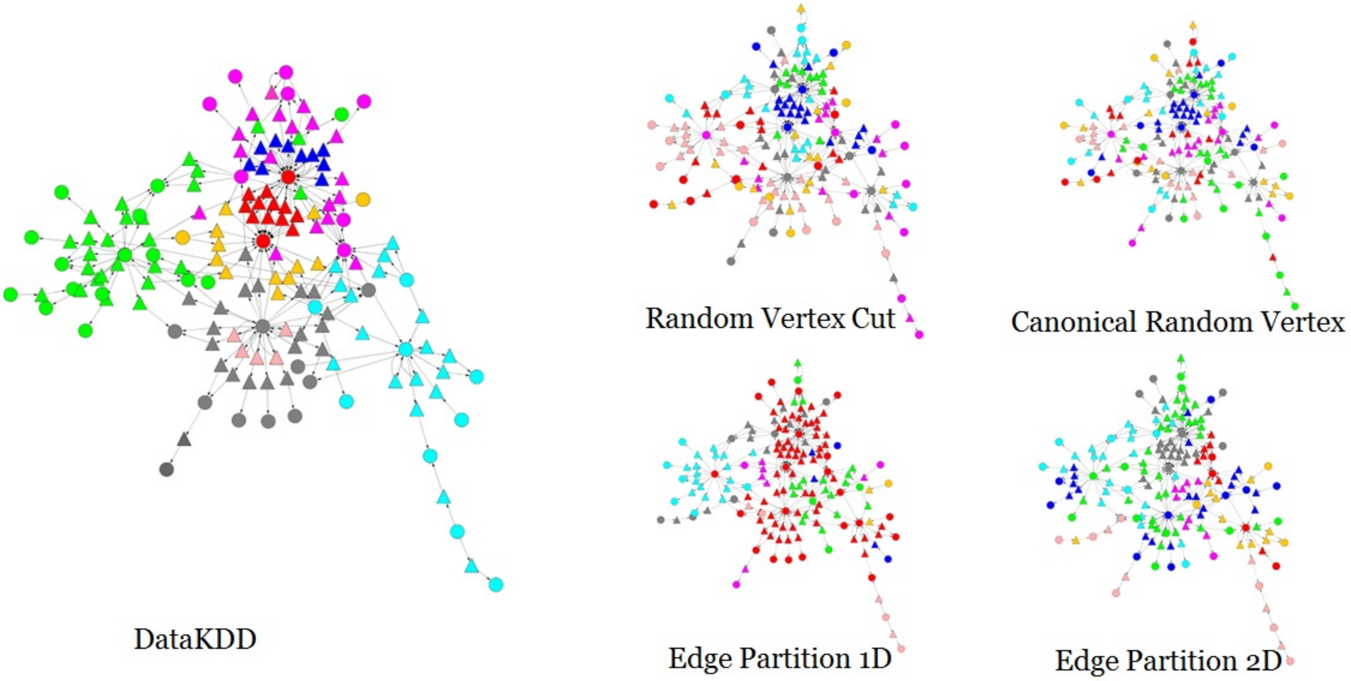


Figure 44: GraphKDD vs. Graph Partition on YAGO

unobserved topics by analyzing observed co-occurrence of words in the document.

We first convert triplets to text input for LDA and give LDA the same number of topic generated by GraphKDD (12 for DBpedia and 8 for YAGO). We then measured the similarity among topics generated by the GraphKDD and LDA for DBpedia and YAGO respectively. Results are shown in Fig. 45.

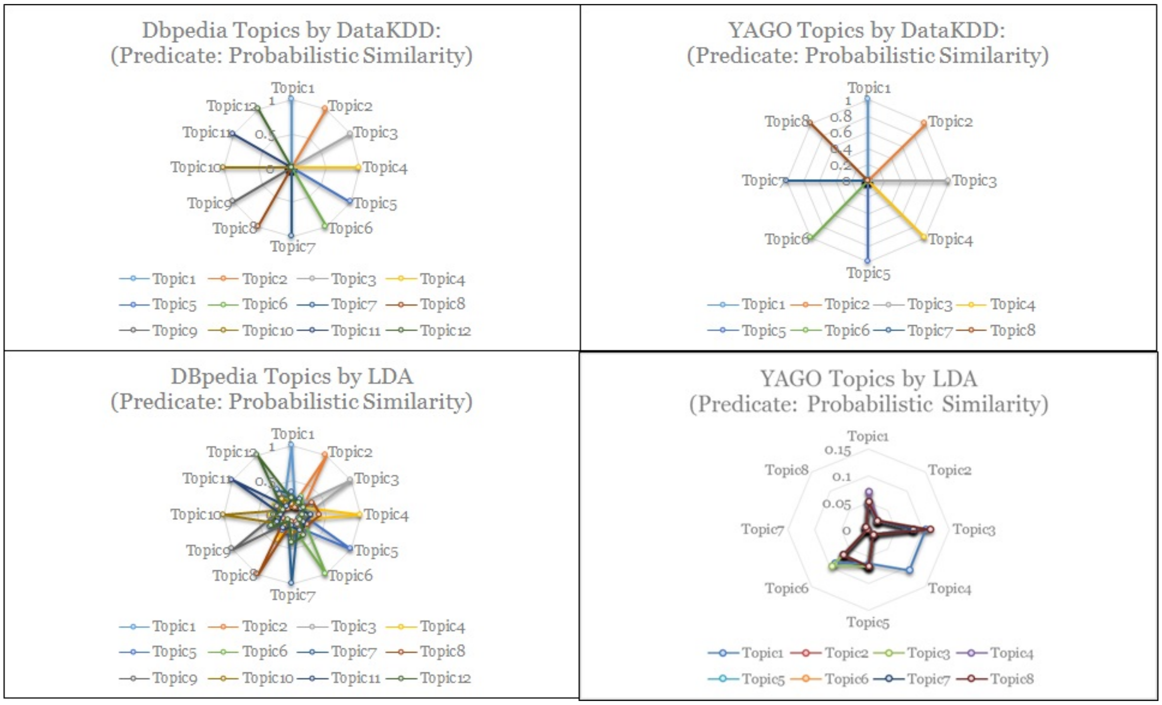
For DBpedia, we find that the GraphKDD is able to give clear partition of topics while LDA gives blurred topic generation results. Similarly, the GraphKDD produced non-overlapped topics but give highly overlapped and confusion topic results for YAGO. The reason for the less optimized output is that LDA usually accepts human language as input and decides topic based on co-occurrence among word characters. However, we give triplets as input to LDA, which makes more duplication than normal text input and leads to the confusion among topics.

3.6.3 Cross Domain Analysis

In this section, we target on the evaluation with cross domains datasets. In addition to HPKM, we also evaluate on Predicate oriented Hierarchical Agglomerative Clustering (PHAL).

3.6.3.1 Bio2RDF 9 domains Case Study

As Table 29 shows, we select 9 bio2rdf datasets (ClinicalTrials, CTD, DrugBank, HGNC, MGI, OMIM, PharmGKB and Sider) to show a cross domains case study. Specifically, this section includes 1) compare PHAL with PHKM on Bio2RDF 9 domains datasets; 2) topic discovery in Bio2RDF 9 domains datasets; 3) ranking of patterns and



Dbpedia Topics by DataKDD: (Predicate: Probabilistic Similarity)

YAGO Topics by DataKDD: (Predicate: Probabilistic Similarity)

DBpedia Topics by LDA (Predicate: Probabilistic Similarity)

YAGO Topics by LDA (Predicate: Probabilistic Similarity)

Figure 45: GraphKDD vs. LDA

topics in cross domains.

PHAL with HPKM

The case studies involve the comparative analysis with the HPKM and PHAL algorithms and experiments with the both algorithms to confirm the effectiveness of the proposed method. For the given nine ontologies shown in Table 29, we conduct the topic discovery by applying the proposed PHAL algorithm and the HPKM algorithm. In Table 29, each ontology is assigned with a color (for example, the color of ClinicalTrials is yellow) that is used in a topic/patten graph. P represents Predicates, C represents Concepts and T represents Triples. Some of the built-in OWL/RDF concepts and predicates are omitted in this work. The information in Table 29 is extracted from the Bio2RDF project $http://download.openbiocloud.org/release/3/release.html$. As mentioned previously, HPKM is an excellent way to summarize an integrated multiple cross-domain datasets, as shown in Fig. 21. However, HPKM could not capture interesting patterns from heterogeneous information networks of cross domains. From the HPKM analysis in Table 23, only seven coarse grained topics are discovered and two of them are cross domains. It is because predicates from a single domain are strongly related compared to ones from cross domains. From the PHAL analysis in Table 23, we find 43 topics from the heterogeneous information networks of the given cross domains and 93% of the discovered patterns (40 are cross domains and 3 are single domain) are cross domains. In addition, we compute the average predicate number per topic, the average in-degree and output-degree per topic, the average density per topic and the association score per topic. The density is computed using $D = \frac{2E}{N(N-1)}$ where N is the number of nodes (concepts

Table 23: Cross Domain Clustering: PHAL vs. HPKM

	PHAL	**HPKM**
Topic #	43	7
Cross Domain Topic #	40	2
Average Diversity (Domain#)	4.14	2.28
Total Predicate Size	539	330
Average Predicate Size per Topic	12.5	47.14
Average In-degree and Out-degree per Topic	45(I) 30(O)	142(I) 89(O)
Average Density per Topic	252	490
Average Predicate Association Score	0.42	0.70

Comparison between Top-down Clustering (HPKM - Hierarchical Predicate-based K-Means Clustering) and Bottom-up Clustering (PHAL - Predicate-based Hierarchical Agglomerative Clustering). In PHAL, the fuzzy clustering is allowed for predicates so that the predicates may appear in more than one topic. The density is computed using $D = \frac{2E}{N(N-1)}$ where N is the number of nodes (concepts and predicates) and E is the number of edges (links between nodes). The association score are computed by the Predicate Association formula Eq. (2.4). Zero is defined as the smallest number. The closer to zero, the smaller it is.

and predicates) and E is the number of edges (links between nodes). The association score are computed by the Predicate Association formula Eq. (2.4). Zero is defined as the smallest number. The closer to zero, the smaller it is. The results demonstrate the PHAL algorithm provides superior outcomes compared with HPKM in topic discovery from heterogeneous information network.

Table 24 shows that there are 330 unique predicates and 275 unique concepts. Interestingly, about 88% of the predicates and 65% of the concepts are cross domains. Fig. 6(a) and Fig. 6(b) show top 10 concepts and top 25 predicates, respectively. Fig. 7(a) and Fig. 7(b) show the top 40 cross domains concepts and predicates, respectively. The nine ontologies used in our case study show high potentials to be used for cross-domain analysis and linking for semantic interoperability.

As seen in Table 24, about 26% of concepts (99 out of 374) appear in more than

Table 24: Cross Domain Concepts and Predicates before/after Clustering

	Before Clustering			**After Clustering**	**PONP Pattern**		**Count per Topic**		
	Unique	Total	Cross Domain	Total	Share	Connectivity	Average	Max	Min
Predicates	330	330	291	539	329	330	12.5	119	2
Concepts	275	374	243	1745	275	374	40.6	181	2

The predicate/concept count before and after clustering. Many of them are cross domains that can be easily associated with concepts/predicates from other domains. After the clustering, both concepts and predicates are duplicated (fuzzy clustering). The concepts/predicate counts for share and connectivity patterns are reported. In addition, average, min and max of concepts and predicates per topic are reported.

one ontology even before the clustering while all 330 predicates are unique (this means each predicate appear in only one ontology among 9 ontologies). Specifically, a generic concept like *Resource* appears 92 times and *pubmed_vocabulary:Resource* appears in all 9 domains. This indicates that concepts like *Resource* are mainly used for high level mapping between different domains. Thus, these concepts are too abstract to interpret such integrated data. For the data integration, data normalization is performed to map 30 Semanticscience Integrated Ontology (SIO) concepts to domain concepts. In addition, about 45% (149 of 330 predicates) are named with a prefix *x*. This indicates that the predicates are also too abstract to provide meaningful relationships between concepts. After clustering, the size of predicates became doubled and the concepts quintupled. All the predicates except *sider_vocabulary:reported.frequency* are fully contributed to the integration of cross domains and discovery of relevant patterns. Through the normalization and clustering, relevant concepts and predicates are integrated and clustered according to their contexts.

Table 25: Cross Domain Neighborhood Patterns

	Share Pattern			**Connection Pattern**		
Patterns	Provider	Consumer	Reacher	DC	NDC	Total
Cross Domain	842	2690	1432	1990	14434	21388
Total	1676	5953	3572	1990	14434	27625

Cross Domain Patterns per type of the PONP patterns (Provider, Consumer, Reacher, Directional Connector and Non-Directional Connector

Topic Discovery in Bio2RDF 9 domain datasets

Table 25 shows the Predicate Oriented Neighborhood Patterns (PONP) discovered from 43 topics: 1676 *Provider* Patterns, 5953 *Consumer* Patterns, 3572 *Reacher* Patterns, 1990 *Directional Connector* patterns and 14434 *Non-Directional Connector* patterns. Interestingly, 77% of the PONP patterns we discovered are cross domains (50% of the *Provider* patterns, 45% of the *Consumer* Patterns, 40% of the *Reacher* Patterns, 100% of the *Directional Connector* patterns, and 100% of the *Non-Directional Connector* patterns). The lower level share patterns are part of the higher level Connectivity patterns. From these results, we can see that the PONP patterns play a significant role in integrating data and finding cross domains topics from heterogeneous information networks.

3.6.3.2 Bio2RDF 13 domains Case Study

We select top 13 most frequently used Bio2RDF datasets to demonstrate another cross domains case study. Specifically, we include 1) Bio2RDF statistics with HPKM and PHAL; 2) Topic Ranking for Bio2RDF with HPKM and PHAL; 3) predicate oriented approach with concept oriented output.

Statistics of Bio2RDF with HPKM and PHAL

Table 26: Bio2RDF HPKM vs. PHAL

	HPKM	**PHAL**
Num of Topic	5	47
Topic Size	73	91
Average Silhouette Width	0.43	0.52
Similarity Score	0.02	0.54

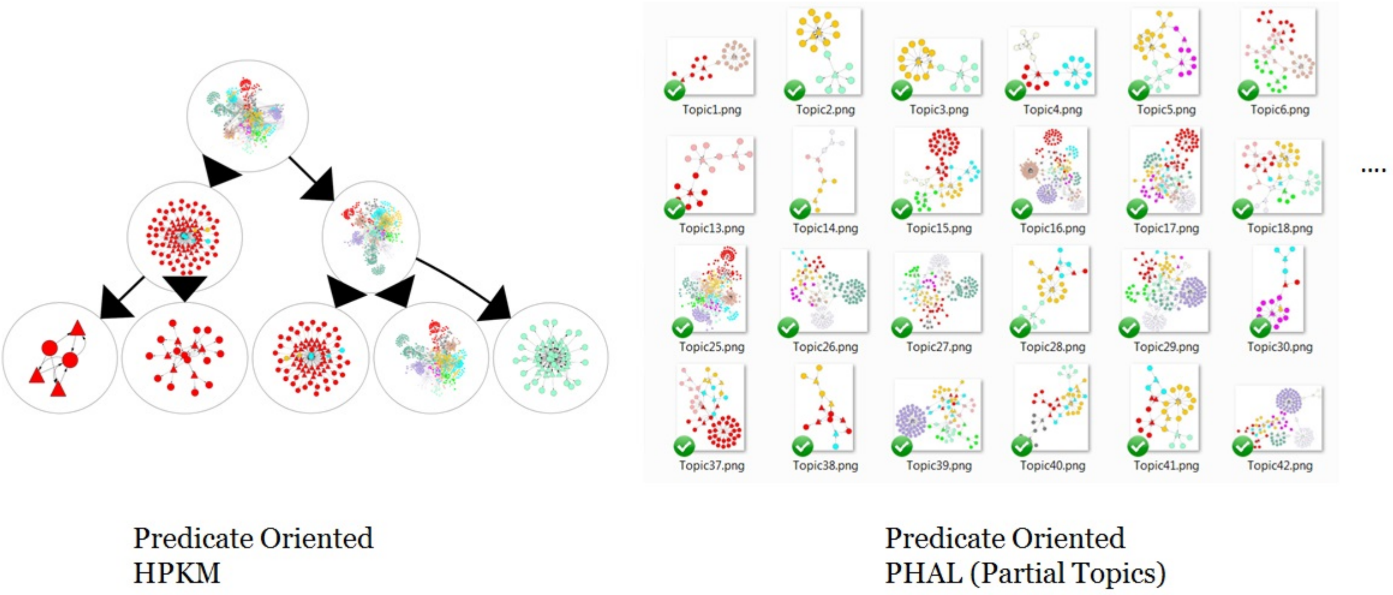
We apply a top-down algorithm HPKM and a bottom-up algorithm PHAL on 13 integrated Bio2RDF datasets. As Table 26 shows, HPKM only generated 5 topics while PHAL produced 47 topics, which is much more than HPKM. In addition, average topic size for HPKM is 73 and which for PHAL is 91. Moreover, average silhouette width for each topic generated by HPKM is 0.43 while which for PHAL is 0.52. What is more, PHAL also achieves a higher similarity score 0.54 than HPKM (0.02)

In terms of cross domains topic generation, PHAL also performs better. Fig. 46 shows all Bio2RDF topic visualization with HPKM and partial topic visualization with PHAL (different color represent different domain). We find that HPKM only produces 60% (3 out of 5) cross domains topics while PHAL generates 100% (47 out of 47) cross domains topics. In general, we conclude that for cross domains datasets, PHAL gives a better performance than HPKM.

Topic Ranking for Bio2RDF with HPKM and PHAL

We first count the number of predicates and concepts for each topic generated by Bio2RDF with HPKM and PHAL respectively as Fig. 47 shows. From Fig. 47, we also find that for HPKM, Topic 4 is the dominant topic while for PHAL, Topic 25 is the dominant one.

We also conduct a topic ranking evaluation based on I/O, page rank, topic size



Predicate Oriented
HPKM

Predicate Oriented
PHAL (Partial Topics)

Figure 46: Bio2RDF 13 domain Topic Visualization with HPKM and PHAL

rank and entropy rank for Bio2RDF HPKM and PHAL as Fig. 48 shows. We find that this ranking is consistent with predicate count and concept count statistics for both datasets. In HPKM, Topic 4 has the most I/O, page rank, topic size and entropy. In PHAL, Topic 25 holds the most I/O, page rank, topic size and entropy. From here we conclude that there is no necessary relationship between predicate/concept number with topic ranking. However, if one topic has larger size of predicate/concept, it is more like to have higher I/O, page rank, topic size and entropy. Detailed topic visualization and contents for HPKM Topic 4 and PHAL Topic 25 are shown in Fig. 49 and 50.

Compare Predicate oriented Approach with Concept oriented Approach

For predicate and concept oriented topics generated by HPKM and PHAL, we apply three similarity measurements (Cosine, Jaccard and Probability Similarity) to test

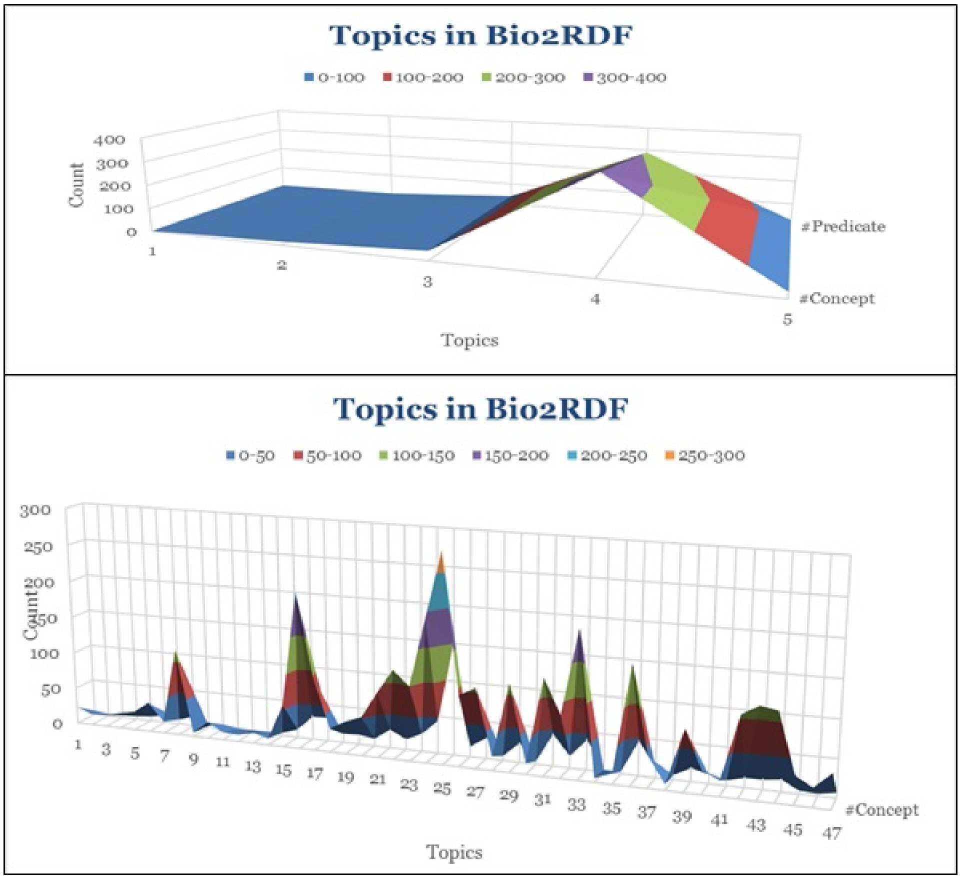


Figure 47: Bio2RDF Topic Statistics with HPKM and PHAL

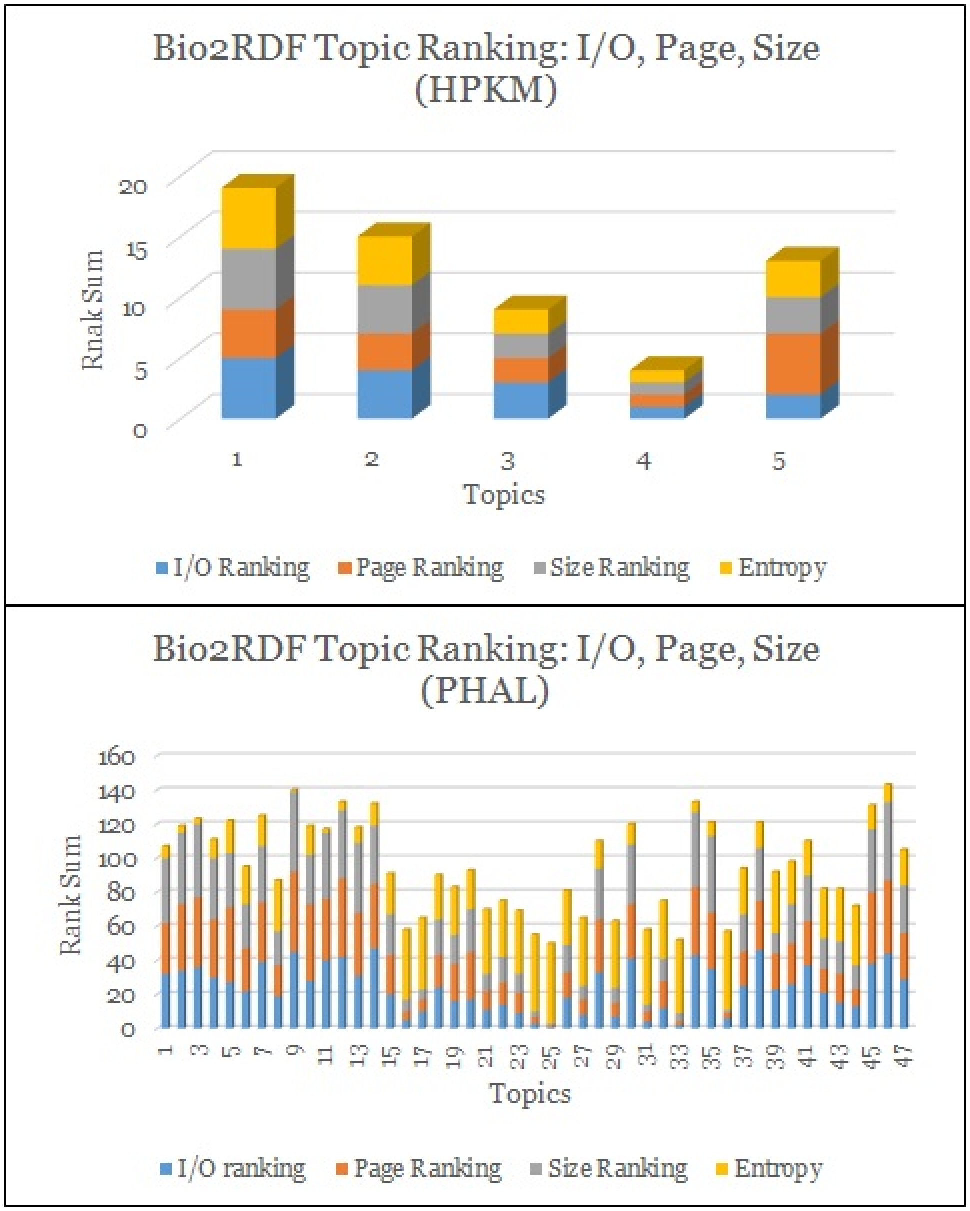


Figure 48: Bio2RDF Topic Ranking with HPKM and PHAL

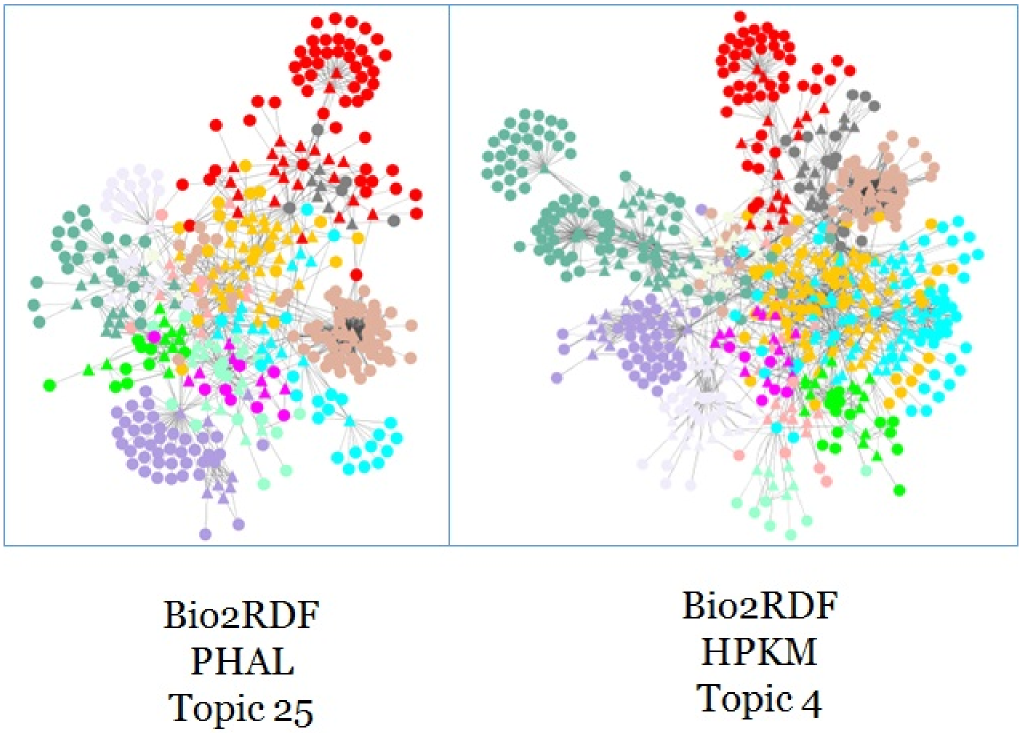


Figure 49: Bio2RDF Dominant Topic Visualization with HPKM and PHAL

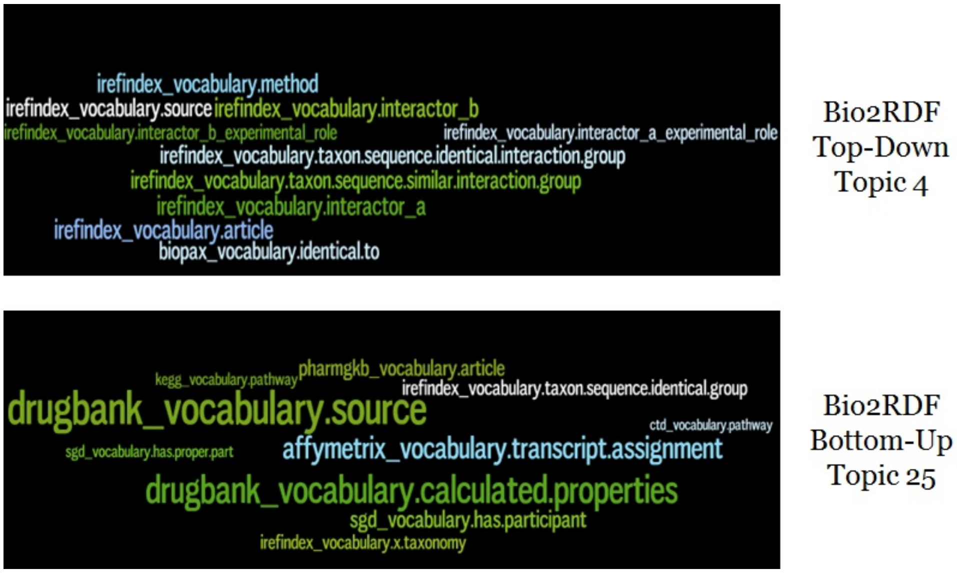

irefindex_vocabulary.method
irefindex_vocabulary.source
irefindex_vocabulary.interactor_b
irefindex_vocabulary.interactor_b_experimental_role
irefindex_vocabulary.interactor_a_experimental_role
irefindex_vocabulary.taxon.sequence.identical.interaction.group
irefindex_vocabulary.taxon.sequence.similar.interaction.group
irefindex_vocabulary.interactor_a
irefindex_vocabulary.article
biopax_vocabulary.identical.to
Bio2RDF
Top-Down
Topic 4
kegg_vocabulary.pathway
pharmgkb_vocabulary.article
irefindex_vocabulary.taxon.sequence.identical.group
drugbank_vocabulary.source
ctd_vocabulary.pathway
sgd_vocabulary.has.proper.part
affymetrix_vocabulary.transcript.assignment
drugbank_vocabulary.calculated.properties
sgd_vocabulary.has.participant
irefindex_vocabulary.x.taxonomy
Bio2RDF
Bottom-Up
Topic 25

Figure 50: Bio2RDF Dominant Topic Content with HPKM and PHAL

the overlap among topics. As Fig. 51 shows, for 5 topics generated by HPKM, concepts oriented approach gets highly overlapped topics than predicate oriented for all three measurement. Similarly in Fig. 52, for 47 topics generated by PHAL, predicate oriented approach gives better hard partition of topics than concept oriented one. In general, predicate oriented approach gives more clear topic generation without much overlap and confusion.

As a summary shown in Table 27, Bio2RDF with bottom-up approach (PHAL) generate a higher topic similarity score in terms of concepts than Bio2RDF with top-down algorithm (HPKM). It is because PHAL is a fuzzy algorithm but HPKM provides only hard clustering solution. From this point, we conclude that PHAL is more suitable to find topic with more cross domains information. In addition, we plot a graph based on Table 27 as shown in Fig. 72. We find that for predicate based approach, Bio2RDF

Table 27: Average Topic Similarity Comparison for Cross Domain GraphKDD

	Cosine Concept	**Jaccard Concept**	**Prob Concept**	**Cosine Predicate**	**Jaccard Predicate**	**Prob Predicate**
Bio2RDF bottom-up	0.473	0.367	0.265	0.196	0.109	0.07
Bio2RDF top-down	0.43	0.34	0.25	0.22	0.21	0.2

bottom-up approach gives less similarity scores for all three measurement when compared with top-down approach. It proves that compared with concept, predicate is more easier to differentiate topic and create less confusion among topics. As a result, for cross domains datasets, PHAL combined with predicate oriented clustering algorithm give the best output.

3.6.4 Classification Performance

In this section, we build classifier based on training topics generated by DBpedia HPKM, YAGO HPKM, Bio2RDF HPKM and Bio2RDF PHAL. In addition, we apply 10 fold cross-validation on the training model and give precision, recall and F-measure for each case. We try to predict which predicate belong to which topic by using supervised learning approach. Table 28 gives the detailed classification evaluation results with Naive Bayes algorithm. We find that for single domain hard partition topic generation approach (DBpedia, YAGO and Bio2RDF HPKM), we get relative higher precision, recall and F-Measure outputs. However, for cross domains fuzziness topics (Bio2RDF PHAL), performance is relative low. The reason is that for HPKM results, all predicates are hard partitioned, there is no duplication of predicates across topics, which makes the prediction

Bio2RDF: Concept using Cosine Similarity

Bio2RDF: Predicate using Cosine Similarity

Bio2RDF: Concept using Jaccard Similarity

Bio2RDF: Predicate using Jaccard Similarity

Bio2RDF: Concept using Probabilistic Similarity

Bio2RDF: Predicate using Probabilistic Similarity

Figure 51: Bio2RDF Topic Similarity with HPKM

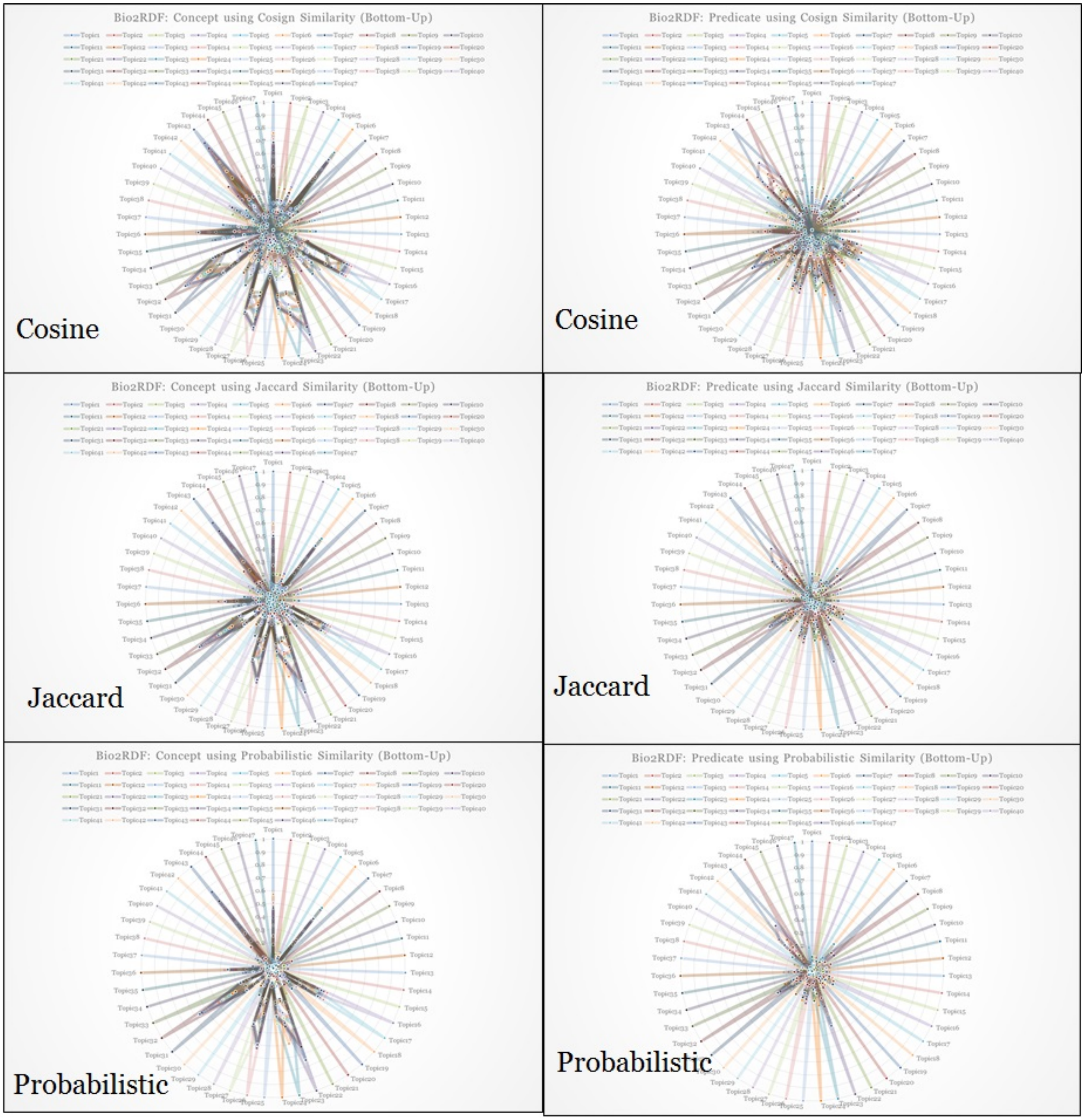


Figure 52: Bio2RDF Topic Similarity with PHAL

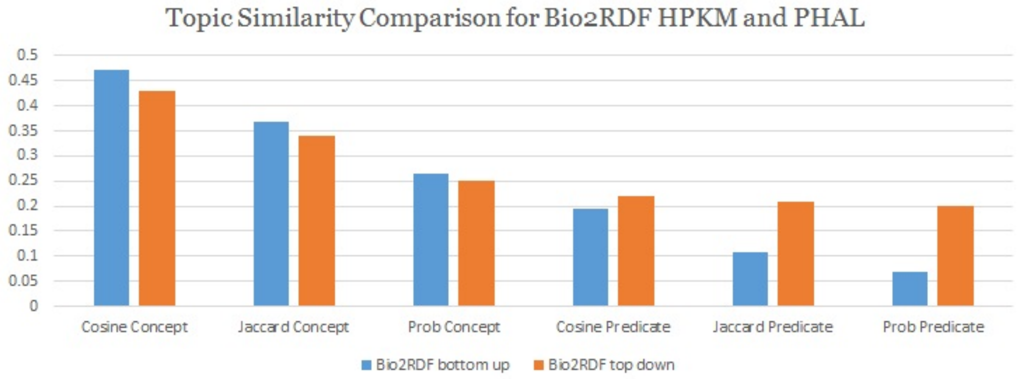


Figure 53: Bio2RDF Topic Similarity with HPKM and PHAL

Table 28: Classification Evaluation

	Num of Topics	Precision	Recall	F-Measure
DBpedia HPKM	12	0.75	0.83	0.78
YAGO HPKM	8	0.63	0.75	0.67
Bio2RDF HPKM	5	0.7	0.8	0.73
Bio2RDF PHAL	47	0.06	0.15	0.08

easier. However, for PHAL approach, one predicate belongs to more than one topic,which make the decision making difficult. Naive Bayes cannot handle such multiple label case very well, that is why precision, recall and F-Measure do not give an optimal output.

3.7 Summary

We have formally introduced a Hierarchical Predicate oriented K-Means (HPKM) and a Predicate oriented Hierarchical Agglomerative (PHAL) unsupervised approaches to cluster data and generate topics. For HPKM, we have described algorithms of HPKM with global optimization, HPKM with local optimization. For PHAL, we have introduced four phases algorithms involved: hierarchical heterogeneous clustering, middle level initial topic groups generation, disjoint topic construction and hierarchical topic refinement. Specifically, we use HPKM to deal with single domain dataset while use PHAL to handle cross domains knowledge discovery.

In evaluation, we first conducted an experiment on a single domain dataset (DrugBank, DBpedia and YAGO). For DrugBank data, we gave statistics of topic predicates and concepts in terms of predicate in-degree and out-degree and duplicated concepts and their topic ID in DrugBank. Moreover, we validated HPKM optimal branching factor and listed DrugBank topic generation results . For DBpedia and YAGO data, we first conducted an experiment to determine the optimal predicate neighbourhood radius boundary by a heuristic study. All the results have showed the optimal solution provided by the GraphKDD HPKM approach.

In addition, we also tested the optimal HPKM branching factor, topic ranking in terms of in-degree/outdgree, page rank, size of topic and entropy, predicate oriented approach with concept oriented approach with similarity measurement, compare HPKM with graph partition algorithms, compare HPKM with LDA algorithm. Secondly, we adopted a analysis for cross domains datasets. As a use case study, we selected Bio2RDF 13 domains datasets. We have compared the topic ranking for HPKM and PHAL and predicate oriented with concept oriented approaches. All the outputs have showed that PHAL is more suitable to handle cross domains datasets than HPKM, and predicate oriented approach gives better outputs.

Furthermore, we have trained four different topic results (DBpedia with HPKM, YAGO with HPKM, Bio2RDF with HPKM, Bio2RDF with PHAL) and applied naive bayes algorithm on them to perform 10-fold cross validation. Results have showed that single domain topic classifier has a better performance on predicting topic than cross domains one.

Table 29: Case Study Datasets: Ontologies

Ontology	**P#**	**C#**	**T#**	**Description**
ClinicalTrials (Yellow)	56	62	486	a registry and results database of publicly and privately supported clinical studies of human participants conducted around the world.
CTD (Magenta)	14	19	74	cross-species chemical-gene/protein interactions and chemical- and gene-disease relationships to illuminate molecular mechanisms underlying variable susceptibility and environmentally influenced diseases.
DrugBank (Red)	63	92	401	a bioinformatics and chemoinformatics resource that combines detailed drug (i.e. chemical, pharmacological and pharmaceutical) data with comprehensive drug target (i.e. sequence, structure, and pathway) information.
HGNC (Pink)	14	16	34	The HGNC gives unique and meaningful names to every human gene.
KEGG (Orange)	72	61	299	an integrated database resource consisting of 16 main databases, broadly categorized into biological systems information, genomic information, and chemical information.
MGI (Green)	14	20	68	This includes data on gene characterization, nomenclature, mapping, gene homologies, among mammals sequence links, phenotypes, allelic variants and mutants, and strain data.
OMIM (Light Green)	35	30	175	a comprehensive, authoritative, and timely compendium of human genes and genetic phenotypes. The full-text, referenced overviews in OMIM contain information on mendelian disorders and over 12,000 genes. OMIM focuses on the relationship between phenotype and genotype.
PharmGKB (Cyan)	47	60	218	PharmGKB curates primary genotype and phenotype data, annotates gene variants and gene-drug-disease relationships via literature review, and summarizes important PGx genes and drug pathways.
SIDER (Gray)	15	14	82	SIDER contains information on marketed medicines and their recorded adverse drug reactions. The information include side effect frequency, drug and side effect classifications and links to further information
Total	330	374	1837	Cross domain data model based on these 9 datasets

CHAPTER 4

QUERY GENERATION AND TOPIC AWARE LINK DISCOVERY

4.1 Introduction

Some applications are derived from the GraphKDD framework. Query generator and path finding tools are two of knowledge discovery applications. In this chapter, we first introduce some related work in developing knowledge discovery applications. Then we formally introduce our query generation and path finding model. In addition, we conduct evaluations on both applications to give performance analysis and result discussion.

4.2 Related Work

Knowledge discovery from RDF data mainly focuses on information extraction. Query generation is one of the approach. SP2Bench [102] proposed a query design system focusing on generating queries with combination of different operations. But this query generation was not designed from a semantic perspective for cross domains. LUBM [45] and BSBM [13] generated benchmarks on university and ecommerce respectively, but neither benchmark was based on more than one domain. FedBench [101] provided a benchmark suite for federated queries on semantic data which can cover semantic multiple domains data use cases. However, query benchmarks were manually generated by authors. Our approach provides a way to automatically help people find the semantic relationship without acquiring knowledge explicitly.

For query processing, H2RDF+ [87] provided a scalable distributed RDF store to facilitate query processing performance. Trinity [132] presented a RDF management framework over a distributed in-memory key-value store. There are still some other indexing approaches for speeding query processing like RDF-3X [84] and gStore [28], which can reduce graph searching space and avoid looking up unnecessary blocks. However, neither of them works on a cross domains and semantic perspective. Indexing technology in GraphKDD provides semantic meaning for graphs. Some other ontology query related researches, such as [76] [17] [43]. Queries can often be difficult to formulate across these datasets [75]. In particular, the work from [76] has a similar approach to our work in terms of detecting recurring query patterns based on the distance among RDF graph patterns and identifying query templates from the analysis of the RDF graph structure. However, this work focuses more on concepts of the instance level of RDF graphs for the pattern identification and template extraction. Unlike this work, we focus on a new paradigm, such as predicate based similarity patterns, at the schema level for topic discovery and query suggestion.

Biomedical data contributors have provided public SPARQL endpoints to query the datasets. However, Quilitz and Leser [90] and Alexander et al., [2] merely provided the statistical information on the datasets instead of conceptual analysis for knowledge discovery from biomedical datasets. There is little effort for the schema level analysis of the concepts and their relationships in these datasets with respect to systematic and semantic querying. Seaborne and Prudhommeaux [89] pointed out the difficulty with the SPARQL syntax and expression, because the precise details of the structure of the graph should be specified for queries in the triple pattern through the various heterogeneous schemas. In reality, users may not be familiar with the details of datasets, and it is hard to express the precise relationships between concepts in the SPARQL syntax and expressions. Thus, this can be a bottleneck for users to query through the endpoints of medical ontologies.

Callahan et al., [17] provided a SPARQLed web application for SPARQL query generation by suggesting context sensitive IRI. However, they could not provide strong associated queries as we do. Unlike this work, we can provide not only valid but also meaningful query suggestions in a dynamic manner according to users' interesting topics. Godoy et al., [43] presented a collaborative environment to allow user to register queries manually through wiki pages and share and execute the queries for linked data. A series of desired queries might be generated using large ontologies like the NCI thesaurus by extracting relevant information [83]. The GLEEN project aims to develop a useful service for simplified, materialized views of complex ontologies [30].

There also exist some knowledge discovery applications in bioinformatics domain. Most of the work has mainly focused on building or using ontologies for data normalization, bridging and reasoning. Widely used medical ontologies are Bio2RDF [10], TMO (Translational Medicine Ontology) [77], Chem2Bio2RDF [20], SIO (Semanticscience Integrated Ontology) [34], ATC (Anatomical Therapeutic Chemical) and DrugBank integration [21], Linked Life Data [82].

However, these works lack the ability to design specific queries from topics and are not able to perform the comprehensive context awareness and semantic analysis of large sources and the usage of the knowledge for query processing. Unlike these works, the GraphKDD based query generator is to automate query generation through predicate neighborhood pattern-based topic discovery without any human intervention.

Drug discovery research heavily relies on multiple information sources to validate potential drug candidates as shown in the Open PHACTS project [127]. In complicated domains, it takes time to develop and maintain ontologies [7] [135]. There have been various studies on using semantic techniques to improve data integration and share information. DrugBank is one of the key resources which provide bioinformatics and cheminformatics studies with complete information on drug and drug targets.

However, these efforts merely support physical integration of multiple biomedical ontologies without considering semantic integration of data. In particular, human intervention is strongly required so that these are not suitable for comprehensive and accurate knowledge discovery especially from a large amount of data. Furthermore, semantic interoperability is difficult to achieve in these systems as the conceptual models underlying datasets are not fully exploited.

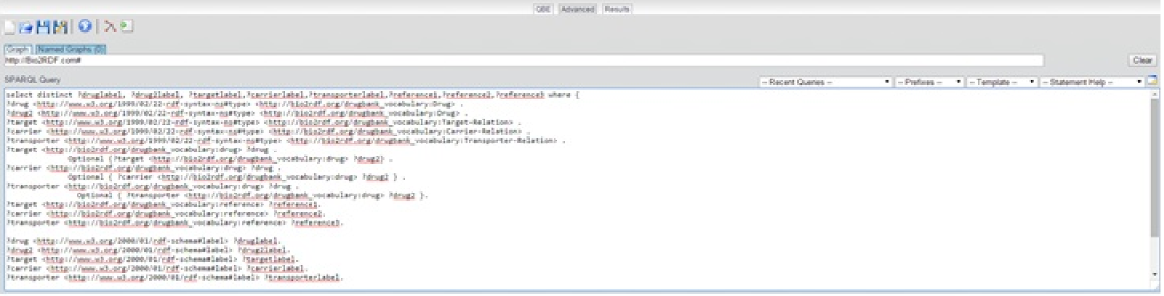
There also exist some RDF path finding tools. LIMES [85] proposes a efficient large scale link discovery tool by using the similarity optimization to filter out unnecessary path in advance. Freek Dijkstra et al., [31] investigated a RDF-based shortest path finding tool in multi-layer network, which is able to find more valid paths than single-layer path vector algorithms like BGP [93], OSPF-TE [68] and SS7 [41]. Silk [123] proposed a way to find entities between different web data sources by utilizing owl:sameAs links and other types of RDF links. It also supports using SPARQL language to specify conditions. Viswanathan and Krishnamurthi [122] presented a modified bidirectional BFS algorithm to find paths between entities and is able to give path ranking based on users' specific needs. BRAHMS [58] is an efficient RDF storage system that designed for fast association discovery.

However, compared with the GraphKDD path finding tool, these research work lack the ability to aware context for discovered paths. In addition, the GraphKDD proposes a predicated based path finding approach that also supports efficient path finding solution through predicate oriented topic association.

4.3 The GraphKDD System

The GraphKDD system is implemented using Java in Eclipse Juno Integrated Development Environment [4]. Apache Jena API [61] is used to parse OWL/RDF datasets and retrieve triple information. We use R computing environment [54] for our experimental validation. We implemented a software plugin for query and schema graph visualization using CytoScape 3.0.2 [103]. In addition, we have built a SPARQL query endpoint on a single machine that is hosted at the UMKC Distributed Intelligent Computing (UDIC) lab. The OPEN LINK Virtuoso server version 6.1.3 is installed and different ontologies are imported into the graph domain $http : //Bio2RDF.com\#$. The endpoint for SPARQL query services is $http : //134.193.129.248 : 8890/isparql/$.

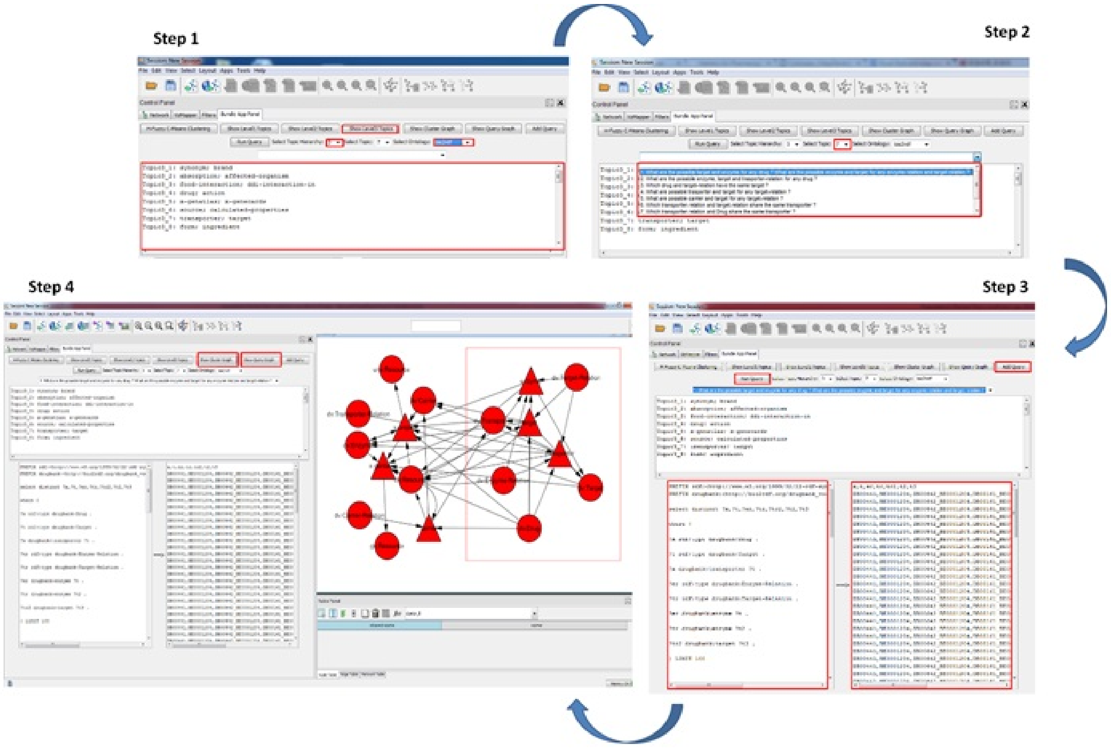
Fig. 54 shows how to perform the SPARQL query only for Bio2RDF Drugbank



http://bio2RDF.com#

```
select distinct ?druglabel, ?drug2label, ?targetlabel,?carrierlabel,?transporterlabel,?reference1,?reference2,?reference3 where {
?drug <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Drug> .
?drug2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Drug> .
?target <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Target-Relation> .
?carrier <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Carrier-Relation> .
?transporter <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Transporter-Relation> .
?target <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug .
            Optional {?target <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug2} .
?carrier <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug .
            Optional { ?carrier <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug2 } .
?transporter <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug .
             Optional { ?transporter <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug2 }.
?target <http://bio2rdf.org/drugbank_vocabulary:reference> ?reference1.
?carrier <http://bio2rdf.org/drugbank_vocabulary:reference> ?reference2.
?transporter <http://bio2rdf.org/drugbank_vocabulary:reference> ?reference3.

?drug <http://www.w3.org/2000/01/rdf-schema#label> ?druglabel.
?drug2 <http://www.w3.org/2000/01/rdf-schema#label> ?drug2label.
?target <http://www.w3.org/2000/01/rdf-schema#label> ?targetlabel.
?carrier <http://www.w3.org/2000/01/rdf-schema#label> ?carrierlabel.
?transporter <http://www.w3.org/2000/01/rdf-schema#label> ?transporterlabel.
```

Figure 54: SPARQL Endpoint: Query Example.



```
Topic0_1: synonym; brand
Topic0_2: absorption; affected-organism
Topic0_3: food-interaction; ddi-interactor-in
Topic0_4: drug; action
Topic0_5: x-genelist; x-genecards
Topic0_6: source; calculated-properties
Topic0_7: transporter; target
Topic0_8: form; ingredient
```

1. What are the possible target and enzyme for any drug ? What are the possible enzyme and target for any enzyme relation and target relation ?
2. What are the possible enzyme, target and transporter relation for any drug ?
3. Which drug and target-relation have the same target ?
4. What are possible transporter and target for any target-relation ?
5. What are possible carrier and target for any target-relation ?
6. Which transporter relation and target relation share the same transporter ?
7. Which transporter relation and Drug share the same transporter ?

Figure 55: GraphKDD Interactive Query Tool

generated from one of the topic. Fig. 55 shows how to use the GraphKDD tool for browsing the generated topics and performing interactive design and processing of queries. Here we still use DrugBank dataset as the use case. Step 1 shows the list of topics for a given ontology (DrugBank). Step 2 shows the list of NLP questions for a selected topic (Topic 7). Step 3 shows the automatically generated SPARQL query and the query results. Step 4 shows the topic and query graphs for the selected query. The steps for the query generation and processing using GraphKDD tool are explained as follows:

Step1: A user first selects a dataset (e.g., DrugBank) to be analyzed, then choose an algorithm to generate a topic hierarchy (e.g., three level hierarchy). In Fig. 55, a clustering algorithm (e.g., Hierarchical K-Means Clustering) button is selected for the construction of a topic hierarchy (DrugBank). Topics generated from the topic hierarchy construction are listed in the top left box. In this example, the eight topics are shown with the detailed description including a list of the highest ranked predicates and their concepts (with high in-degree/out-degree).

Step2: The user selects a topic (e.g., 7th topic) to view, then this allows users to explore top ten natural language queries automatically generated by the proposed query generation algorithm.

Step3: A query can be selected and modified through the interactive query editor based on the topics or predicates shown in Step 2. Once the design of a query is complete, the corresponding natural language query expressions and the corresponding SPARQL query will be generated.

Step4: After choosing the natural language query expressions (e.g., what are the enzyme, target and transporter-relation of a drug?), the add query button can be clicked to select its corresponding SPARQL query into the bottom left box.

Step5: When the query button is clicked, the SPARQL query will be executed and the query output will be shown in the bottom right box.

Step6: When the show query cluster button is clicked, the corresponding cluster graph will be displayed on the canvas in the right panel. Moreover, by clicking the show query graph button, the relevant concepts and predicates in the SPARQL query will also be highlighted as seen in Fig. 55.

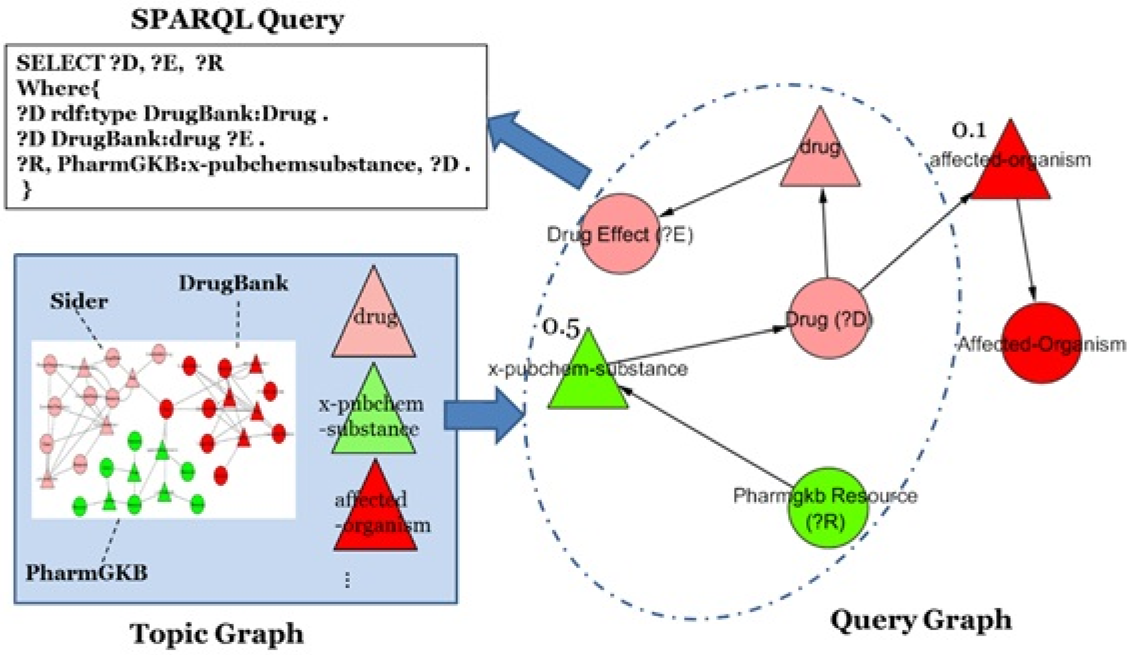
4.4 Query Generation

From the HPKM and PHAL clustering algorithm, topic hierarchies are generated. The Query Generation algorithm will start crawling the leaf nodes (the topics at on the bottom level) in a given topic hierarchy and generate a query that is a part of a particular topic TG (a RDF graph) in the topic hierarchy. The algorithm will crawl the topic graph TG to generate a query graph QG; QG is a subset of the TG. Many variation of queries can be generated from this process. In this work, we first give the relationship among variable, query, topic and graph in Definition 14.

Definition 14: $\forall\ G$, topic T, query Q and variable V, $V \subset Q \subset T \subset G$.

Fig. 56 shows how the query generation algorithm generates queries. The topic graph shown in this figure has three predicates, namely drug from the Sider domain (in pink), *affected-organism* from the DrugBank domain (in red) and *x-pubchem-substance* from the PharmGKB domain (in green).

We start to generate a query by traversing the predicate that has the highest rank δ (the highest sum of the in-degree and out-degree of the predicate) and traverse its neighbors level-by-level (Breadth-first Search) in the descending order of the similarity in the SM computed by the PONP algorithm. For this traversal, we consider the neighbors whose similarity scores are higher than threshold β. In this example, we start with the best predicate drug and then visit its neighbors whose similarity scores are higher than the threshold (β = 0.2) in a descending order. In Fig. 56, for drug, its nearest neighbor, *x-pubchem-substance* with the similarity score 0.5, thus we expand drug with an additional predicate, *x-pubchem-substance*. And then drug's next nearest neighbor is affected-organism with the similarity score 0.1. Since the similarity score is less than the threshold β, (i.e., 0.1 $\leq$ 0.2), we terminate the navigation. The algorithm runs until there is no more neighboring predicates to be considered. The generated query includes triples with two predicates, drug and *x-pubchem-substance*, and their subject variables ($?E$ and $?D$) and object variables ($?D$ and $?R$) as seen in Fig. 56. The type of variable $?E$ is known as Drug Effect, $?D$ as Drug, and $?R$ as PharmGKB Resource. This can be converted to a triplet form such as $\langle ?D$ typeof Drug $\rangle$. Fig. 56 shows an example of the automatically generated SPARQL query for a topic graph. Corresponding query generation algorithm is also shown in Algorithm 7



```
SELECT ?D, ?E,  ?R
Where{
?D rdf:type DrugBank:Drug .
?D DrugBank:drug ?E .
?R, PharmGKB:x-pubchemsubstance, ?D .
 }
```

Figure 56: Automatic Query Generation

Algorithm 7 Query Generation

```
Input: Topic $T = \{t_1, \ldots t_n\}$
Output: Query $Q = \{q_1, \ldots, q_k\}$
Define Queue U
for each topic $t_k$ in $T$ do
    Sort predicates by descending order of their in-degree+out-degree, save in $S$
    for each predicate $p$ in $S$ do
        U.push ($p$)
    end
    while $U$ is not empty do
        $p$ = U.pull
        Get all $p$'s neighbors $p_n$, sort them by descending order of similarity score, save in $N$
        Find shared concepts between $p$ and $p_n$, generate triples, save in $T$
        Replace $T$ with variables, save results in $Q$
        U.push ($N$)
    end
end
return $Q$
```

4.5 Topic Aware Link Discovery

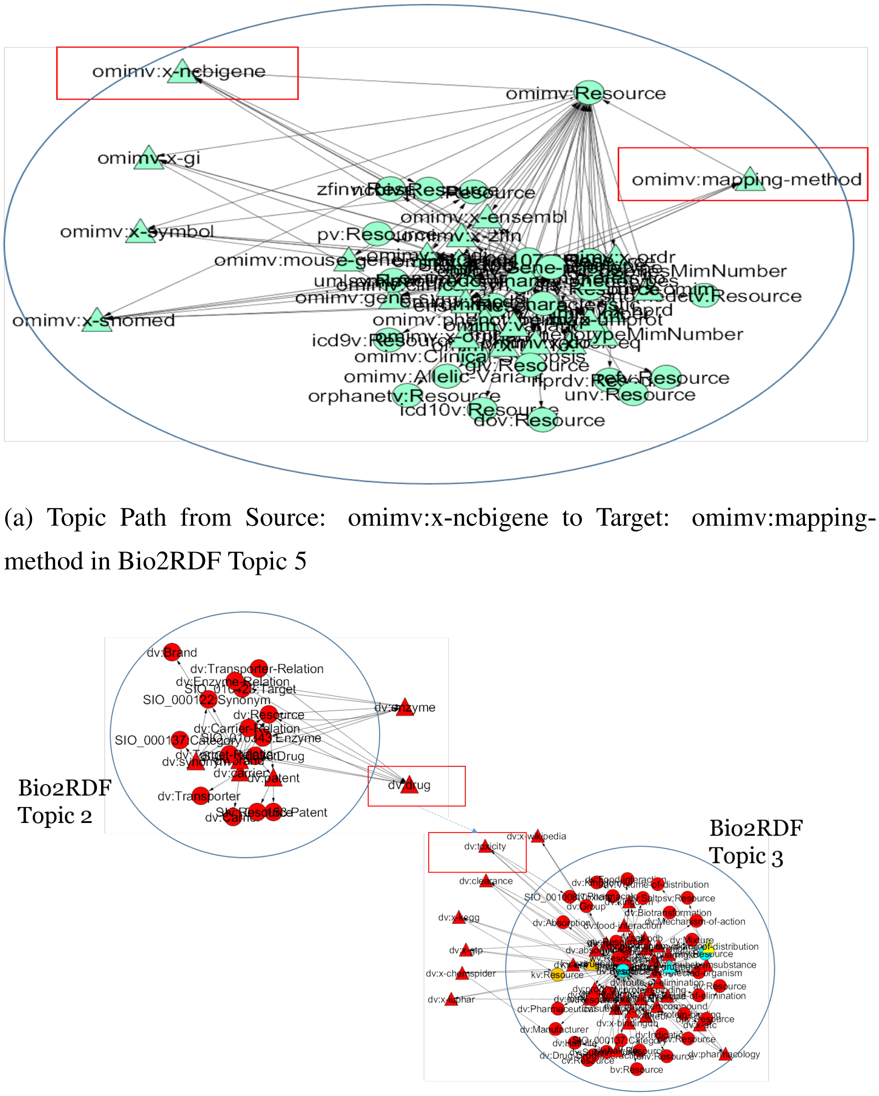
We now describe how to process topic aware link discovery. In the topic aware link discovery in TopicGraph, *topic connects* of topics are used to find a path from source to target through topic links. There are three cases of topic aware link discovery as follows:

- Case 1: For given two topics $Topic_x$ and $Topic_y$, $Topic_x$ has a direct relationship (i.e., the distance D with $Topic_y$ through the connectors P_x and P_y where the distance between P_x and P_y, D(P_x,P_y)= 1 and P_x=$\{P_{x1}, P_{x2}, \ldots\}$ and P_y=$\{P_{y1}, P_{y2}, \ldots\}$.

- Case 2: For given two topics $Topic_y$ and $Topic_z$, $Topic_y$ has a direct relationship (i.e., the distance D between P_y and P_z, D(P_y,P_z)= 1) with $Topic_y$ through the connectors P_y and P_z where P_y=$\{P_{y1}, P_{y2}, \ldots\}$ and P_z=$\{P_{z1}, P_{z2}, \ldots\}$.

- Case 3: For given three topics $Topic_x$, $Topic_y$ and $Topic_z$, $Topic_x$ has an extended relationship with $Topic_Z$ through $Topic_y$ (i.e., the distance D between P_x and P_z, D(P_x,P_z)= 2).

The topics are discovered with the bounded contexts which are a central concept in the knowledge discovery. The clustering technique is applied to partition a large and complex network into multiple smaller topics in the same context in an optimal manner. The bounded contexts are specifically tailored for a set of cross domains patterns. Fig. 57 shows two cases of the topic aware link discovery: (i) link from omimv:x-ncbigene to omimv:mapping-methodin in Topic 5 (ii) link from dv:drug in Bio2RDF Topic 2 to dv:toxicity in Bio2RDF Topic 3.



(a) Topic Path from Source: omimv:x-ncbigene to Target: omimv:mapping-method in Bio2RDF Topic 5

(b) Topic Path from Source: dv:drug in Bio2RDF Topic 2 Target: dv:toxicity in Bio2RDF Topic 3

Figure 57: Topic Aware Link Discovery

The Topic Aware Link Discovery algorithms have three cases: source and target are within the same topic; source and target belong to different topics but they are a connection pair; source and target are in different topics and they do not form a connector pair. Detailed algorithm for each case are shown in Algorithm 8, 9, and 10.

Algorithm 8 Topic Aware Link Discovery - Case1

Input: TopicMap $[T_1, \ldots, T_i]$, Source s, Target t

Output: Paths between s and t $P = [P_1, \ldots, P_n]$, Topics per path $T_p = [< T_i, \ldots T_j >]$

```
 Let $CM < s, t >$ be ConntectorPairMap
   Let $PM[< PL_s, PL_t >, PathList]$ be PathMap
   Let $IM < s, t >$ be InnerPairMap
   /* Case1: $s$ and $t$ belong to the same topic group */
for each topic $T_i$ not in $T$ do
    if $T_i$ contains $s$ and $t$ then
        if $s$ reaches $t$ then
            pathList = PM.get($< PL_s, PL_t >$) //get the path from PathMap
             P.add(pathList) // add paths and eliminate duplicate paths
             $T_p$.add($< T_i >$)
        end
    end
end
```

Algorithm 9 Topic Aware Link Discovery - Case2

Input: TopicMap $[T_1, \ldots, T_i]$, Source s, Target t

Output: Paths between s and t $P = [P_1, \ldots, P_n]$, Topics per path $T_p = [< T_i, \ldots T_j >]$

```
/* Case2: $s$, $t$ belong to different topics but they are a connector pair */
for any two topics $T_i$ and $T_j$ do not exist in $T$ do
    if $T_i$ contains $P_m$ and $T_j$ contains $P_n$ then
        if $CM$ contains $<$ s,t $>$ then
            pathList = $s \rightarrow t$
            P.add(pathList) // add paths and eliminate duplicate paths
            $T_p$.add($< T_i, T_j >$)
        end
    end
end
```

Algorithm 10 Topic Aware Link Discovery - Case3

Input: TopicMap $[T_1, \ldots, T_i]$, Source s, Target t

Output: Paths between s and t $P = [P_1, \ldots, P_n]$, Topics per path $T_p = [< T_i, \ldots T_j >]$

```
for any topic $T_i$ and $T_j$ exist in $T$ do
  if $T_i$ contains $s$ and $T_j$ contains $t$ then
    if $CM$ doesn't contain $<$ s,t $>$ then
      if $IM_i$ contains pair $<$ s,m $>$ and $CM$ contains $<$ m,t $>$) or ($CM$ contains
       pair $<$ s,m $>$ and $IM_j$ contains $<$ m,t $>$ then
        pathList = $s \rightarrow m \rightarrow t$
        P.add(pathList) // add paths and eliminate duplicate paths
        $T_p$.add($< T_i, T_j >$)
      end
      if $<$ s,m $>$ exist in $IM_i$ and NOT $< m_1$,t $>$ exist in $CM$ and $< m_2$,t $>$ exist
       in $IM_j$ then
        if $< m_1$,$m_2 >$ exist in $CM$ then
          pathList = $s \rightarrow m_1 \rightarrow m_2 \rightarrow t$
          P.add(pathList) // add paths and eliminate duplicate paths
          $T_p$.add($< T_i, T_j >$)
          if $< m_1$,$m_2 >$ does not exist in $CM$ then
            Go to the next Topic $T_x$, try to find a path $P_x$ between $P_{m1}$ and
             $P_{m2}$
            // $T_x$ represents multiple topics, $P_x$ represent multiple paths
            if $P_x$ exists then
              pathList = $P_s \rightarrow P_{m1} \rightarrow P_x \rightarrow P_{m2} \rightarrow P_t$
              P.add(pathList) // add paths and eliminate duplicates
              $T_p$.add($< T_i, T_j, T_x >$)
            end
          end
        end
      end
    end
  end
end
```

Based on Algorithm 8, 9, and 10, Theorem 5 also indicates that if p_i and p_j are reachable, our path finding algorithm will never miss any such cases.

Theorem 5. $\forall$ predicates p_i and p_j, if p_i and p_j are reachable with any direction, the GraphKDD can always find this reachable pair (p_i, p_j).

Proof. If p_i and p_j are in the same topic, based on Algorithm 8, the GraphKDD is able to find all paths between p_i and p_j. If p_i and p_j are in different topics but they form a connector pair, based on Algorithm 9, the GraphKDD is capable of retrieving all paths between p_i and p_j. If p_i and p_j are in different topics and they do not form a connector pair, based on Algorithm 10, the GraphKDD can extract all paths between p_i and p_j. Therefore, the GraphKDD is able to find $\forall$ pairs (p_i, p_j) if p_i and p_j are reachable with any direction.

4.6 Complexity Analysis

In this section, we give a brief review of complexity for each phase of the GraphKDD framework. Matrix generation is the first step. The complexity of matrix generation is based on the total number of predicates. Assume m is the total size of predicates, the GraphKDD will traverse all m's neighbor to complete the matrix, therefore, the total complexity is O($m(m-1)$) = O($m^2 - m$). For both HPKM and PHAL, matrix generation has the same complexity. Topic generation step is conducted based on matrix generation. This step has different complexity depending on different approaches we use. K-Means has complexity O($n^{dk+1} \log_2 n$) [49], where n is the number of objects to be clustered, d is the dimension and k is the number to cluster. HPKM applies K-Means algorithm

Table 30: Complexity for HPKM and PHAL

	Matrix Generation	**Topic Generation**	**Query Generation**	**Link Discovery**
HPKM	O($m^2 - m$)	O($n^{dk+1} \log_2 n$)	O(mk)	O(c)
PHAL	O($m^2 - m$)	O(n^3)	O(mk)	O(c)

on different level, so the complexity of HPKM is also O($n^{dk+1} \log_2 n$). Similarly, PHAL applies the same complexity O(n^3) as hierarchical clustering [62]. Complexity for query generation is also same for both HPKM and PHAL. Assume m is the number predicates, k is the number of linked predicates for each predicate, the time to build query is O(mk). In addition, link discovery find paths between source and target based on pre-processed neighborhood map. Therefore, time complexity to extract paths is a constant c, which is O(c).

4.7 Evaluation and Results

In this section, we list results for query generation and conduct performance experiment for topic aware link discovery.

4.7.1 Query Generation Outputs

In this section, we use DrugBank as a use case to demonstrate query generation and processing results for each topic. Specifically, we show four best topic graphs at level 3 of DrugBank topic hierarchy as follows: Topic 3_4, Topic 3_7, Topic 3_6, and Topic 3_2. In addition, the automatically generated query and query results of each topic are also shown.

Rank 1: Topic 3_4 (T3_4): This topic consists of 6 predicates and 12 concepts with 72 in-degree and out-degree. Among 12 concepts, five concepts (dv:Resource; dv:Drug, dv:Carrier, dv:Enzyme, dv:Target) are ranked among Top 20 Concepts and all 6 predicates of this topic are ranked among Top 20 Predicates. In particular, there are two groups of predicates; one is with four predicates such as transporter, target, enzyme, carrier with concepts dv:Target-Relation, dv:Target-Relation, dv:Enzyme-Relation, dv:Carrier-Relation, respectively. Another group of predicates such as x-genbank and x-uniprot is a connector predicate group that is mainly used to connect between internal concepts (e.g., dv:Drug, dv:Enzyme) and external concepts (e.g., gv:Resource, unv:Resource). Specifically, T3_4 shows very high rankings for Similarity, Silhouette Width, and Density while showing a relatively low ranking for Top 20 Concepts. The T3_4's overall rank is 1st (together with T3_7) among 8 topics. Fig. 58 shows the T3_4 topic graph. In this graph, concepts are represented as a circle, predicates as a triangle, and links as an arrow. In addition, the dark red items are the predicates and concepts mentioned in Query-1.

Query-1: The following query is automatically generated from Topic 3_4 (one of the top ranked topics) by our query generation algorithm. This query allows users to find the most relevant drugs in terms of their target, enzyme, enzyme relation, and target relation. The SPARQL format of Query-1 is automatically generated by the Query Generation algorithm considering the top predicates and their concepts in Fig. 59. This query can be translated as following: *For any two drugs which share the same target and transporter enzyme, what are all the possible drugs, enzyme, target, enzyme relations, target relations?*

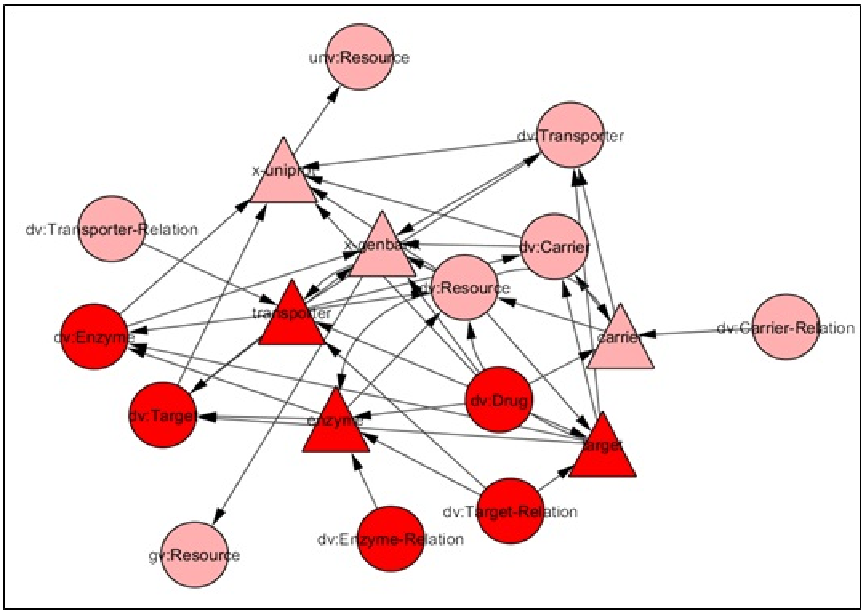
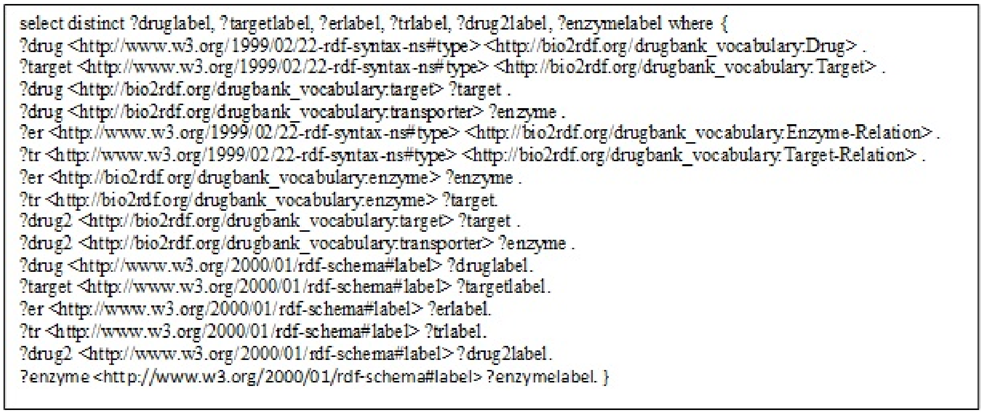
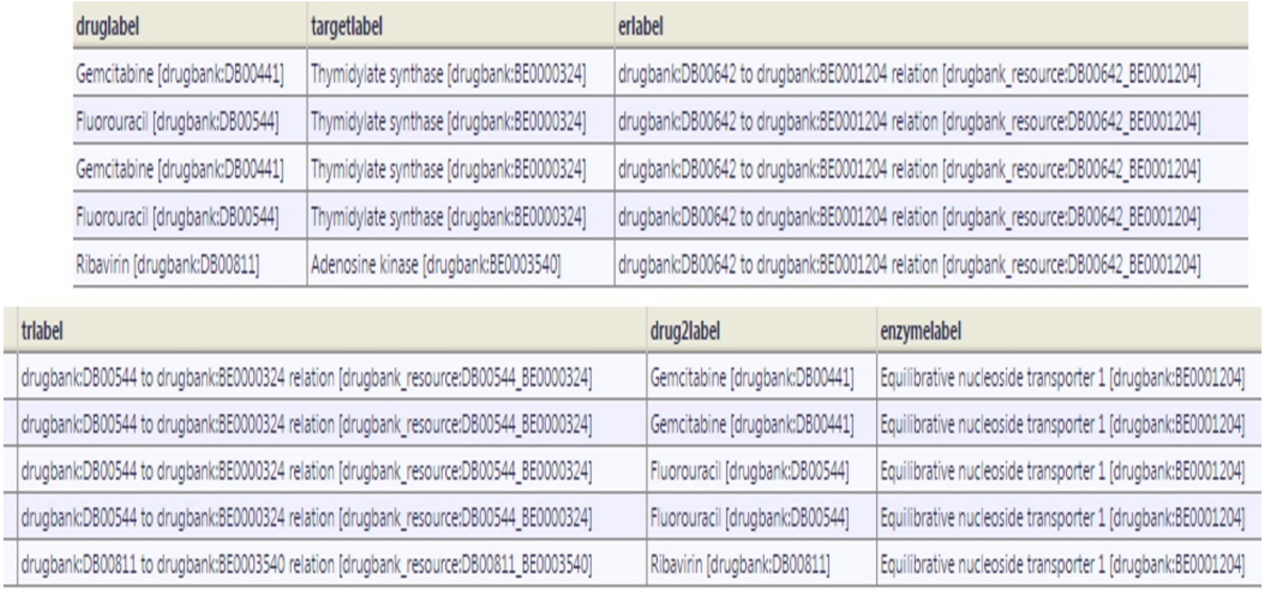


Figure 58: Topic 3_4 (T3_4) Graph in DrugBank.



```
select distinct ?druglabel, ?targetlabel, ?erlabel, ?trlabel, ?drug2label, ?enzymelabel where {
?drug <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Drug> .
?target <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Target> .
?drug <http://bio2rdf.org/drugbank_vocabulary:target> ?target .
?drug <http://bio2rdf.org/drugbank_vocabulary:transporter> ?enzyme .
?er <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Enzyme-Relation> .
?tr <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Target-Relation> .
?er <http://bio2rdf.org/drugbank_vocabulary:enzyme> ?enzyme .
?tr <http://bio2rdf.org/drugbank_vocabulary:enzyme> ?target.
?drug2 <http://bio2rdf.org/drugbank_vocabulary:target> ?target .
?drug2 <http://bio2rdf.org/drugbank_vocabulary:transporter> ?enzyme .
?drug <http://www.w3.org/2000/01/rdf-schema#label> ?druglabel.
?target <http://www.w3.org/2000/01/rdf-schema#label> ?targetlabel.
?er <http://www.w3.org/2000/01/rdf-schema#label> ?erlabel.
?tr <http://www.w3.org/2000/01/rdf-schema#label> ?trlabel.
?drug2 <http://www.w3.org/2000/01/rdf-schema#label> ?drug2label.
?enzyme <http://www.w3.org/2000/01/rdf-schema#label> ?enzymelabel. }
```

Figure 59: Query-1 SPARQL in DrugBank.



druglabel	targetlabel	erlabel	trlabel	drug2label	enzymelabel
Gemcitabine [drugbank:DB00441]	Thymidylate synthase [drugbank:BE0000324]	drugbank:DB00642 to drugbank:BE0001204 relation [drugbank_resource:DB00642_BE0001204]	drugbank:DB00544 to drugbank:BE0000324 relation [drugbank_resource:DB00544_BE0000324]	Gemcitabine [drugbank:DB00441]	Equilibrative nucleoside transporter 1 [drugbank:BE0001204]
Fluorouracil [drugbank:DB00544]	Thymidylate synthase [drugbank:BE0000324]	drugbank:DB00642 to drugbank:BE0001204 relation [drugbank_resource:DB00642_BE0001204]	drugbank:DB00544 to drugbank:BE0000324 relation [drugbank_resource:DB00544_BE0000324]	Gemcitabine [drugbank:DB00441]	Equilibrative nucleoside transporter 1 [drugbank:BE0001204]
Gemcitabine [drugbank:DB00441]	Thymidylate synthase [drugbank:BE0000324]	drugbank:DB00642 to drugbank:BE0001204 relation [drugbank_resource:DB00642_BE0001204]	drugbank:DB00544 to drugbank:BE0000324 relation [drugbank_resource:DB00544_BE0000324]	Fluorouracil [drugbank:DB00544]	Equilibrative nucleoside transporter 1 [drugbank:BE0001204]
Fluorouracil [drugbank:DB00544]	Thymidylate synthase [drugbank:BE0000324]	drugbank:DB00642 to drugbank:BE0001204 relation [drugbank_resource:DB00642_BE0001204]	drugbank:DB00544 to drugbank:BE0000324 relation [drugbank_resource:DB00544_BE0000324]	Fluorouracil [drugbank:DB00544]	Equilibrative nucleoside transporter 1 [drugbank:BE0001204]
Ribavirin [drugbank:DB00811]	Adenosine kinase [drugbank:BE0003540]	drugbank:DB00642 to drugbank:BE0001204 relation [drugbank_resource:DB00642_BE0001204]	drugbank:DB00811 to drugbank:BE0003540 relation [drugbank_resource:DB00811_BE0003540]	Ribavirin [drugbank:DB00811]	Equilibrative nucleoside transporter 1 [drugbank:BE0001204]

Figure 60: Query Results of Query-1 in DrugBank.

The Query-1 results include Gemcitabine, Fluorouracil, Ribavirin as the relevant drugs, Thymidylate synthase and Adenosine kinaseas as the target and Equilibrative nucleoside transporter 1 as the enzyme. Fig. 60 shows the partial outputs of Query-1.

Rank 2: Topic 3_7 (T3_7): This graph is composed of 7 concepts represented as a circle and 3 predicates as a triangle with 31 in-degree and out-degree. In T3_7, three predicates, drug, action, reference, whose in-degree and out-degree are 14, 11, and 6, respectively, are all nicely connected with 7 concepts. The predicates drug and action are ranked at 6th, 9th and many of the concepts in this topic are ranked among Top 20 Concepts. For T3_7, the rankings for Top 20 Concepts, Top 20 Predicates, Similarity, Silhouette Width, and Density are very good. T3_7's overall rank is 1st among 8 topics (together with T3_4). Fig. 61 shows the T3_7 topic graph. In this graph, concepts are represented as a circle, predicates as a triangle, and links as an arrow. In addition, the

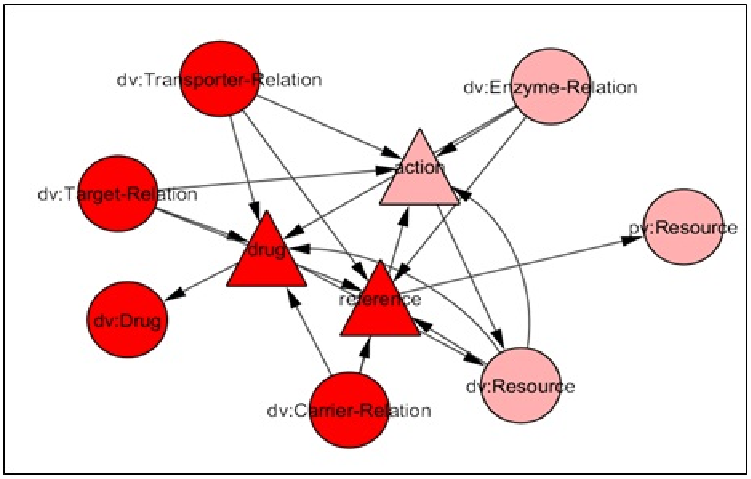
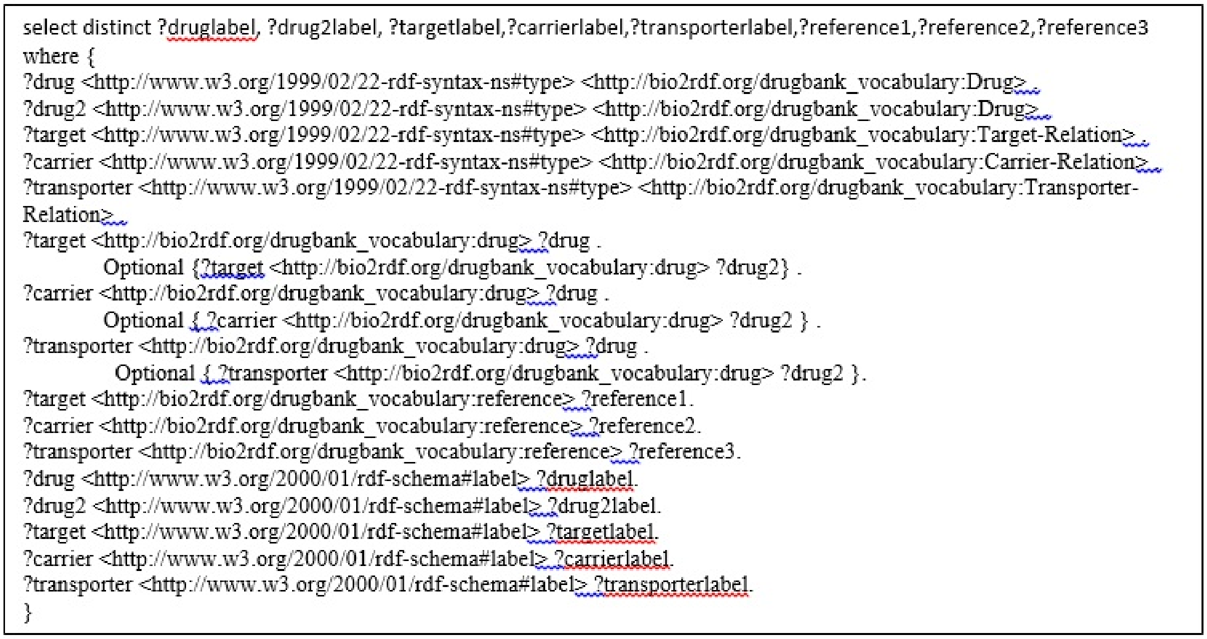


Figure 61: Topic 3_7 (T3_7) Graph in DrugBank.

dark red items like drug and reference are the predicates mentioned in Query-2

Query-2: The query graph is automatically generated from Topic 3_4 (one of the top ranked topics) to depict the query information. This query allows users to find the relevant drugs that have common Target-Relation, Carrier-Relation and Transporter-Relation and also provide their PubMed references for relations of target, transporter, and carrier with these drugs. The SPARQL format of Query-2 is automatically generated by the Query Generation algorithm considering the top predicates and their concepts in Fig. 62. This query can be translated as: *For any two drugs which share the common target-relation, carrier-relation and transporter-relation, what are all the possible combinations? What are the pubmed references for these target-relations, carrier-realtion and transporter-relation ?*

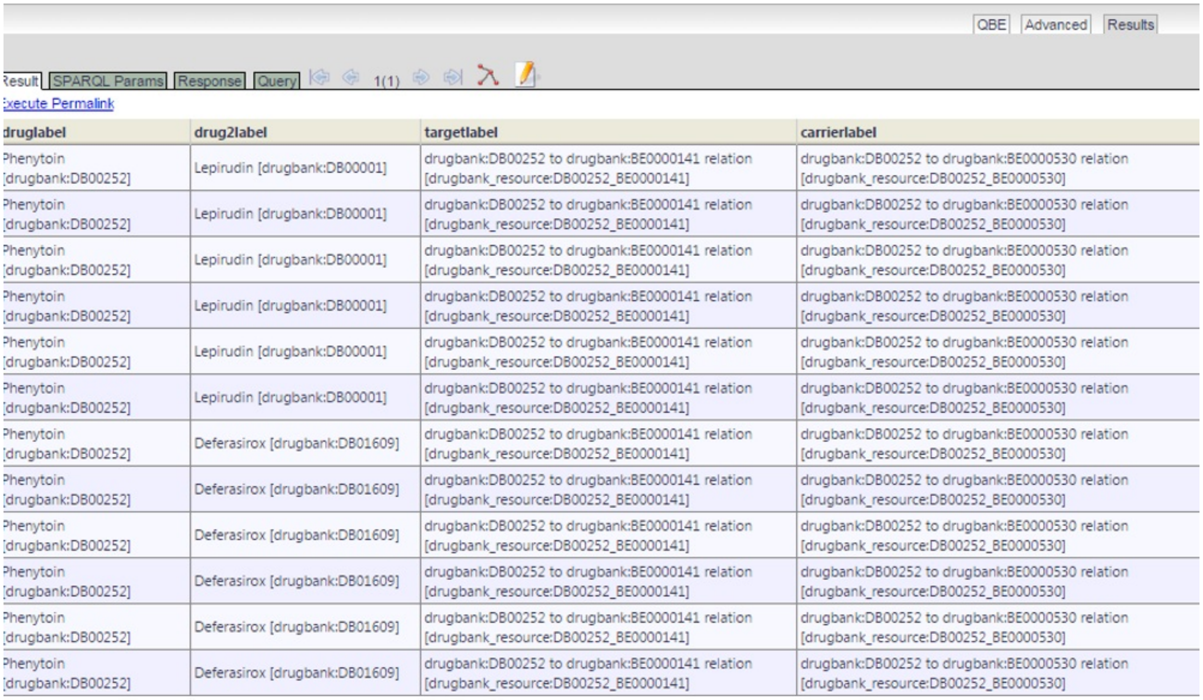
Fig. 63 shows the partial results from the query about some drugs like Phenytoin (DrugBank:DB00252), Lepirudin (DrugBank:DB00001) and Deferasirox (DrugBank:DB01609).



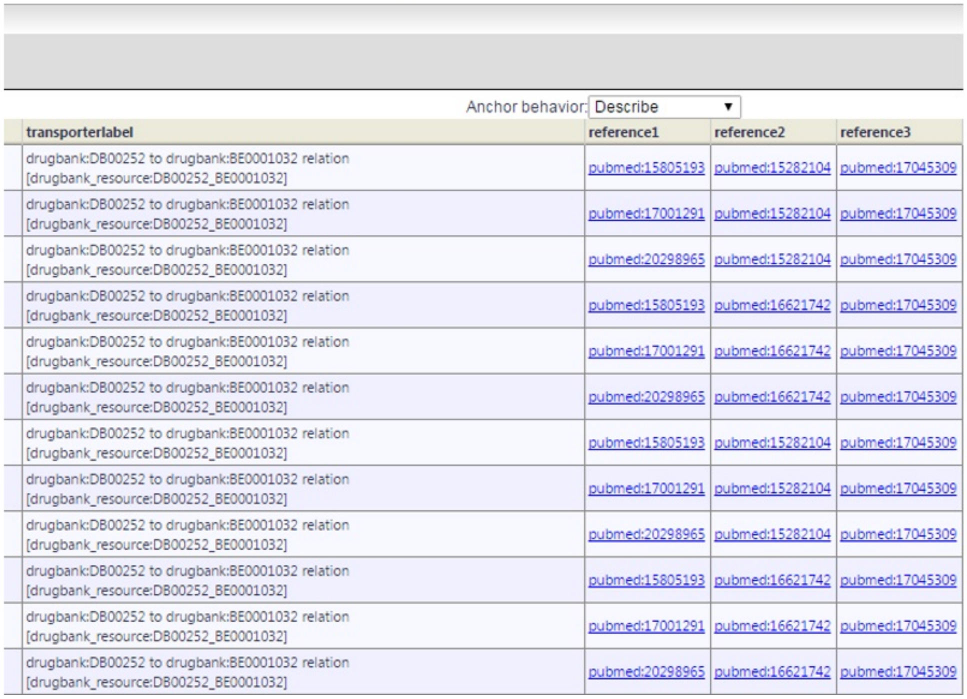
```
select distinct ?druglabel, ?drug2label, ?targetlabel,?carrierlabel,?transporterlabel,?reference1,?reference2,?reference3
where {
?drug <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Drug> .
?drug2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Drug> .
?target <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Target-Relation> .
?carrier <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Carrier-Relation> .
?transporter <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Transporter-
Relation> .
?target <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug .
        Optional {?target <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug2} .
?carrier <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug .
        Optional { ?carrier <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug2 } .
?transporter <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug .
         Optional { ?transporter <http://bio2rdf.org/drugbank_vocabulary:drug> ?drug2 }.
?target <http://bio2rdf.org/drugbank_vocabulary:reference> ?reference1.
?carrier <http://bio2rdf.org/drugbank_vocabulary:reference> ?reference2.
?transporter <http://bio2rdf.org/drugbank_vocabulary:reference> ?reference3.
?drug <http://www.w3.org/2000/01/rdf-schema#label> ?druglabel.
?drug2 <http://www.w3.org/2000/01/rdf-schema#label> ?drug2label.
?target <http://www.w3.org/2000/01/rdf-schema#label> ?targetlabel.
?carrier <http://www.w3.org/2000/01/rdf-schema#label> ?carrierlabel.
?transporter <http://www.w3.org/2000/01/rdf-schema#label> ?transporterlabel.
}
```

Figure 62: Query-2 SPARQL in DrugBank.

Rank 3: Topic 3_6 (T3_6): This topic consists of 3 predicates and 34 concepts with a very high sum of in-degree and out-degree, 150. Fig. 64 shows the T3_6 topic graph in which concepts are represented as a circle, predicates as a triangle, and links as an arrow. In particular, there are two subgraphs; one is with two predicates such as source and calculated-properties with concepts dv:Boiling-Point and dv:Bioavailability, respectively. Another predicate experimental-properties is connected with concepts such as dv:Water-Solubility. Specifically, T3_6 highly ranked in Top 20 Predicates and Silhouette Width while being lowly ranked in Similarity. This means each predicate has their own concepts while having the least common concepts with other predicates. Since the similarity ranking of this topic is low, the shared information is limited. Interestingly, this



druglabel	drug2label	targetlabel	carrierlabel
Phenytoin [drugbank:DB00252]	Lepirudin [drugbank:DB00001]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Lepirudin [drugbank:DB00001]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Lepirudin [drugbank:DB00001]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Lepirudin [drugbank:DB00001]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Lepirudin [drugbank:DB00001]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Lepirudin [drugbank:DB00001]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Deferasirox [drugbank:DB01609]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Deferasirox [drugbank:DB01609]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Deferasirox [drugbank:DB01609]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Deferasirox [drugbank:DB01609]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Deferasirox [drugbank:DB01609]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]
Phenytoin [drugbank:DB00252]	Deferasirox [drugbank:DB01609]	drugbank:DB00252 to drugbank:BE0000141 relation [drugbank_resource:DB00252_BE0000141]	drugbank:DB00252 to drugbank:BE0000530 relation [drugbank_resource:DB00252_BE0000530]



transporterlabel	reference1	reference2	reference3
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:15805193	pubmed:15282104	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:17001291	pubmed:15282104	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:20298965	pubmed:15282104	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:15805193	pubmed:16621742	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:17001291	pubmed:16621742	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:20298965	pubmed:16621742	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:15805193	pubmed:15282104	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:17001291	pubmed:15282104	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:20298965	pubmed:15282104	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:15805193	pubmed:16621742	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:17001291	pubmed:16621742	pubmed:17045309
drugbank:DB00252 to drugbank:BE0001032 relation [drugbank_resource:DB00252_BE0001032]	pubmed:20298965	pubmed:16621742	pubmed:17045309

Figure 63: Results of Query-2 in DrugBank.

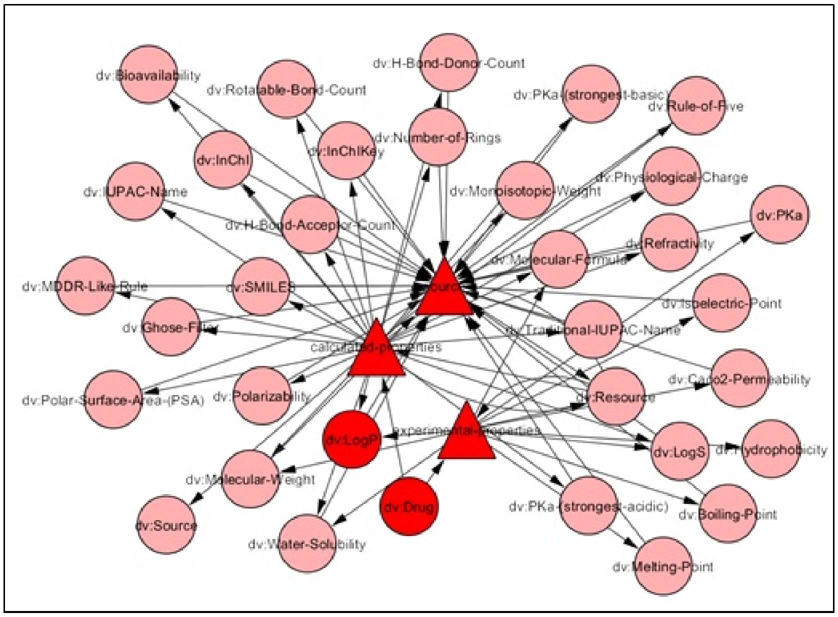
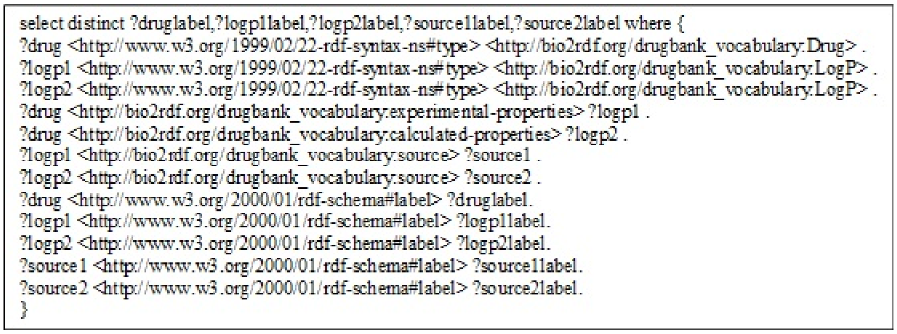


Figure 64: Topic 3_6 (T3_6) Graph in DrugBank.

graph shows a connection pattern from dv:experimental-properties to dv: source. The overall rank is 3rd among the eight topics. In this graph, the dark red items like source and calculated-properties are the predicates mentioned in Query-3.

Query-3. The query graph is automatically generated from Topic 3_6 to depict the query information. This query allows users to find drugs and all their experimental properties and calculated properties which have LogP experimental properties (octanol-water partition coefficient). The SPARQL format of Query-3 is automatically generated by the Query Generation algorithm considering the top predicates and their concepts in Fig. 65. This query can be translated as: *For any drug, what are all its experimental properties and calculated properties which contain octanol-water partition coefficient?*



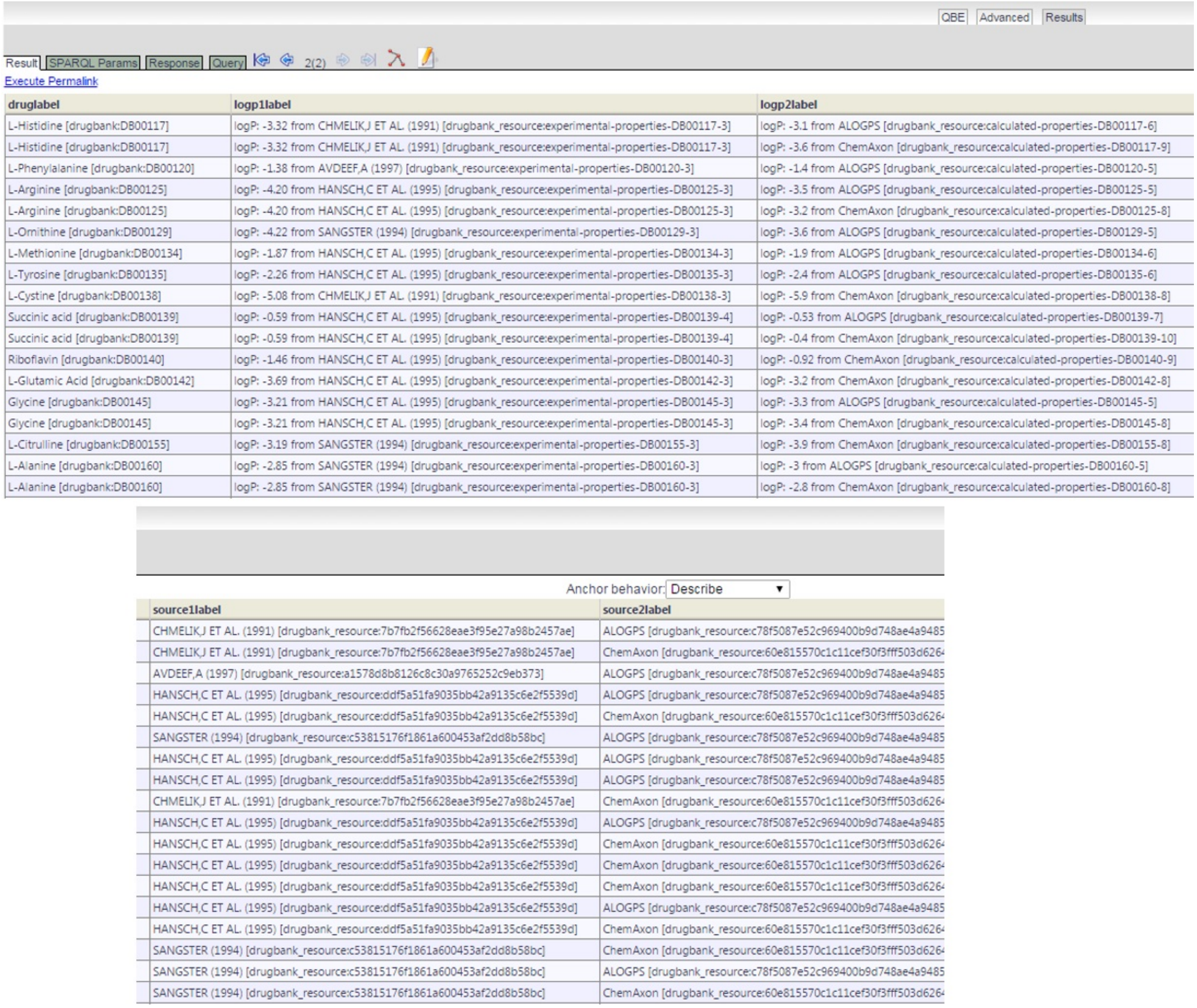
```
select distinct ?druglabel,?logp1label,?logp2label,?source1label,?source2label where {
?drug <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Drug> .
?logp1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:LogP> .
?logp2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:LogP> .
?drug <http://bio2rdf.org/drugbank_vocabulary:experimental-properties> ?logp1 .
?drug <http://bio2rdf.org/drugbank_vocabulary:calculated-properties> ?logp2 .
?logp1 <http://bio2rdf.org/drugbank_vocabulary:source> ?source1 .
?logp2 <http://bio2rdf.org/drugbank_vocabulary:source> ?source2 .
?drug <http://www.w3.org/2000/01/rdf-schema#label> ?druglabel.
?logp1 <http://www.w3.org/2000/01/rdf-schema#label> ?logp1label.
?logp2 <http://www.w3.org/2000/01/rdf-schema#label> ?logp2label.
?source1 <http://www.w3.org/2000/01/rdf-schema#label> ?source1label.
?source2 <http://www.w3.org/2000/01/rdf-schema#label> ?source2label.
}
```

Figure 65: Query-3 SPARQL in DrugBank.

As the Query-3 results, the relevant drug and their experimental and calculated-properties are reported as (L-Histidine, logP: -3.32 from CHMELIK,J ET AL. (1991), logP: -3.1 from ALOGPS) and (L-Phenylalanine, logP: -1.38 from AVDEEF,A (1997), logP: -1.4 from ALOGPS). Fig. 66 shows the details of the partial results from Query-3.

Rank 4: Topic 3_2 (T3_2): The unique pattern in T3_2 is two dominant concepts, Resource and Drug, whose in-degree and out-degree are 46 and 33, respectively, are fully connected to the remaining 17 concepts via 20 different predicates such as absorption, protein-binding. The rankings for Top 20 Concepts and Silhouette Width are relatively good while the rankings for Top 20 Predicates and Density are poor. The overall rank is 4th among the eight topics. Fig. 67 shows the T3_2 topic graph. In this graph, concepts are represented as a circle, predicates as a triangle, and links as an arrow. The dark red items like abortion and product are the predicates and dv:Drug and dv:Pharmaceutical are the concepts mentioned in Query-4.

Query-4: The query graph is automatically generated from Topic 3_2 to depict the topic and query information. This query allows users to find drugs and their absorption,



druglabel	logp1label	logp2label
L-Histidine [drugbank:DB00117]	logP: -3.32 from CHMELIK,J ET AL. (1991) [drugbank_resource:experimental-properties-DB00117-3]	logP: -3.1 from ALOGPS [drugbank_resource:calculated-properties-DB00117-6]
L-Histidine [drugbank:DB00117]	logP: -3.32 from CHMELIK,J ET AL. (1991) [drugbank_resource:experimental-properties-DB00117-3]	logP: -3.6 from ChemAxon [drugbank_resource:calculated-properties-DB00117-9]
L-Phenylalanine [drugbank:DB00120]	logP: -1.38 from AVDEEF,A (1997) [drugbank_resource:experimental-properties-DB00120-3]	logP: -1.4 from ALOGPS [drugbank_resource:calculated-properties-DB00120-5]
L-Arginine [drugbank:DB00125]	logP: -4.20 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00125-3]	logP: -3.5 from ALOGPS [drugbank_resource:calculated-properties-DB00125-5]
L-Arginine [drugbank:DB00125]	logP: -4.20 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00125-3]	logP: -3.2 from ChemAxon [drugbank_resource:calculated-properties-DB00125-8]
L-Ornithine [drugbank:DB00129]	logP: -4.22 from SANGSTER (1994) [drugbank_resource:experimental-properties-DB00129-3]	logP: -3.6 from ALOGPS [drugbank_resource:calculated-properties-DB00129-5]
L-Methionine [drugbank:DB00134]	logP: -1.87 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00134-3]	logP: -1.9 from ALOGPS [drugbank_resource:calculated-properties-DB00134-6]
L-Tyrosine [drugbank:DB00135]	logP: -2.26 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00135-3]	logP: -2.4 from ALOGPS [drugbank_resource:calculated-properties-DB00135-6]
L-Cystine [drugbank:DB00138]	logP: -5.08 from CHMELIK,J ET AL. (1991) [drugbank_resource:experimental-properties-DB00138-3]	logP: -5.9 from ChemAxon [drugbank_resource:calculated-properties-DB00138-8]
Succinic acid [drugbank:DB00139]	logP: -0.59 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00139-4]	logP: -0.53 from ALOGPS [drugbank_resource:calculated-properties-DB00139-7]
Succinic acid [drugbank:DB00139]	logP: -0.59 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00139-4]	logP: -0.4 from ChemAxon [drugbank_resource:calculated-properties-DB00139-10]
Riboflavin [drugbank:DB00140]	logP: -1.46 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00140-3]	logP: -0.92 from ChemAxon [drugbank_resource:calculated-properties-DB00140-9]
L-Glutamic Acid [drugbank:DB00142]	logP: -3.69 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00142-3]	logP: -3.2 from ChemAxon [drugbank_resource:calculated-properties-DB00142-8]
Glycine [drugbank:DB00145]	logP: -3.21 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00145-3]	logP: -3.3 from ALOGPS [drugbank_resource:calculated-properties-DB00145-5]
Glycine [drugbank:DB00145]	logP: -3.21 from HANSCH,C ET AL. (1995) [drugbank_resource:experimental-properties-DB00145-3]	logP: -3.4 from ChemAxon [drugbank_resource:calculated-properties-DB00145-8]
L-Citrulline [drugbank:DB00155]	logP: -3.19 from SANGSTER (1994) [drugbank_resource:experimental-properties-DB00155-3]	logP: -3.9 from ChemAxon [drugbank_resource:calculated-properties-DB00155-8]
L-Alanine [drugbank:DB00160]	logP: -2.85 from SANGSTER (1994) [drugbank_resource:experimental-properties-DB00160-3]	logP: -3 from ALOGPS [drugbank_resource:calculated-properties-DB00160-5]
L-Alanine [drugbank:DB00160]	logP: -2.85 from SANGSTER (1994) [drugbank_resource:experimental-properties-DB00160-3]	logP: -2.8 from ChemAxon [drugbank_resource:calculated-properties-DB00160-8]

Anchor behavior: Describe ▾

source1label	source2label
CHMELIK,J ET AL. (1991) [drugbank_resource:7b7fb2f56628eae3f95e27a98b2457ae]	ALOGPS [drugbank_resource:c78f5087e52c969400b9d748ae4a9485
CHMELIK,J ET AL. (1991) [drugbank_resource:7b7fb2f56628eae3f95e27a98b2457ae]	ChemAxon [drugbank_resource:60e815570c1c11cef30f3fff503d626
AVDEEF,A (1997) [drugbank_resource:a1578d8b8126c8c30a9765252c9eb373]	ALOGPS [drugbank_resource:c78f5087e52c969400b9d748ae4a9485
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ALOGPS [drugbank_resource:c78f5087e52c969400b9d748ae4a9485
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ChemAxon [drugbank_resource:60e815570c1c11cef30f3fff503d626
SANGSTER (1994) [drugbank_resource:c53815176f1861a600453af2dd8b58bc]	ALOGPS [drugbank_resource:c78f5087e52c969400b9d748ae4a9485
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ALOGPS [drugbank_resource:c78f5087e52c969400b9d748ae4a9485
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ALOGPS [drugbank_resource:c78f5087e52c969400b9d748ae4a9485
CHMELIK,J ET AL. (1991) [drugbank_resource:7b7fb2f56628eae3f95e27a98b2457ae]	ChemAxon [drugbank_resource:60e815570c1c11cef30f3fff503d626
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ALOGPS [drugbank_resource:c78f5087e52c969400b9d748ae4a9485
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ChemAxon [drugbank_resource:60e815570c1c11cef30f3fff503d626
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ChemAxon [drugbank_resource:60e815570c1c11cef30f3fff503d626
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ChemAxon [drugbank_resource:60e815570c1c11cef30f3fff503d626
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ALOGPS [drugbank_resource:c78f5087e52c969400b9d748ae4a9485
HANSCH,C ET AL. (1995) [drugbank_resource:ddf5a51fa9035bb42a9135c6e2f5539d]	ChemAxon [drugbank_resource:60e815570c1c11cef30f3fff503d626
SANGSTER (1994) [drugbank_resource:c53815176f1861a600453af2dd8b58bc]	ChemAxon [drugbank_resource:60e815570c1c11cef30f3fff503d626
SANGSTER (1994) [drugbank_resource:c53815176f1861a600453af2dd8b58bc]	ALOGPS [drugbank_resource:c78f5087e52c969400b9d748ae4a9485
SANGSTER (1994) [drugbank_resource:c53815176f1861a600453af2dd8b58bc]	ChemAxon [drugbank_resource:60e815570c1c11cef30f3fff503d626

Figure 66: Results of Query-3 in DrugBank.

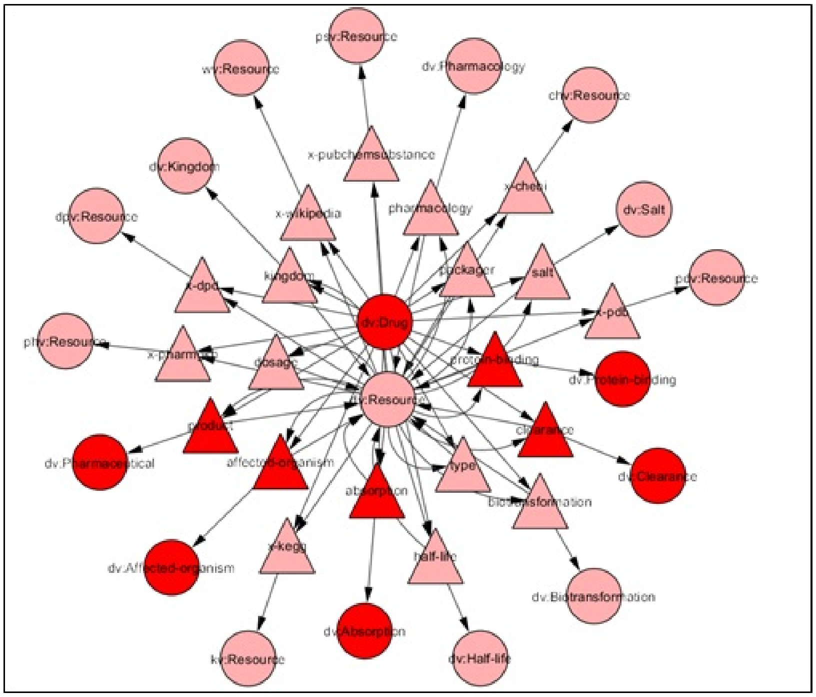
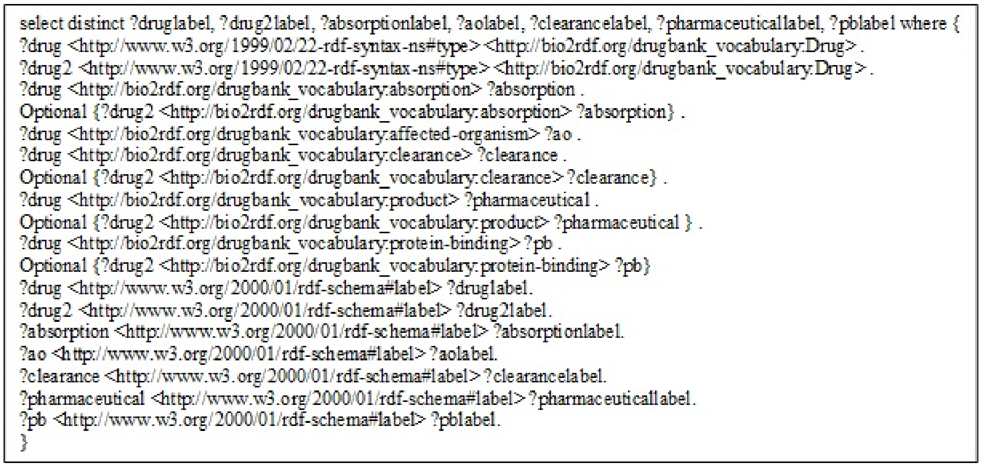


Figure 67: Topic 3_2 (T3_2) Graph in DrugBank.



```
select distinct ?druglabel, ?drug2label, ?absorptionlabel, ?aolabel, ?clearancelabel, ?pharmaceuticallabel, ?pblabel where {
?drug <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Drug> .
?drug2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://bio2rdf.org/drugbank_vocabulary:Drug> .
?drug <http://bio2rdf.org/drugbank_vocabulary:absorption> ?absorption .
Optional {?drug2 <http://bio2rdf.org/drugbank_vocabulary:absorption> ?absorption} .
?drug <http://bio2rdf.org/drugbank_vocabulary:affected-organism> ?ao .
?drug <http://bio2rdf.org/drugbank_vocabulary:clearance> ?clearance .
Optional {?drug2 <http://bio2rdf.org/drugbank_vocabulary:clearance> ?clearance} .
?drug <http://bio2rdf.org/drugbank_vocabulary:product> ?pharmaceutical .
Optional {?drug2 <http://bio2rdf.org/drugbank_vocabulary:product> ?pharmaceutical } .
?drug <http://bio2rdf.org/drugbank_vocabulary:protein-binding> ?pb .
Optional {?drug2 <http://bio2rdf.org/drugbank_vocabulary:protein-binding> ?pb}
?drug <http://www.w3.org/2000/01/rdf-schema#label> ?druglabel.
?drug2 <http://www.w3.org/2000/01/rdf-schema#label> ?drug2label.
?absorption <http://www.w3.org/2000/01/rdf-schema#label> ?absorptionlabel.
?ao <http://www.w3.org/2000/01/rdf-schema#label> ?aolabel.
?clearance <http://www.w3.org/2000/01/rdf-schema#label> ?clearancelabel.
?pharmaceutical <http://www.w3.org/2000/01/rdf-schema#label> ?pharmaceuticallabel.
?pb <http://www.w3.org/2000/01/rdf-schema#label> ?pblabel.
}
```

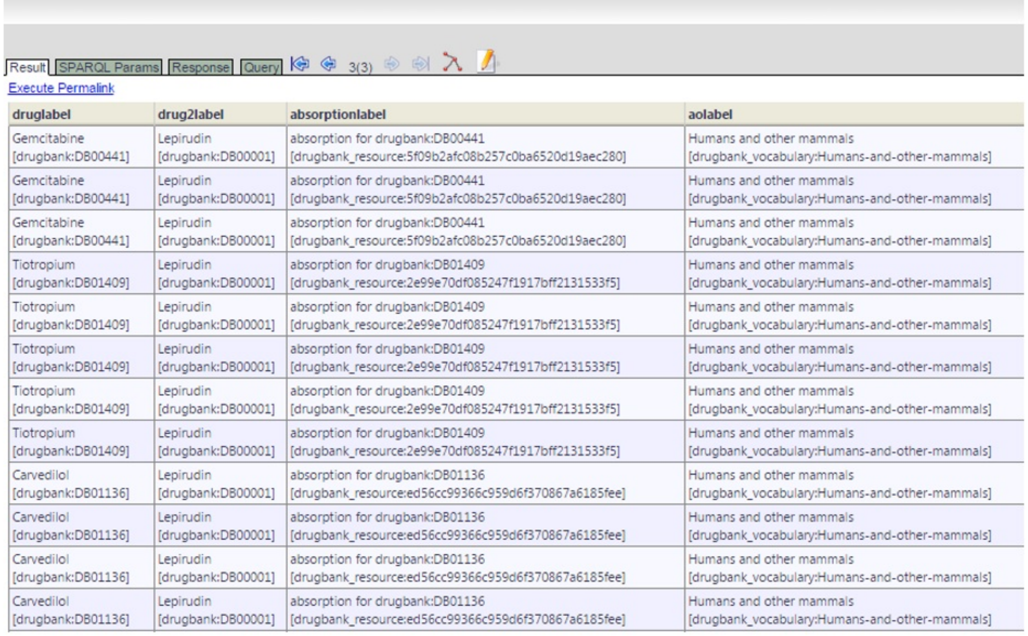
Figure 68: Query-4 SPARQL in DrugBank.

affected-organism, clearance pharmacokinetic measurement, pharmaceutical information, and protein binding information. The SPARQL format of Query-4 is automatically generated by the Query Generation algorithm considering the top predicates and their concepts in Fig. 68. This query can be translated as: *For any two drugs which share the same absorption, affected-organism, clearance and pharmaceutical, what are all the possible combinations?*

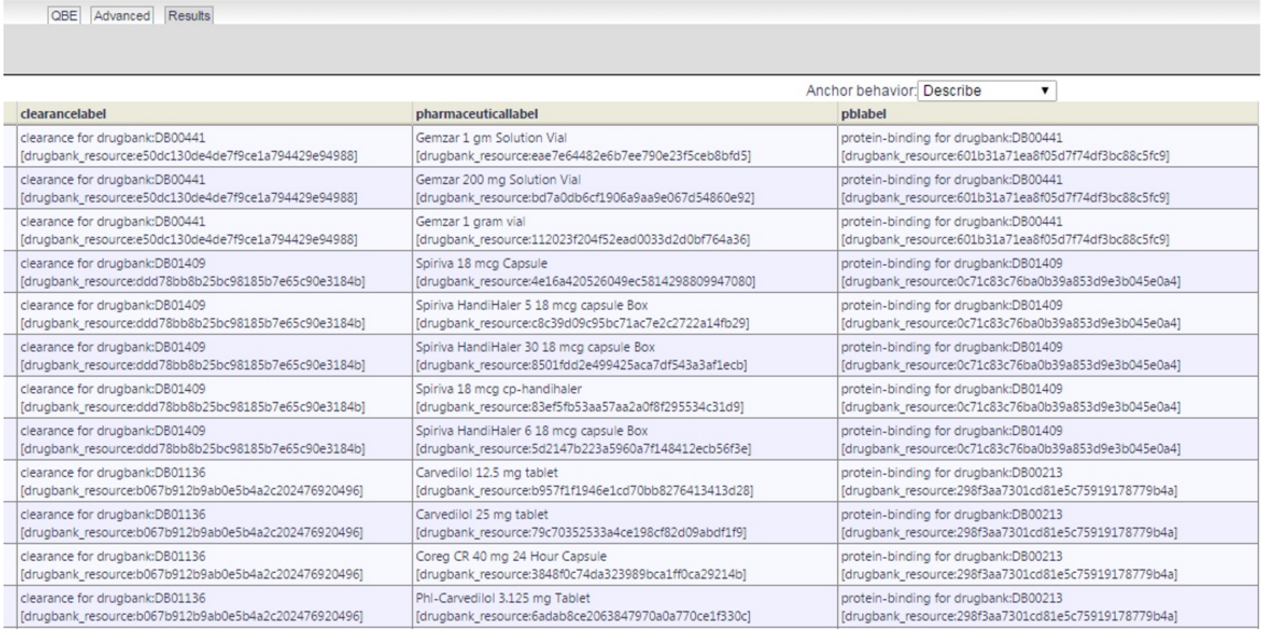
The Query-4 results on any relevant drugs and their pharmaceutical information include (Gemcitabine, Lepirudin, Gemzar 1 gm Solution Vial), (Tiotropium, Lepirudin, Spiriva 18 mcg Capsule). Fig. 69 shows the partial results from Query-4.

4.7.2 Performance for Topic Aware Link Discovery

We conduct several comparison experiments between the GraphKDD and existing path finding and query processing tools to validate the optimized solution provided by the



druglabel	drug2label	absorptionlabel	aolabel
Gemcitabine [drugbank:DB00441]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB00441 [drugbank_resource:5f09b2afc08b257c0ba6520d19aec280]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Gemcitabine [drugbank:DB00441]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB00441 [drugbank_resource:5f09b2afc08b257c0ba6520d19aec280]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Gemcitabine [drugbank:DB00441]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB00441 [drugbank_resource:5f09b2afc08b257c0ba6520d19aec280]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Tiotropium [drugbank:DB01409]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB01409 [drugbank_resource:2e99e70df085247f1917bff2131533f5]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Tiotropium [drugbank:DB01409]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB01409 [drugbank_resource:2e99e70df085247f1917bff2131533f5]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Tiotropium [drugbank:DB01409]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB01409 [drugbank_resource:2e99e70df085247f1917bff2131533f5]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Tiotropium [drugbank:DB01409]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB01409 [drugbank_resource:2e99e70df085247f1917bff2131533f5]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Tiotropium [drugbank:DB01409]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB01409 [drugbank_resource:2e99e70df085247f1917bff2131533f5]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Carvedilol [drugbank:DB01136]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB01136 [drugbank_resource:ed56cc99366c959d6f370867a6185fee]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Carvedilol [drugbank:DB01136]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB01136 [drugbank_resource:ed56cc99366c959d6f370867a6185fee]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Carvedilol [drugbank:DB01136]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB01136 [drugbank_resource:ed56cc99366c959d6f370867a6185fee]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]
Carvedilol [drugbank:DB01136]	Lepirudin [drugbank:DB00001]	absorption for drugbank:DB01136 [drugbank_resource:ed56cc99366c959d6f370867a6185fee]	Humans and other mammals [drugbank_vocabulary:Humans-and-other-mammals]



Anchor behavior: Describe

clearancelabel	pharmaceuticallabel	pblabel
clearance for drugbank:DB00441 [drugbank_resource:e50dc130de4de7f9ce1a794429e94988]	Gemzar 1 gm Solution Vial [drugbank_resource:eae7e64482e6b7ee790e23f5ceb8bfd5]	protein-binding for drugbank:DB00441 [drugbank_resource:601b31a71ea8f05d7f74df3bc88c5fc9]
clearance for drugbank:DB00441 [drugbank_resource:e50dc130de4de7f9ce1a794429e94988]	Gemzar 200 mg Solution Vial [drugbank_resource:bd7a0db6cf1906a9aa9e067d54860e92]	protein-binding for drugbank:DB00441 [drugbank_resource:601b31a71ea8f05d7f74df3bc88c5fc9]
clearance for drugbank:DB00441 [drugbank_resource:e50dc130de4de7f9ce1a794429e94988]	Gemzar 1 gram vial [drugbank_resource:112023f204f52ead0033d2d0bf764a36]	protein-binding for drugbank:DB00441 [drugbank_resource:601b31a71ea8f05d7f74df3bc88c5fc9]
clearance for drugbank:DB01409 [drugbank_resource:ddd78bb8b25bc98185b7e65c90e3184b]	Spiriva 18 mcg Capsule [drugbank_resource:4e16a420526049ec5814298809947080]	protein-binding for drugbank:DB01409 [drugbank_resource:0c71c83c76ba0b39a853d9e3b045e0a4]
clearance for drugbank:DB01409 [drugbank_resource:ddd78bb8b25bc98185b7e65c90e3184b]	Spiriva HandiHaler 5 18 mcg capsule Box [drugbank_resource:c8c39d09c95bc71ac7e2c2722a14fb29]	protein-binding for drugbank:DB01409 [drugbank_resource:0c71c83c76ba0b39a853d9e3b045e0a4]
clearance for drugbank:DB01409 [drugbank_resource:ddd78bb8b25bc98185b7e65c90e3184b]	Spiriva HandiHaler 30 18 mcg capsule Box [drugbank_resource:8501fdd2e499425aca7df543a3af1ecb]	protein-binding for drugbank:DB01409 [drugbank_resource:0c71c83c76ba0b39a853d9e3b045e0a4]
clearance for drugbank:DB01409 [drugbank_resource:ddd78bb8b25bc98185b7e65c90e3184b]	Spiriva 18 mcg cp-handihaler [drugbank_resource:83ef5fb53aa57aa2a0f8f295534c31d9]	protein-binding for drugbank:DB01409 [drugbank_resource:0c71c83c76ba0b39a853d9e3b045e0a4]
clearance for drugbank:DB01409 [drugbank_resource:ddd78bb8b25bc98185b7e65c90e3184b]	Spiriva HandiHaler 6 18 mcg capsule Box [drugbank_resource:5d2147b223a5960a7f148412ecb56f3e]	protein-binding for drugbank:DB01409 [drugbank_resource:0c71c83c76ba0b39a853d9e3b045e0a4]
clearance for drugbank:DB01136 [drugbank_resource:b067b912b9ab0e5b4a2c202476920496]	Carvedilol 12.5 mg tablet [drugbank_resource:b957f1f1946e1cd70bb8276413413d28]	protein-binding for drugbank:DB00213 [drugbank_resource:298f3aa7301cd81e5c7591917877 9b4a]
clearance for drugbank:DB01136 [drugbank_resource:b067b912b9ab0e5b4a2c202476920496]	Carvedilol 25 mg tablet [drugbank_resource:79c70352533a4ce198cf82d09abdf1f9]	protein-binding for drugbank:DB00213 [drugbank_resource:298f3aa7301cd81e5c7591917877 9b4a]
clearance for drugbank:DB01136 [drugbank_resource:b067b912b9ab0e5b4a2c202476920496]	Coreg CR 40 mg 24 Hour Capsule [drugbank_resource:3848f0c74da323989bca1ff0ca29214b]	protein-binding for drugbank:DB00213 [drugbank_resource:298f3aa7301cd81e5c7591917877 9b4a]
clearance for drugbank:DB01136 [drugbank_resource:b067b912b9ab0e5b4a2c202476920496]	Phl-Carvedilol 3.125 mg Tablet [drugbank_resource:6adab8ce2063847970a0a770ce1f330c]	protein-binding for drugbank:DB00213 [drugbank_resource:298f3aa7301cd81e5c7591917877 9b4a]

Figure 69: Results of Query-4 in DrugBank.

GraphKDD.

4.7.2.1 GraphKDD vs. LIMES

In this section, we make a comparison study between GraphKDD and LIMES link discovery tool in terms of running performance and context aware ability. GraphKDD is able to find paths between source and target nodes efficiently with topic context. In this experiment, we include 4 different datasets and their topic generation results: DBpdia HPKM, YAGO HPKM, Bio2RDF HPKM and Bio2RDF PHAL. For each topic result, we randomly select 10% source/target pairs and test with both GraphKDD and LIMES. Specifically, we test their preprocessing time, processing time, topic context awareness. Because LIMES has four different optimization to measure the similarity (Trigram, Cosine, Jaccard and Levenshtein Siilarity), we compare GraphKDD with each optimization solution. Fig. 70 shows the size for each dataset. YAGO is the smallest dataset in terms of predicate size, concept size and triple size, DBpedia has the most predicate while Bio2RDF has the most triple.

Correspondingly, we evaluate pre-processing running performance for each case as shown in Fig. 71. In general, pre-processing is divided into two parts: association matrix generation time and topic generation time. YAGO has the smallest size, so it has the minimum pre-processing time. Bio2RDF HPKM and Bio2RDF PHAL have the same association matrix generation time. However, Bio2RDF HPKM needs more time to generate topics because HPKM needs more time to run multiple iterations to generate hierarchy of topics. DBpedia has the most predicates, so it needs more time to generate

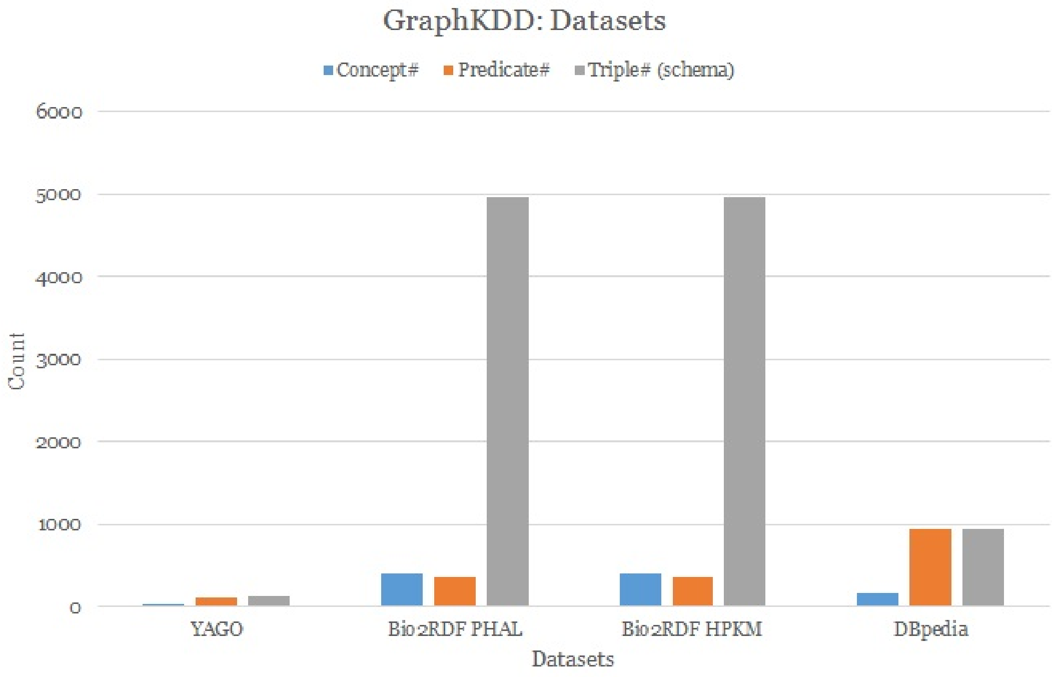


Figure 70: Size for Each dataset

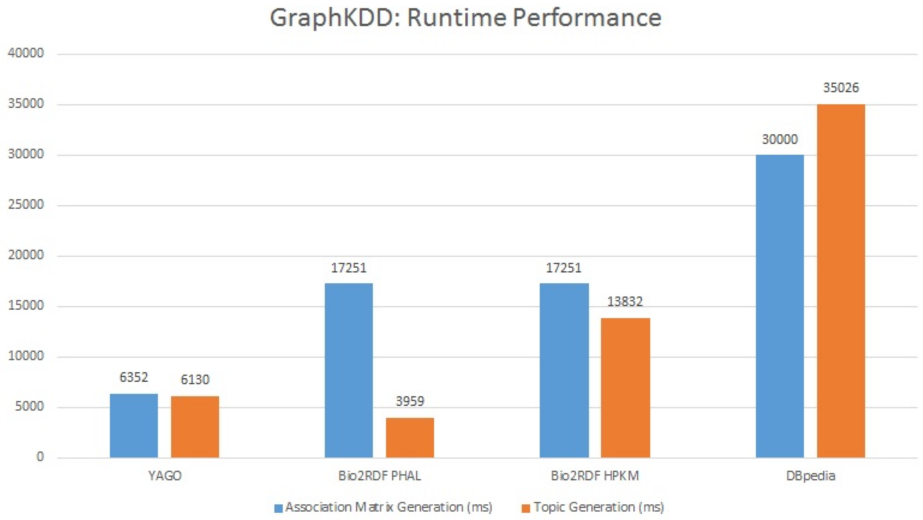


Figure 71: Pre-Processing Performance for GraphKDD

matrix as well as topics.

In addition, we compare pre-processing time between GraphKDD and LIMES. Fig. 72 gives the logarithmic time evaluation for each case. From the evaluation, it is obviously to see that GraphKDD needs more time to prepare association matrix and topic generation while LIMES needs less time to indexing datasets with four optimization solutions. The reason is that GraphKDD spends more time on collecting context information from all topics and which could be useful to recognize context for source and target nodes.

Furthermore, we compare processing time between GraphKDD and LIMES. Fig. 73 shows the time evaluation for each case. We find that GraphKDD has the best performance to find paths between source and target nodes compared to all 4 optimization similarity measurement provided by LIMES.

What is more, we also list topic awareness statistics for source/target finding for

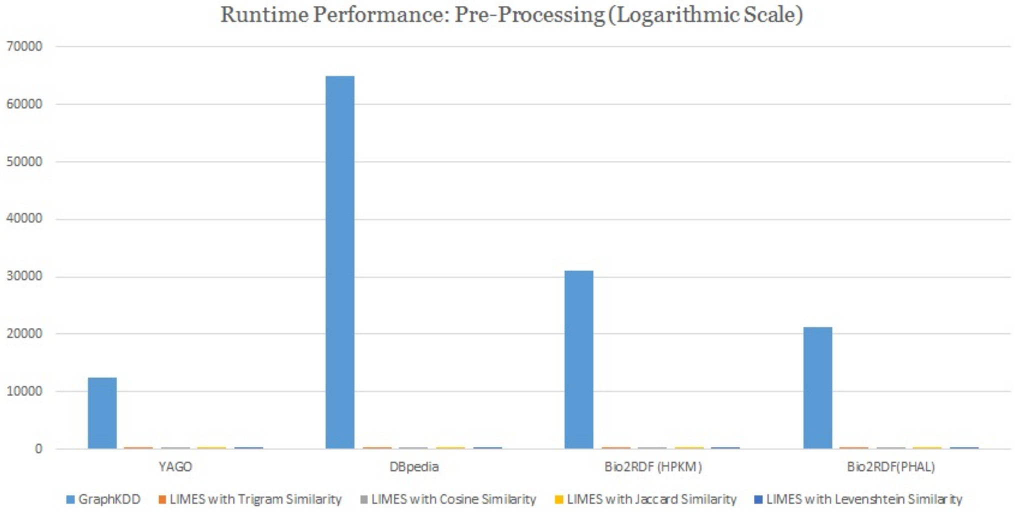


Figure 72: Pre-Processing Performance between GraphKDD and LIMES

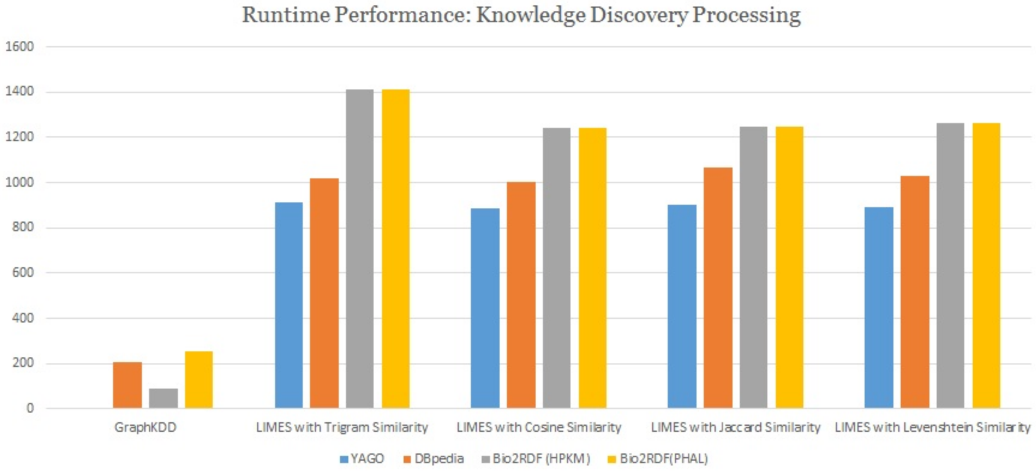


Figure 73: Processing Performance between GraphKDD and LIMES

Table 31: Topic Awareness Statistics for GraphKDD

	# Pairs (source, target)	**# Pairs in 1 topic**	**# Pairs in 2 topics**	**# Pairs in 3 topics**
DBpedia HPKM	8892	4880	2439	811
YAGO HPKM	1416	247	778	220
Bio2RDF HPKM	1600	932	358	107
Bio2RDF PHAL	1600	462	477	426

each dataset as Table 31 shows. DBpedia HPKM has the most pairs in one topic. YAGO HPKM has the most pairs in 2 topics. Bio2RDF HPKM has the most pairs in one topic and Bio2RDF PHAL has the most pairs in 2 topics.

4.7.2.2 GraphKDD vs. SLAP

We also conduct an experiment combining query generation and path finding. As a comparison, we introduce the Semantic Link Association Prediction (SLAP) [19], which is a drug target association detection framework published on 2012. Datasets covered by the SLAP are Pubchem, CHEBI, DrugBank, UniProt, UniProtKB-GOA, HGNC, SIDER, OMIM, KEGG, HPRD, ChEMBL, TTD, BindingDB, CTD and POSP. We use Bio2RDF 9 domains dataset to demonstrate such comparison, which has 6 datasets (DrugBank, HGNC, SIDER, OMIM, KEGG, CTD) overlap with datasets that the SLAP uses. The detailed steps of comparison are described as follow.

Step1: Because we focus on the results of interaction and association between drug and target, we first try to find how many topics are necessary to cover both drug, target and gene information. From Bio2RDF 9 domains topic distribution outputs, we find that Topic 16 and 27 contain both concepts $dv : Drug(SIO\_010038)$, $dv : Target(SIO\_010423)$

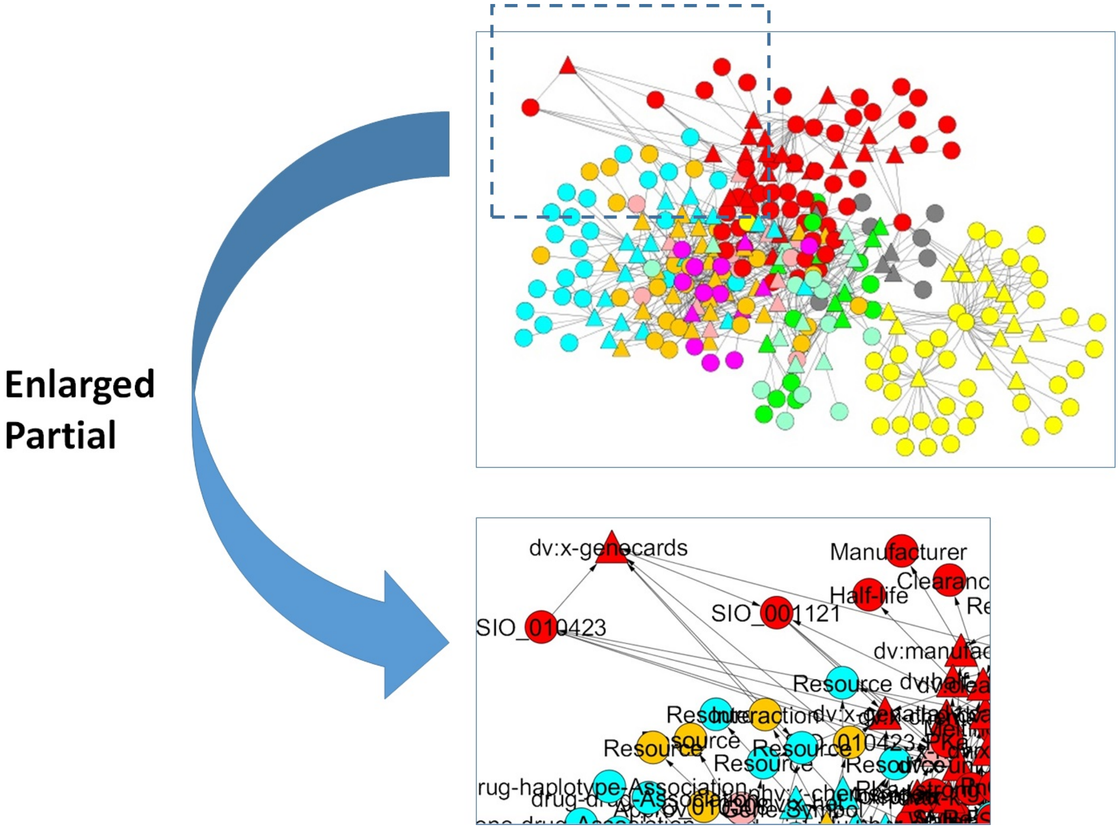


Figure 74: Bio2RDF 9 Domains Topic 16

and $dv : Gene(SIO\_001121)$ in schema level. Fig. 74 and 75 show specific topic visualization for Topic 16 and 17 respectively. In addition, from predicates' perspective, to find the link among $dv : Drug(SIO\_010038)$, $dv : Target(SIO\_010423)$ and $dv : Gene(SIO\_001121)$ is about finding the path between predicates $dv : target$ and $dv : x - genecards$ as shown in Fig. 76. The path shown in Fig. 76 is $dv : Drug(SIO\_010038) - > dv : target - > dv : Target(SIO\_010423) - > dv : x - genecards - > dv : Gene(SIO\_001121)$.

Step2: A specific query related to $dv : Drug(SIO\_010038)$, $dv : Target(SIO\_010423)$

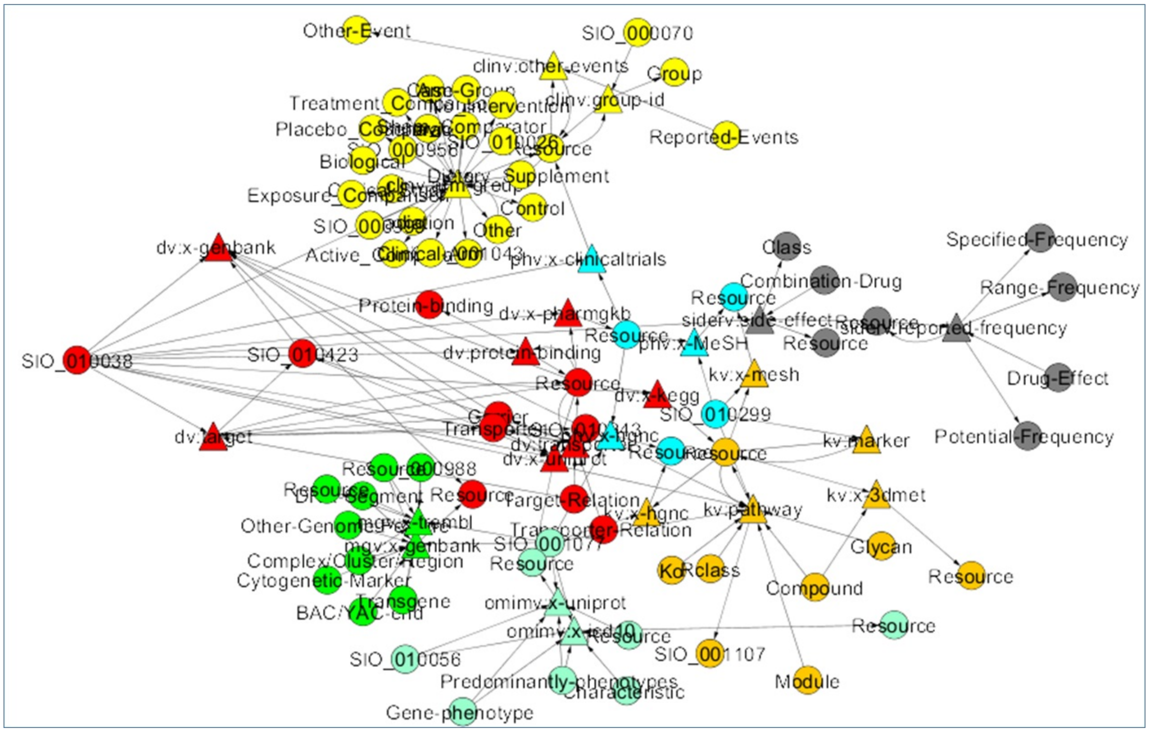


Figure 75: Bio2RDF 9 Domains Topic 27

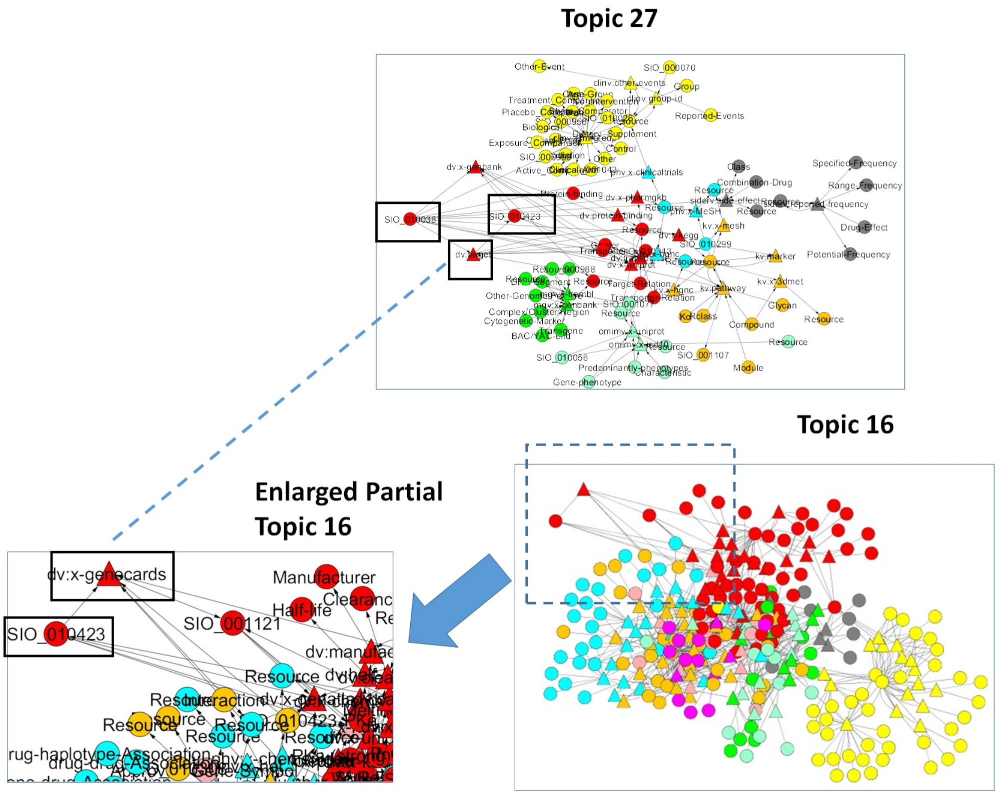
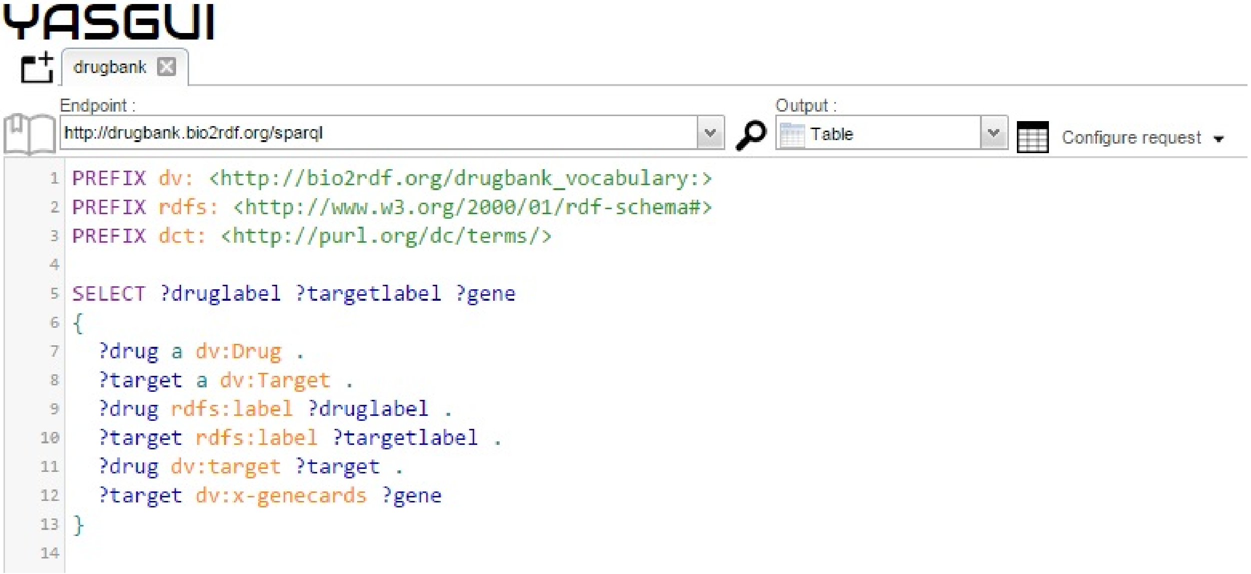


Figure 76: Path between Topic 16 and 27

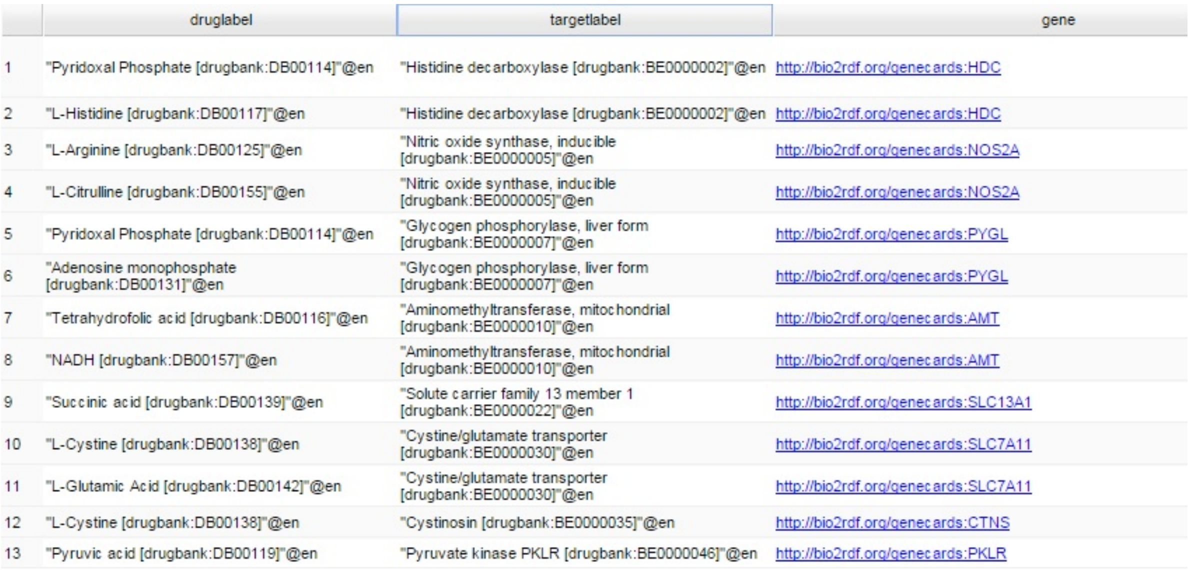


```
PREFIX dv: <http://bio2rdf.org/drugbank_vocabulary:>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dct: <http://purl.org/dc/terms/>

SELECT ?druglabel ?targetlabel ?gene
{
  ?drug a dv:Drug .
  ?target a dv:Target .
  ?drug rdfs:label ?druglabel .
  ?target rdfs:label ?targetlabel .
  ?drug dv:target ?target .
  ?target dv:x-genecards ?gene
}

```

Figure 77: Drug Target Association Query.

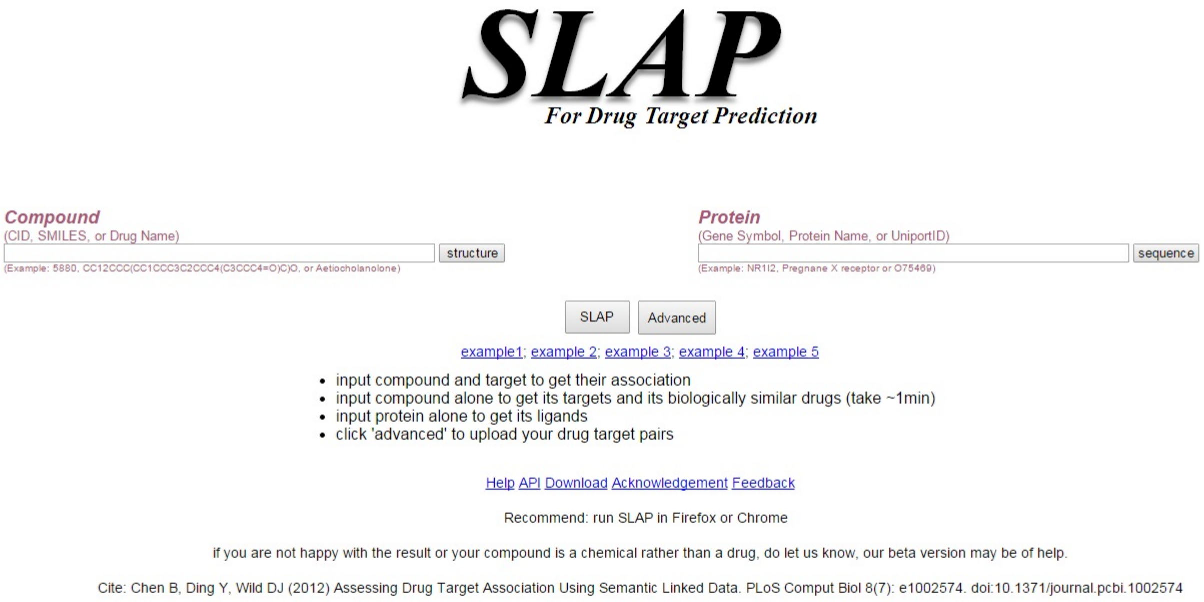
and $dv : Gene(SIO\_001121)$ is generated for Topic 16 and 27. An example is shown in Fig. 77 with YASGUI query endpoint ($http : //legacy.yasgui.org/$). Partial query outputs are also shown in Fig. 78 with drug name, target name and gene name. The total number of drug is 6071. For each drug, it initiates an association relationship with 1 or many targets. We sort drug by the descending order of amount of related targets and genes. Table 32 gives top 5 drug in terms of their corresponding number of targets and gene. As Table 32 shows, 1 drug has m targets, and m targets relates to n genes where m>n>0.

Step3: Fig. 79 gives the interface of the SLAP service. Each drug listed in Table 32 is given to the SLAP as compound input. By running the SLAP framework, target nodes and paths of each drug will be given as well. We then compare the query results from the SLAP and the GraphKDD.



	druglabel	targetlabel	gene
1	"Pyridoxal Phosphate [drugbank:DB00114]"@en	"Histidine decarboxylase [drugbank:BE0000002]"@en	http://bio2rdf.org/genecards:HDC
2	"L-Histidine [drugbank:DB00117]"@en	"Histidine decarboxylase [drugbank:BE0000002]"@en	http://bio2rdf.org/genecards:HDC
3	"L-Arginine [drugbank:DB00125]"@en	"Nitric oxide synthase, inducible [drugbank:BE0000005]"@en	http://bio2rdf.org/genecards:NOS2A
4	"L-Citrulline [drugbank:DB00155]"@en	"Nitric oxide synthase, inducible [drugbank:BE0000005]"@en	http://bio2rdf.org/genecards:NOS2A
5	"Pyridoxal Phosphate [drugbank:DB00114]"@en	"Glycogen phosphorylase, liver form [drugbank:BE0000007]"@en	http://bio2rdf.org/genecards:PYGL
6	"Adenosine monophosphate [drugbank:DB00131]"@en	"Glycogen phosphorylase, liver form [drugbank:BE0000007]"@en	http://bio2rdf.org/genecards:PYGL
7	"Tetrahydrofolic acid [drugbank:DB00116]"@en	"Aminomethyltransferase, mitochondrial [drugbank:BE0000010]"@en	http://bio2rdf.org/genecards:AMT
8	"NADH [drugbank:DB00157]"@en	"Aminomethyltransferase, mitochondrial [drugbank:BE0000010]"@en	http://bio2rdf.org/genecards:AMT
9	"Succinic acid [drugbank:DB00139]"@en	"Solute carrier family 13 member 1 [drugbank:BE0000022]"@en	http://bio2rdf.org/genecards:SLC13A1
10	"L-Cystine [drugbank:DB00138]"@en	"Cystine/glutamate transporter [drugbank:BE0000030]"@en	http://bio2rdf.org/genecards:SLC7A11
11	"L-Glutamic Acid [drugbank:DB00142]"@en	"Cystine/glutamate transporter [drugbank:BE0000030]"@en	http://bio2rdf.org/genecards:SLC7A11
12	"L-Cystine [drugbank:DB00138]"@en	"Cystinosin [drugbank:BE0000035]"@en	http://bio2rdf.org/genecards:CTNS
13	"Pyruvic acid [drugbank:DB00119]"@en	"Pyruvate kinase PKLR [drugbank:BE0000046]"@en	http://bio2rdf.org/genecards:PKLR

Figure 78: Drug Target Association Query Results.


SLAP
For Drug Target Prediction
Compound
(CID, SMILES, or Drug Name)
structure
(Example: 5880, CC12CCC(CC1CCC3C2CCC4(C3CCC4=O)C)O, or Aetiocholanolone)
Protein
(Gene Symbol, Protein Name, or UniportID)
sequence
(Example: NR1I2, Pregnane X receptor or O75469)
SLAP
Advanced
example1; example 2; example 3; example 4; example 5

- input compound and target to get their association
- input compound alone to get its targets and its biologically similar drugs (take ~1min)
- input protein alone to get its ligands
- click 'advanced' to upload your drug target pairs

Help API Download Acknowledgement Feedback

Recommend: run SLAP in Firefox or Chrome

if you are not happy with the result or your compound is a chemical rather than a drug, do let us know, our beta version may be of help.

Cite: Chen B, Ding Y, Wild DJ (2012) Assessing Drug Target Association Using Semantic Linked Data. PLoS Comput Biol 8(7): e1002574. doi:10.1371/journal.pcbi.1002574

Figure 79: The SLAP Interface

Table 32: Top 5 Drug Instances

Drug Name	**Unique # of Target**	**Unique # of Gene**
NADH [drugbank:DB00157]	143	141
Beta-D-Glucose [drugbank:DB02379]	90	11
Flavin adenine dinucleotide [drugbank:DB03147]	80	15
Pyridoxal Phosphate [drugbank:DB00114]	66	54
Citric Acid [drugbank:DB04272]	64	12

Table 33 gives a comparison between the GraphKDD and the SLAP framework in terms of the number of gene detected for top 5 drug. For $NADH$, the GraphKDD is able to find 141 genes, which include $BLVRB$, $HSD17B2$, $ADH5$, $EHHADH$ and so on. However, SLAP is not able to find any gene for $NADH$. For $Beta - D - Glucose$, the GraphKDD is able to find 11 genes, which includes $IFNB1$, $LGALS7$, GCK, $SFTPD$, $GNPDA1$, $AMY1A$, $HK1$, $LCTL$ and so on. However, SLAP is still not able to find gene for $Beta - D - Glucose$. For $Flavinadeninedinucleotide$, the GraphKDD detects 15 different genes, which includes $CYB5R3$, $NQO2$, $MAOB$, DAO, $ACOX1$ and so on. While the SLAP gives 3 possible genes $HMOX2$, $ACOX1$ and $HMOX1$. For $PyridoxalPhosphate$, the GraphKDD is able to find 54 genes (e.g., DDC, $GIG18$, $PYGL$) while the SLAP is capable of catching 56 genes (e.g., $GOT1$, $PDXK$, $GAD1$). For $CitricAcid$, the GraphKDD can find 12 different genes (e.g., $GNMT$, $BHMT$, $AKR1B1$), but the SLAP cannot find any genes.

Because SLAP focuses on predicting link for path patterns between $ChemicalCompounds$ and $Gene$ with specific predicates including bind, $hasGo$, $hasSubstructure$, $hasPathway$, $hasTissue$ and PPI. However, the GraphKDD focuses on finding approved existing link for path pattern between $Drug$ and $Target$ through $target$ predicate, we not only focus on $ChemicalCompound$ but also pay attention to a more general $Drug$ information. Therefore, for $NADH$, $Beta - D - Glucose$ and $CitricAcid$, they have all approved existing links but no predicting links, thatâs the reason why the GraphKDD is able to find more genes than the SLAP. For $Flavinadeninedinucleotide$, it has a smaller portion of predicting links than approved existing links, so the GraphKDD can find more gene paths than the SLAP. However, for $PyridoxalPhosphate$, it contains a large number of predicting links, so the SLAP can find more gene paths than the GraphKDD. As a summary, we conclude that SLAP is a good tool to predict link and get association between drug and gene. However, it overlooks some existing links between $Drug$ and $Gene$.

In addition, for single domain knowledge discovery, as Table 32 shows, GraphKDD is able to retrieve drug-target-gene path. Compared to SLAP drug-gene path, GraphKDD gets more information. For example, with GraphKDD, drug $NADH$ has 143 unique $Targets$ and 141 unique $Genes$ for single domain drugbank knowledge discovery.

Moreover, compared to the SLAP, the GraphKDD can provide topic awareness features. For example, for $NADH$, $Beta - D - Glucose$, $Flavinadeninedinucleotide$, $PyridoxalPhosphate$ and $CitricAcid$, they across two topics (Topic 16 and 27) with 8 domains (DrugBank, ClinicalTrials, KEGG, SIDER, OMIM, CTD, HGNC and CTD).

Table 33: Comparison Results Between GraphKDD and SLAP

Drug Name	**GraphKDD** **# of Genes**	**SLAP** **# of Genes**
NADH [drugbank:DB00157]	141	0
Beta-D-Glucose [drugbank:DB02379]	11	0
Flavin adenine dinucleotide [drugbank:DB03147]	15	3
Pyridoxal Phosphate [drugbank:DB00114]	54	56
Citric Acid [drugbank:DB04272]	12	0

4.8 Summary

We have formally introduced important concepts and essential algorithms involved in query generation and topic aware link discovery. Based on this theory, we have conducted evaluations on both applications.

Specifically, for query generation, we have used DrugBank data as an use case to show query generation results coming from top 4 ranking topics. For the topic aware link discovery tool, we applied a performance evaluation with different single domain and cross domain datasets to demonstrate the efficiency and topic awareness feature of it.

CHAPTER 5

THE GRAPHKDD ONTOLOGY LEARNING FRAMEWORK

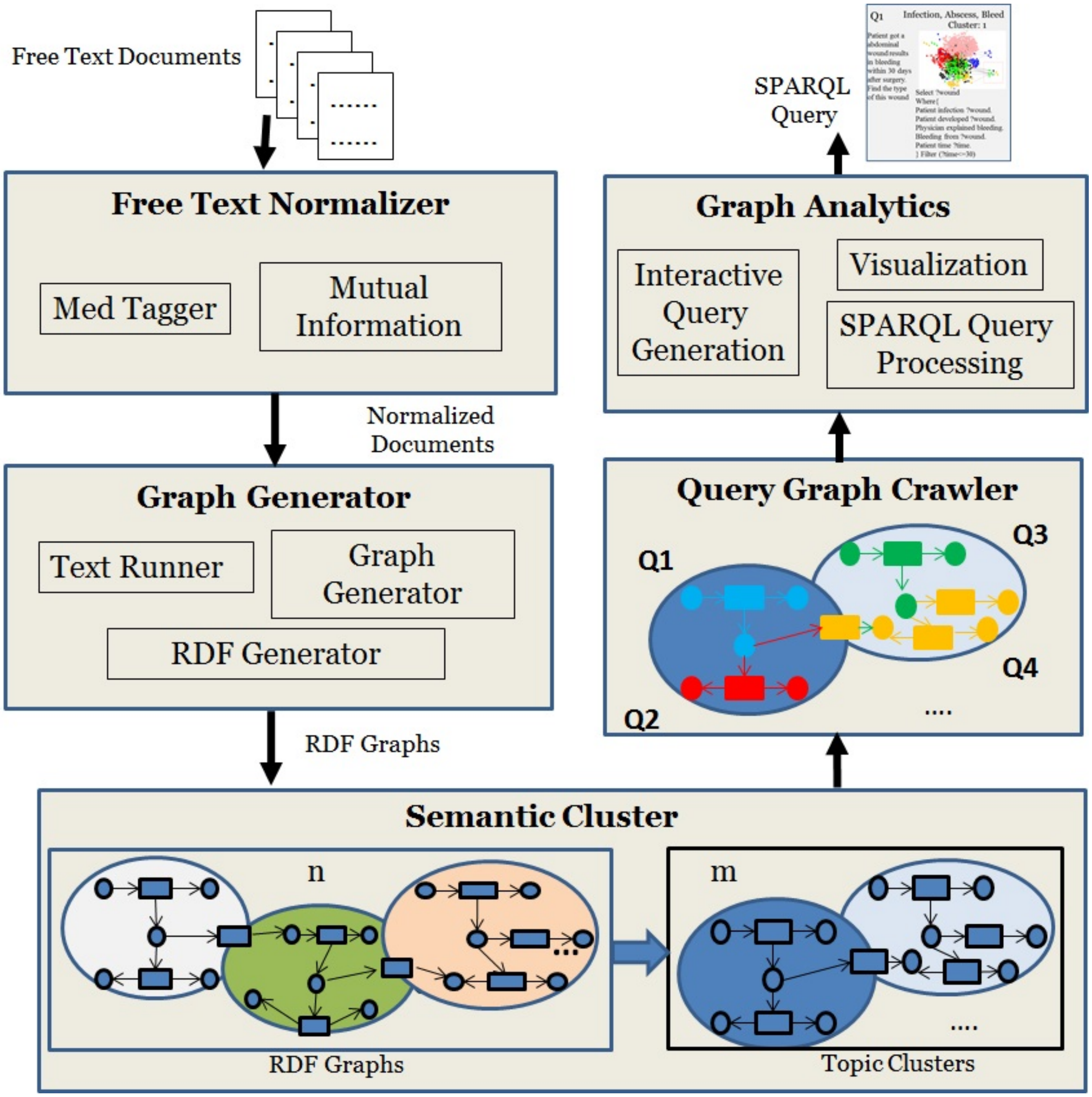
5.1 Introduction

In this chapter, we demonstrate ontology learning ability of the GraphKDD. Based on the GraphKDD framework, we have extended it to extract useful information from unstructured data (e.g., text format) and convert them into RDF/OWL to be accepted. We have implemented a prototype system [108] [109] and evaluated the proposed model in biomedical informatics domain with colorectal surgical cohort from the Mayo Clinic. The framework is shown in Fig. 80.

We first introduce related work on ontology learning, then introduce basic component included in the GraphKDD ontology learning framework. Evaluation and results are also given to demonstrate the function of the GraphKDD.

5.2 Related Work

The motivation of this research is to perform dynamic learning and knowledge discovery from ontology. As mentioned in paper [133], open issues existing in ontology learning include but not limit to learning specific relation, evaluation benchmark, incremental ontology learning etc. GraphKDD framework is able to handle these problems. First of all, GraphKDD is capable of breaking large ontology into smaller topics. Each



Patient got a abdominal wound results in bleeding within 30 days after surgery. Find the type of this wound

```
Select ?wound
Where{
Patient infection ?wound.
Patient developed ?wound.
Physician explained bleeding.
Bleeding from ?wound.
Patient time ?time.
} Filter (?time<=30)
```

Figure 80: The GraphKDD Framework

topic contains nodes with close relationship and represents a specific group of information. In addition, from each topic, GraphKDD is able to build benchmark evaluation query in a systematical way. What is more, GraphKDD also provides incremental ontology building and integration from unstructured data (e.g., text format data).

There exist many different direction of research on ontology learning. Zhou and Lina [133] points out three dominant categories of ontology learning algorithms.
The first one is statistics based approach. Mutual information is widely used to extract cooccurrence relations [24] [73] [53] [27]. Pereira et al., [88] and Glover et al., [42] proposed a clustering based approach to group various of words. Li and Abe [72] provided a Minimum Description Length (MDL) principle based word clustering solution. Sanderson and Croft [99] and Doan et al., [32] proposed conceptual mapping based approach to build ontology from text. Tari et al., [117] indicated a reverse engineering methodology to build ontology schema from relational database.

The GraphKDD also applies mutual information to extract key words from data repository and build ontology. Above statistics approaches mainly focus on building the ontology but lacking of find specific relations. However, in addition to these approaches, GraphKDD works directly on graph to break the large ontology down into smaller pieces, which represent specific groups of information. What is more, by using our top-down Hierarchical Predicate oriented K-Means clustering (HPKM) and bottom-up Predicate oriented Hierarchical Agglomerative clustering (PHAL) approaches, context aware topics are able to be made.

The second essential category of ontology learning algorithm is rule-based approach. Some researchers designed heuristic patterns to construct rules for semantic lexicon [18] [51] [52] [95] [6] . Ruge and Gerda [97] built rules based on term dependency and association. Many researchers combined both syntactic and semantic ways to generate rules [126] [46] [47] [92]. Califf et al., [16] and Riloff et al., [94] used pattern and dictionary based approach respectively to do information extraction from documents. Jannink, Jan and Wiederhold [59] developed a ArcRank based algorithm in directed graph to get relationship among nodes. Schlobach and Stefan [100] merged knowledge discovery method into knowledge representation system to build criteria for calculating concepts. Missikoff et al., [81] and Ciravegna et al., [26] conducted information extraction from corpus and used such information to annotate ontology.

Compared with static rule-based approaches, the GraphKDD proposes a pure unsupervised learning approach with dynamic knowledge discovery and ontology learning features. We analysis ontology based on our own defined neighborhood relationship and quantify the relationship with neighborhood association measurement.

The last popular research direction of ontology learning is hybrid approach. Roark and Eugene [96] combined the advantage of clustering and heuristic pattern. Similarly, [37], [5], [38], [118] and [25] combined clustering approach and natural language processing techniques (e.g, phrase chunker, syntax regulation) to acquire knowledge from ontology. Kietz et al., [65] and Maedche and Staab [78] mined and built ontology from text by parsing cohort and analyzing association rules. Xu et al., [129] used GermaNet [48] to analysis lexico-syntactic pattern from documents and applied clustering technique to extract information. Some other researchers also used supervised classification approaches to achieve information retrieval goal [56] [98].

Although both supervised and unsupervised learning are widely used in discovering ontological concepts and relations, the majority of techniques that been used in ontology learning and knowledge discovery are still unsupervised learning. As Zhou and Lina [133] pointed out, the reason is that usually training data that used to annotate ontology are not available. That provides one support reason that we choose unsupervised learning approaches to do knowledge discovery from ontology. In this research, we also apply hybrid approach. Specifically, we use both top-down unsupervised approach Hierarchical Predicate oriented K-Means clustering (HPKM) and bottom-up unsupervised approach Predicate oriented Hierarchical Agglomerative clustering in this research. But different from the above hybrid methodologies, we used different solutions for various purposes. After evaluation, we found that top-down solution focuses on global optimization which is more suitable for single domain dataset knowledge discovery and bottom-up solution is good at finding local optimization and is a better option for cross domains dataset knowledge discovery.

5.3 Methods

Researchers and health care practitioners prefer to conduct research in an evidence-based practice by using available research results when making decisions in health care. The main challenge we are facing to support evidence-based research is the big data problem along with large, complex, and dynamic medical data (e.g., clinical research data, EHRs, ontologies). A lot of medical ontologies and tools have been developed for biomedical research and applications. However, these are not sufficient for integrating or mapping unstructured data to structured data that will be significant for evidence-based research. It is mainly due to the lack of the ability to integrate data from such a variety of sources and extract both a cohesive structure and semantics from structured or unstructured data to support evidence-based research. Under this drive, we extend the GraphKDD framework with ontology learning feature that supports the ability to learn from text based unstructured doctor notes and patient records to build semi-structured ontology for semantic analysis and knowledge discovery.

The extended framework is based on the following steps: 1) extracting key words from report; 2) converting unstructured (free text) data to semantically structured (RDF/OWL) data; 3) arranging them into groups in a semantically meaningful manner; 4) generating queries for evidence-based practice; and 5) providing visualization based query analytics tool.

As seen in this Fig. 80, the Free Text Normalizer component first reads unstructured free text documents from doctors' notes and patients' records then uses MedTagger [119] to filter unnecessary terms out and convert free text terms to normalized ones. The Graph Generator component then applies TextRunner [131] on each unstructured normalized document and simplifies the term to generate a RDF triplet.

We first explain how to convert unstructured data to a linked data structure (RDF). A slight different from the GraphKDD is that instead of using Hierarchical Predicate

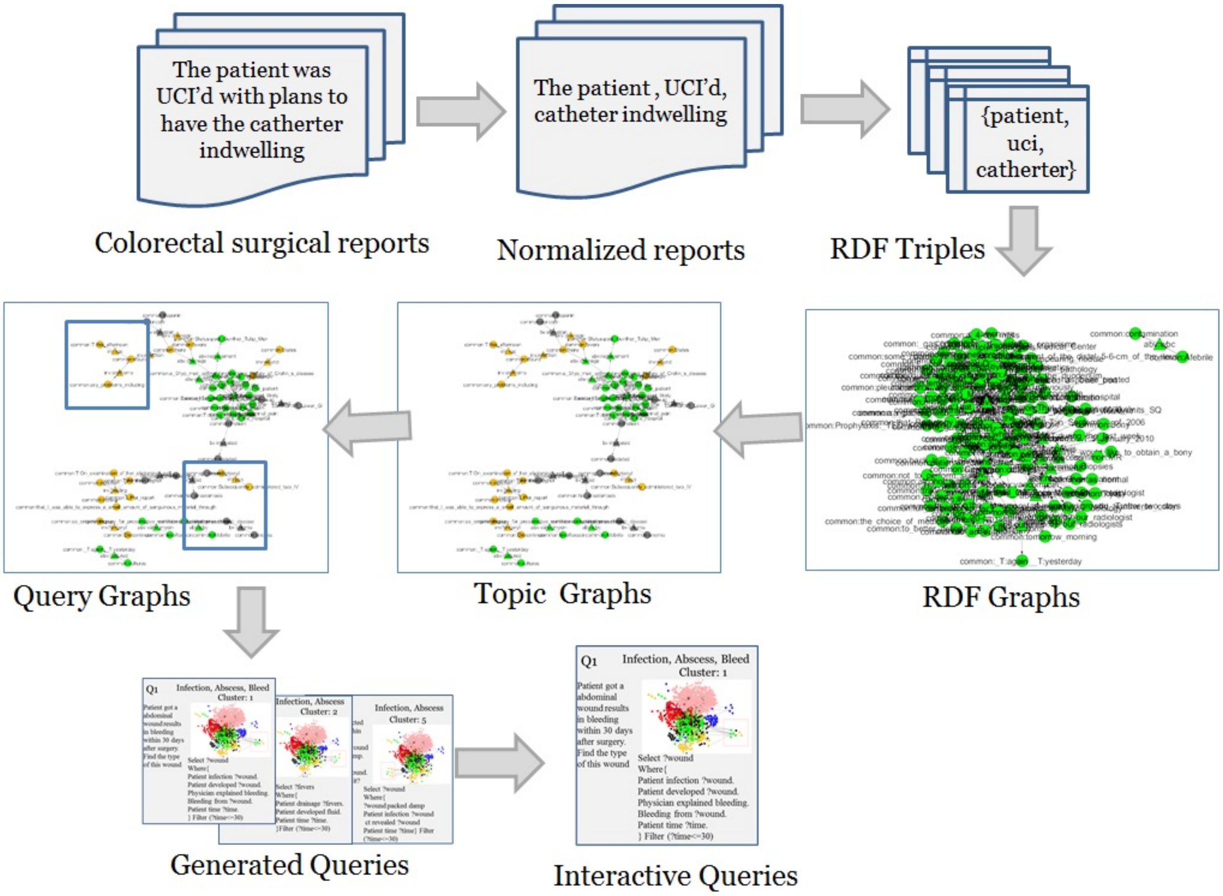

The patient was UCI'd with plans to have the catherter indwelling
Colorectal surgical reports
The patient , UCI'd, catheter indwelling
Normalized reports
{patient, uci, catherter}
RDF Triples
RDF Graphs
Topic Graphs
Query Graphs
Q1
Infection, Abscess, Bleed
Cluster: 1
Infection, Abscess
Cluster: 2
Infection, Abscess
Cluster: 5
Generated Queries
Q1
Infection, Abscess, Bleed
Cluster: 1
Patient got a abdominal wound results in bleeding within 30 days after surgery. Find the type of this wound
Select ?wound
Where{
Patient infection ?wound.
Patient developed ?wound.
Physician explained bleeding.
Bleeding from ?wound.
Patient time ?time.
} Filter (?time<=30)
Interactive Queries

Figure 81: The GraphKDD Data Flow

oriented K-Means (HPKM) clustering, in this ontology learning specific framework, we apply a Hierarchical Predicate oriented Fuzzy C-means (HFCM) clustering algorithm on data. The basic between HPKM and HFCM are same, only different is HFCM gives soft clustering outputs while HPKM can only give hard clustering results. The reason we use a top-down based HFCM algorithm in this specific study is because we want to find cross domains correlation while maintain the global view of the datasets. The data flow of the GraphKDD framework is shown in Fig. 81. The detailed phases are described as follow.

***P*hase1:Feature Selection and Concept Annotation**.

We extract various free text reports and then perform preprocessing to maintain terms consistently and exclude irrelevant terms, using filtering. Inequality of the likelihood holds between two different values to describe their correlations. To get the inequality of the likelihood between any pairs of words, we then extract co-occurring terms from free text clinical notes, and calculated the inequality score to cluster them into a different category (domain). In this study, we use MedTagger [131], which is an open source concept detection and normalization tool through open health natural language processing. Specifically, this tool identifies phrases present in MedLex, a general semantic lexicon created for the clinical domain [67]. The point-wise mutual information is used to assess the inequality of the likelihood for given terms [74] as Eq. (5.1) shows.

$$Inequality(c, o) = \log_2(N(c, o) * \log_2 \frac{N(c, o) + 0.01}{N(o)} - \log_2 \frac{N(c)}{N} \tag{5.1}$$

N is the number of observations (e.g., the number of cases for all patients), N (c) is the number of cases having the concept c, N(c,o) is the concept c and the number of cases with a specific complication o, and N(o) is the number of cases with a specific complication o.

As the example shown in Fig. 81, a given input text, ”*the patient was UCI'd with plans to have the catherter indwelling*”, MedTagger recognized the uci and $catheter$ concepts and we also find these terms come from $ILEUS$ report. Then we use the point-wise mutual information measurement to calculate any inequality likelihood between these concepts (uci and $catheter$) and $ILEUS$. For example, the total number

Table 34: Clinical Free Text

	Clinical Free Text
1	Patientâs abdominal wound was exacerbated by dressing changes
2	Any problems including increased erythema around the wound
3	The residual,urine levels drop below certain level
4	There is substantial,further elevation in patientâs troponins
5	More,hypotension requiring initiation of pressor, to achieve satisfactory blood,pressure

of cases is 1980, number of $ILEUS$ cases is 400, the number of concept uci among all cases is 600, and the number of concept uci along with $ILEUS$ cases is 500. Based on equation, the inequality between uci and $ILEUS$ is 1.69. We do the same for all other concepts and ranked these concepts by their inequality score and only chose the top 60 of them. Then for any free text which contains top 60 concepts with inequality score, we use OpenIE [35] to get the triple (S, P, O) from the free text annotated with the concepts recognized by MedTagger. For this purpose, we first get a triple $\{thepatient, UCI'd, catherterindwelling\}$. To make the triple normalized, we looked up the MedTagger concept in the dictionary again and converted the triple to $\{patient, uci, catherter\}$. For each complication case, we do the same work above in order to generate ontologies, respectively.

In Table 34, we list some examples of clinical free text. Correspondingly, we also list how to map among clinical free text, MedTagger normalized terms and triplets in Table 35.

***P*hase2: RDF Graph Construction**.

First, the top K concepts are selected and each sentence with these concepts in the

Table 35: Mapping among Clinical Free Text, MedTagger Terms and RDF Triples

	MedTagger Terms	**RDF Triples**
1	Abdominal wound exacerbated by dressing change	{abdominal_wound, exacerbated_by, dressing_change}
2	Problems,including erythema around wound	{problems,,erythema, wound}
3	Urine drop,below level	{urine, drop,,below_level}
4	Place has,further elevation in troponins	{place,,elevation, troponins}
5	Hypotension requiring,blood	{hypotension,,requiring, blood}

datasets is annotated with the selected terms. We then extract the assertions (RDF/OWL triples) from a given free test dataset considering the top K concepts of each domain and generated RDF/OWL triples, respectively. We use OpenIE to extract the triples from the free text. The OpenIE that is based on TextRunner, ReVerb [36] using (PoS) patterns, extracts a significant relation without any relation-specific input. OpenIE uses a conditional random field (CRF) classifier to automatically extract triples representing binary relations (Arg1, Relation, Arg2) from sentences. The triples generated from OpenIE are connected to generate RDF graphs.

***P*hase3: Assertion Clustering and Query Generation**.

Our assumption for the Predicate Oriented Neighborhood Patterns (PONP) is that a predicate plays an important role in sharing information and connecting entities among heterogeneous data. For any given domain, the number of unique relations is much less than the number of concepts. Thus, this is another scalable approach for mapping domains than the concept-centric approach. A group of terms (subjects) can be connected to a group of terms (objects) through a single predicate unlike the concept-driven approach. From the unit of ⟨subjects-predicate-objects⟩, a specific context can be discovered from the associated concepts (subjects, objects). From the neighbors of the predicates, a specific context can be discovered from the association of predicates and their subjects and objects. We can infer/predict missing predicates or missing concepts based on existing contexts. Therefore, we generate a hypothesis that when a graph can be clustered based on PONP patterns, data in the same clustered group have a closer relationship than when in different ones. Predicate neighboring patterns are important to link data together with a variety of domains.

After executing the above three steps, we will use the GraphKDD framework to handle input as RDF data to analysis the predicate pattern and conduct unsupervised learning approach as mentioned in previous chapters.

5.4 Evaluation and Results

In this section, we use Mayo Clinic colorecal cohort as a case study to demonstrate the GraphKDD ontology learning.

5.4.1 Case Study

As a case study, the Mayo Clinic's colorectal surgical reports are used to generate queries by categorizing relationships among six colorectal post-surgical complications (deep vein thrombosis/pulmonary embolism, bleeding, wound infections, myocardial infraction, ileus and abscess/leak). Post-surgical complications are related to general or certain type of surgeries. Clinical data of six complications after colorectal surgery are attempted, analyzed to find interesting patterns/associations in a single or multiple complications, and generated comprehensive cross domains queries that might be useful in conducting evidence-based practice by using available research results. We assume six post-surgical complications represent six domains. Predicate profiles and association patterns are important to link data together with a variety of domains.

5.4.2 Convert Colorectal Surgical Cohort to Ontology

Our case study has 1,980 colorectal surgical cases for 1,416 patients between 2005 and 2013 enrolled at the Mayo Clinic in Rochester, MN. We use our previous work, MedTagger to extract concepts from clinical notes written about any complications within 30 days after surgery in cohort. The top 60 concepts (by their inequality scores) are used for information extraction. The definition of the 6 complications is shown in Table 36.

We build six ontologies based on the top 60 terms ranked by their inequality scores. Fig. 82 gives visualizations for each of the six ontologies. We use different colors to indicate different domains, so that $ABSCESS$ is in green, $BLEED$ is in gray, $DVTPE$ is in blue, $ILEUS$ is in pink, MI is in red and $INFECTION$ is in orange. Table 37 shows the statistics of each ontology. Among the six ontologies, there are a total of 445 subjects, 83 predicates, 482 objects and 1210 triples involved. We then integrate six ontologies together to make them interlinked and prepared to apply a Hierarchical Fuzzy C-Means clustering algorithm on it.

5.4.3 Hierarchical Fuzzy C-Means (HFCM) Clustering Approach

We apply Hierarchical Fuzzy C-Means (HFCM) clustering for integrated ontology on an input predicate similarity matrix with size 83*83. As a result, we get eight different topics. The hierarchical clustering graph is shown in Fig. 83. The original integrated

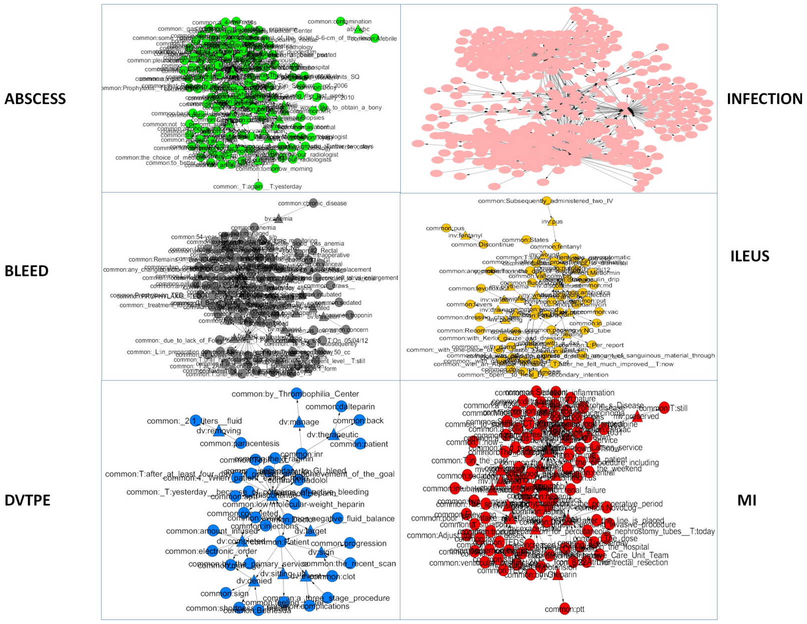


Figure 82: Visualization of 6 Complication Ontologies

ontology is partitioned into three intermediate sub-topics based on the optimal Silhouette Width. In addition, three intermediate sub-topics can be further split into eight smaller topics with the best Silhouette Width. Because eight topics cannot be further clustered, then the GraphKDD framework stops the HFCM algorithm and produced eight topics as the final output. We collect the top five predicates for each topic by their total in-degrees and out-degrees and summarized each topic with those predicates. What is more, out of the top five ranked predicates, we also select top two unique predicates with the most in-degrees and out-degrees for each topic. Unique predicates indicate those predicates that appear in only one topic. Therefore, some topics have unique predicates but some do not. Based on the top predicates and unique predicates, we generate a signature for each topic to summarize the content of each topic.

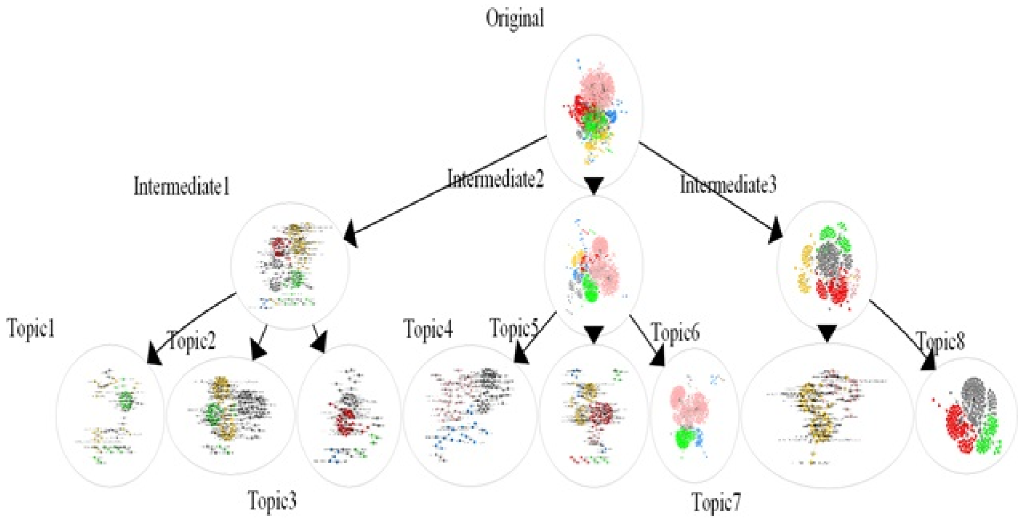


Figure 83: Visualization of Hierarchical Fuzzy C-Means Clustering

Fig. 84, 85, 86 and 87 show Topics 1-4. Topic 1 includes 3 complications (ABSCESS, BLEED and INFECTION) with 24 predicates in total. The top five predicates for Topic 1 are $abv : developed$, $inv : healing$, $abv : drainage$, $bv : anemia$ and $bv : blood$, and the top two unique predicates for Topic 1 are $abv : abscess$ and $abv : replacement$. By analyzing the signature of Topic 1, we find that Topic 1 describes the close relationship among $drainage$, $anemia$, $blood$ and $incisionhealing$. Similar to Topic 1, Topic 2 also covers three complications ($ABSCESS$, $BLEED$ and $INFECTION$) with 16 predicates. Top 5 predicates for Topic 2 are $abv : developed$, $bv : held$, $inv : discontinued$, $inv : packed$ and $bv : drop$, and there are no unique predicates for Topic 2. From this signature, Topic 2 explains that the drop of some life indicators (e.g., hemoglobin) for a patient may be related to the complication (e.g., $abscess$) developed by such patient; the patientâs wound infection is discontinued for the reason that the infection area is packed with gauze. Topic 3 introduces four complications ($ABSCESS$, $BLEED$, $DVTPE$ and MI) with 13 predicates. Top 5 predicates for Topic 3 are $mv : held$, $bv : held$, $mv : signs$, $abv : drainage$ and $bv : blood$, and the unique predicates for Topic 3 are none. This signature illustrates $BLEED$ and MI might hold the same symptoms, which are also related to drainage and blood. Topic 4 mentions three different complications ($BLEED$, $DVTPE$ and $ILEUS$) with 30 predicates. The top five predicates for Topic 4 are $bv : held$, $bv : drop$, $ilv : clamp$, $ilv : fluid$ and $ilv : dilated$, and the top two unique predicates for Topic 4 are $dv : therapeutic$ and $ilv : bolus$. From this signature, we conclude that fluid has a potential relationship with the dilated, drop of life indicator and ng tube; therapeutic is associated with bolus.

Fig. 88, 89, 90 and 91 show Topics 5-8. Topic 5 describes 5 complications ($ABSCESS$, $DVTPE$, $INFECTION$, $ILEUS$ and MI) with 23 predicates. The top 5 predicates for Topic 5 are $abv : developed$, $inv : healing$, $abv : drainage$, $bv : anemia$ and $bv : blood$. The top 2 unique predicates for Topic 5 are $mv : normalized$ and $mv : aggressive$. The top 5 predicates for Topic 5 convey the exact same information as Topic 1 does. However, unique predicates from Topic 5 tell us that the patient's pain remained poorly-controlled even with an aggressive multimodal; meanwhile, the patientâs hypotension had normalized systolic pressure. Topic 6 indicates 4 complications ($ABSCESS$, $BLEED$, $DVTPE$ and $ILEUS$) with 29 predicates. The top 5 predicates for this topic are $ilv : ng$, $ilv : remove$, $ilv : distension$, $abv : nausea$ and $abv : ct$. The top 2 unique predicates are $ilv : experienced$ and $ilv : pulled$. This

Topic1

Domains: ABSCESS, BLEED, INFECTION

# of Predicates: 24

Top Predicates: abv:developed, inv:healing, abv:drainage, bv:anemia, bv:blood

Top Unique Predicates: abv:abscess, abv:replacement

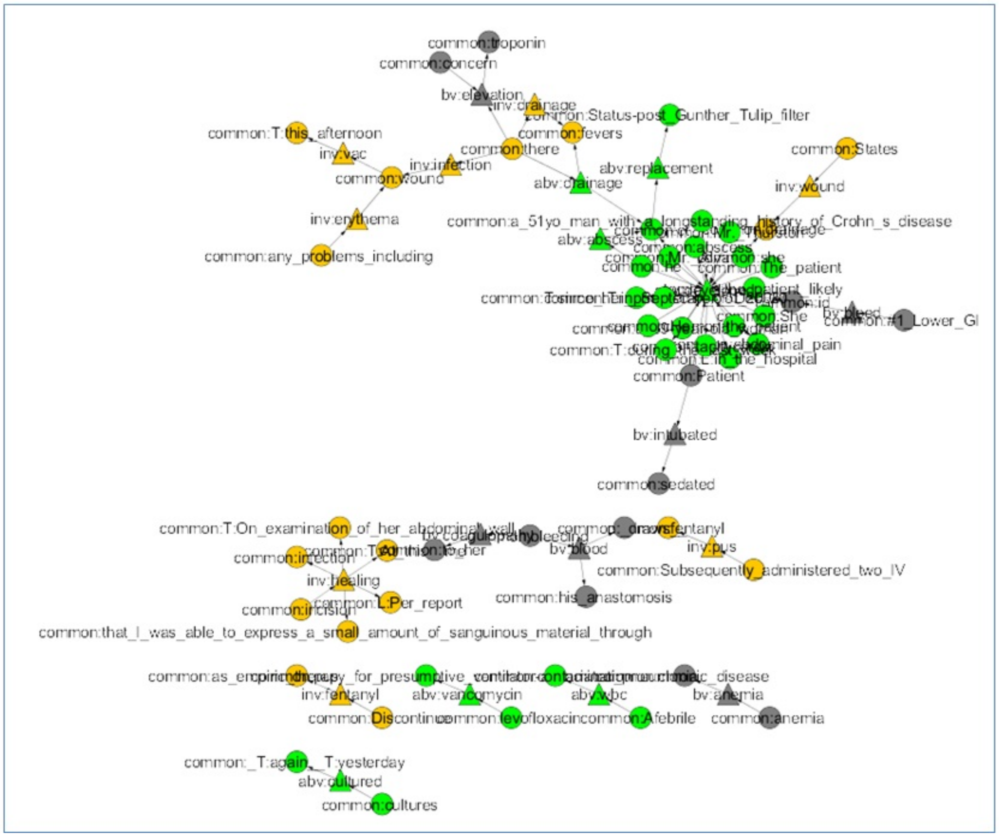


Figure 84: Detailed Information for Topic 1

Topic2

Domains: ABSCESS, BLEED, INFECTION

# of Predicates: 16

Top Predicates: abv:developed, bv:held,
inv:discontinued, inv:packed, bv:drop

Top Unique Predicates: None

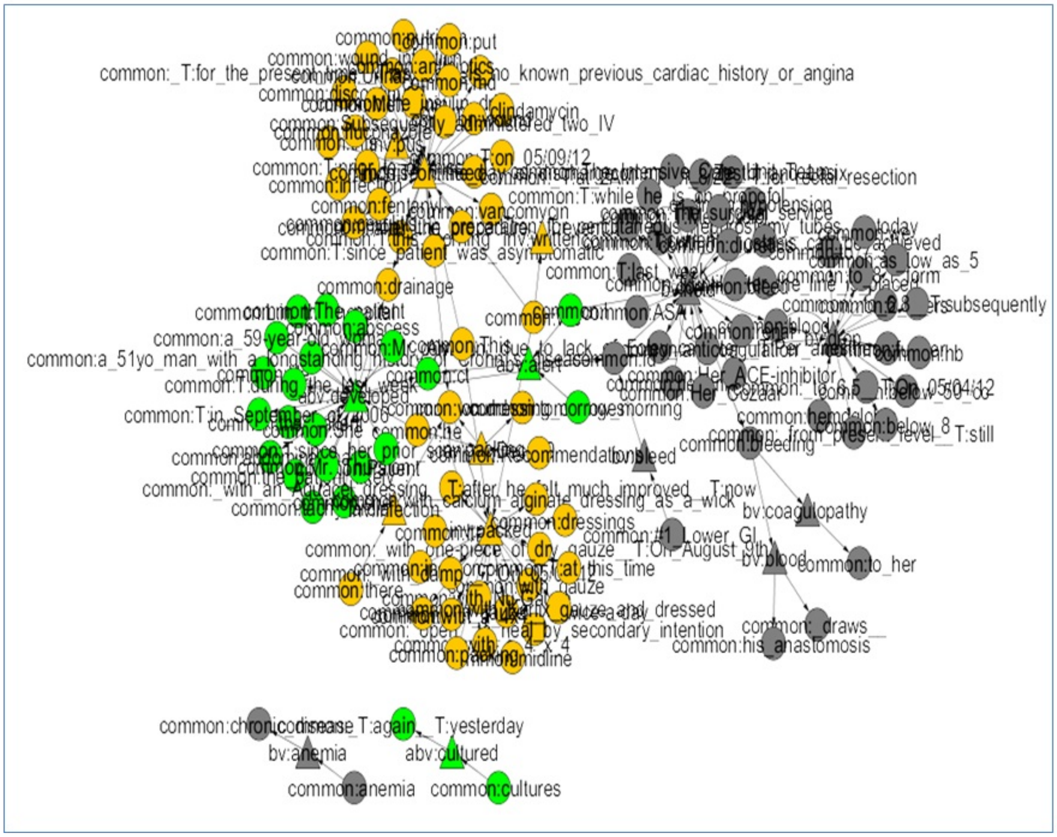


Figure 85: Detailed Information for Topic 2

Topic3

Domains: ABSCESS, BLEED, DVTPE, MI

# of Predicates: 13

Top Predicates: mv:held, bv:held,
mv:signs, abv:drainage, bv:blood

Top Unique Predicates: None

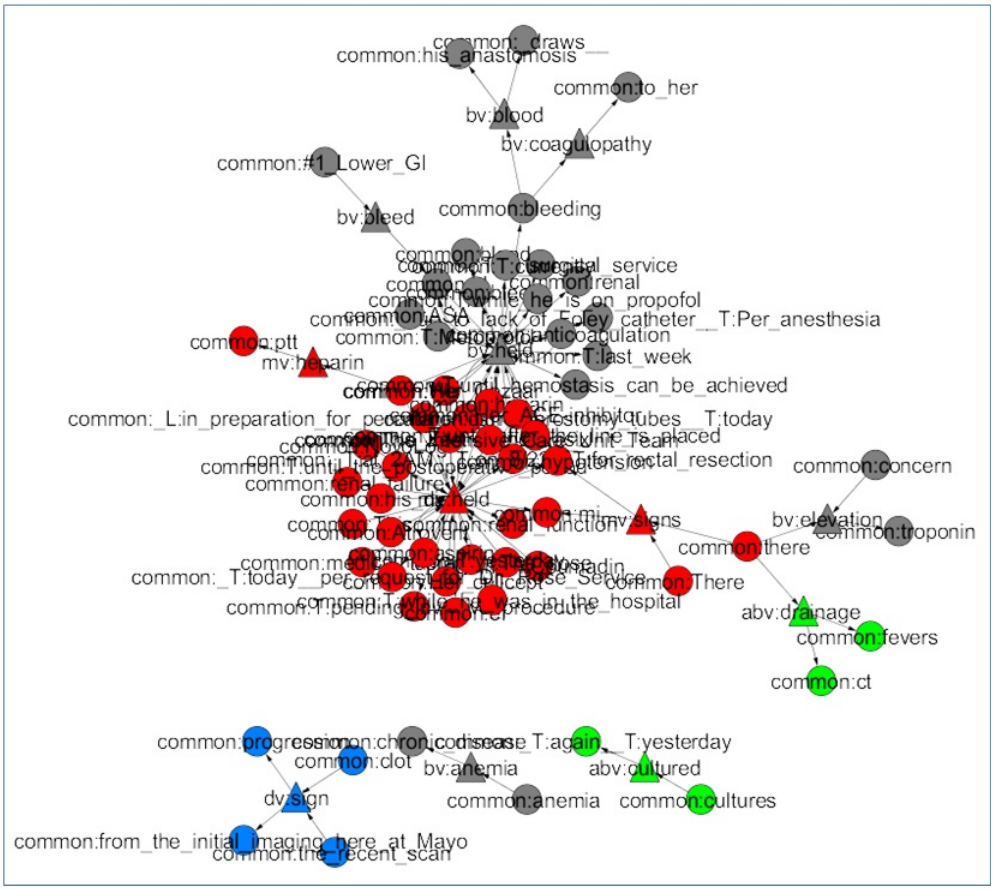


Figure 86: Detailed Information for Topic 3

Topic4

Domains: BLEED, DVTPE, ILEUS

# of Predicates: 30

Top Predicates: bv:held, bv:drop,
ilv:clamp, ilv:fluid, ilv:dilated

Top Unique Predicates: dv: therapeutic, ilv:bolus

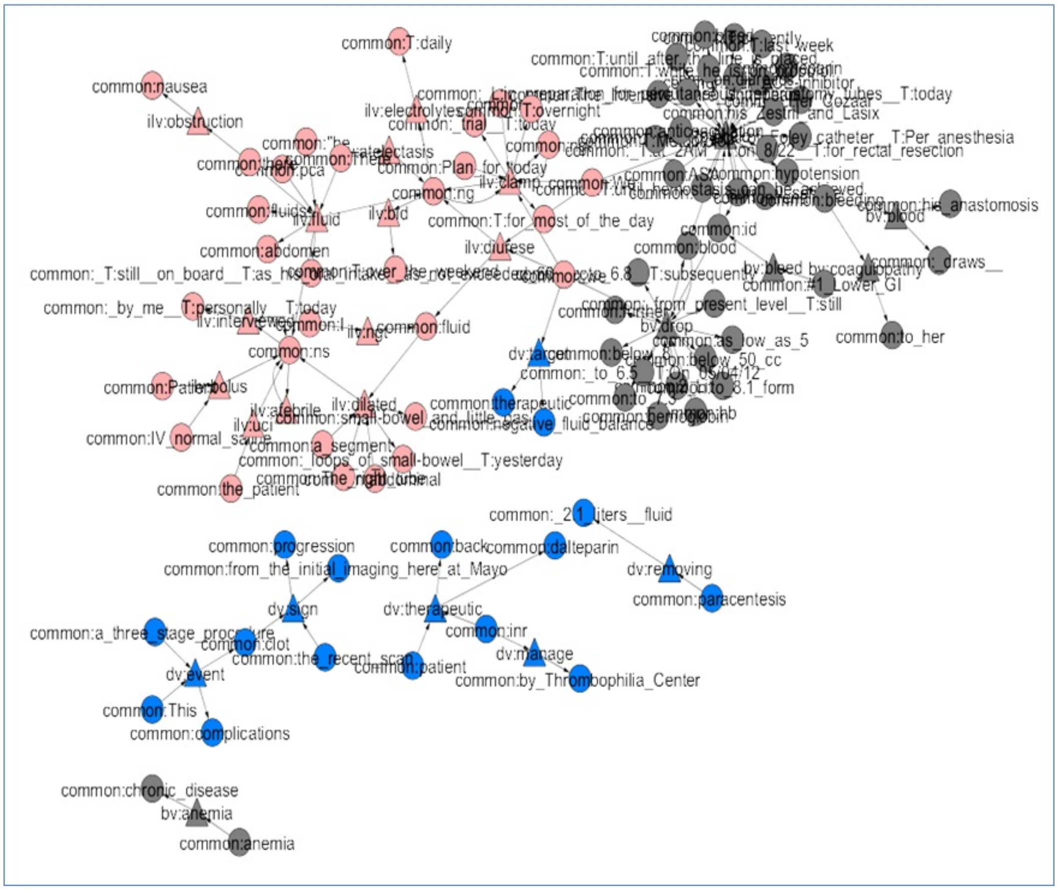


Figure 87: Detailed Information for Topic 4

signature summarizes the scenario that a patient's ng tube was pulled out, and this patient also felt nausea and distension. Topic 7 covers 2 complications ($ABSCESS$ and $ILEUS$) with 11 predicates. The top 5 predicates under Topic 7 are $inv : discontinued$, $inv : packed$, $ilv : clamp$, $ilv : fluid$ and $ilv : diurese$. The top 2 unique predicates are none. This topic is also very similar to Topic 2 and Topic 4 but with more information on diuresis. Topic 8 is related to 3 complications ($ABSCESS$, $BLEED$ and MI) with 16 predicates. The top 5 predicates involved in this topic are $bv : bleed$, $abv : pelvis$, $bv : sedated$, $abv : transferred$ and $abv : read$. The top 2 predicates are $mv : intubated$ and $bv : extubated$. This topic describes the bleeding situation of the patientâs pelvis; such patient was sedated; intubated and extubated operations are also applied on this patient.

We also conduct an experiment among different clustering algorithms to validate that HFCM is the optimal approach to do topic discovery. Silhouette width is a method of validation of consistency within clusters of data. Fig. 92, 93, 94 and 95 show validation for four partitions for each level (one first level and three second levels) with five different clustering algorithms (Hierarchical Fuzzy C-means [29], K-Means [114], Clara [63], Pam [120], and Hierarchical Clustering [62]) on the similarity matrix. Fig. 92 shows the splitting from the original ontology to intermediate clusters. Clara, Pam and Hierarchical clustering algorithms showing a relatively stable Silhouette Width for many cases and could not find an optimal cluster number. Both HFCM and K-Means give the highest Silhouette Width 0.65 when the cluster number is 3. That shows the reason why the original ontology is split in-to three intermediate clusters. Similarly, Fig. 93, 94 and 95 present

Topic5

Domains: ABSCESS, DVTPE, INFECTION, ILEUS, MI

# of Predicates: 23

Top Predicates: abv:developed, inv:healing,
abv:drainage, bv:anemia, bv:blood

Top Unique Predicates: mv:normalized, mv:aggressive

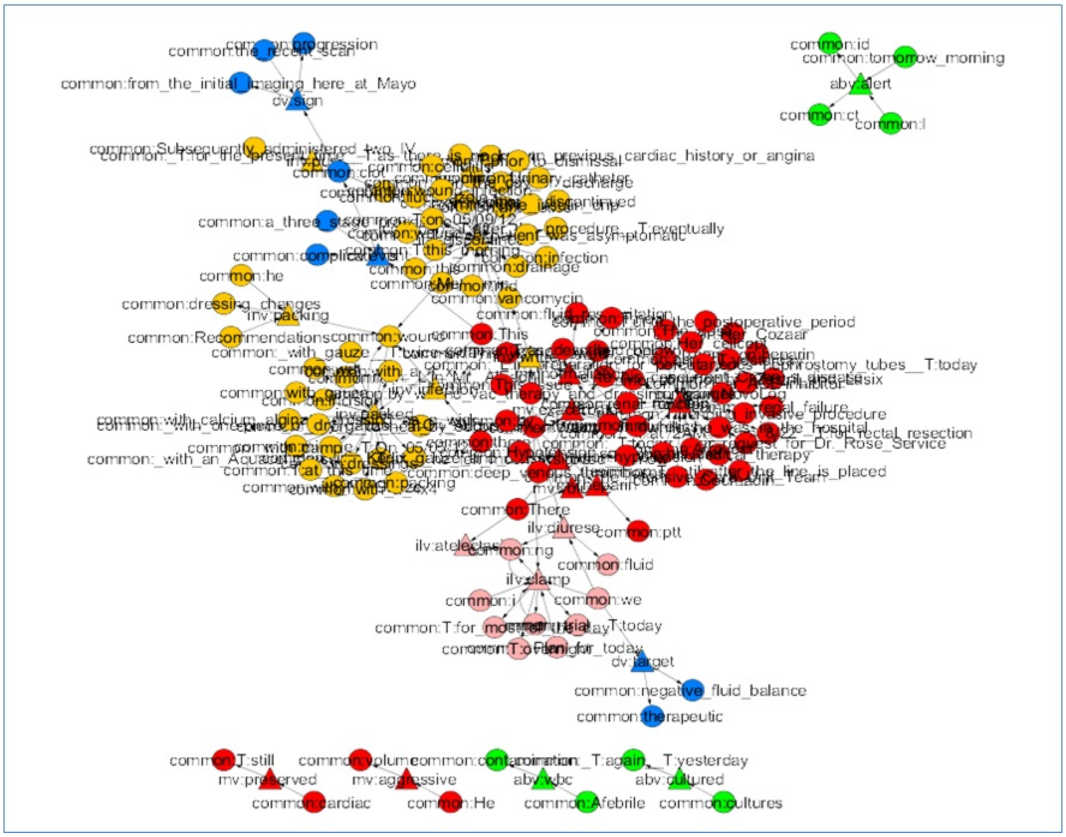


Figure 88: Detailed Information for Topic 5

Topic6

Domains: ABSCESS, BLEED, DVTPE, ILEUS

# of Predicates: 29

Top Predicates: ilv:ng, ilv:remove, ilv:distension, abv:nausea, abv:ct

Top Unique Predicates: ilv:experienced, ilv:pulled

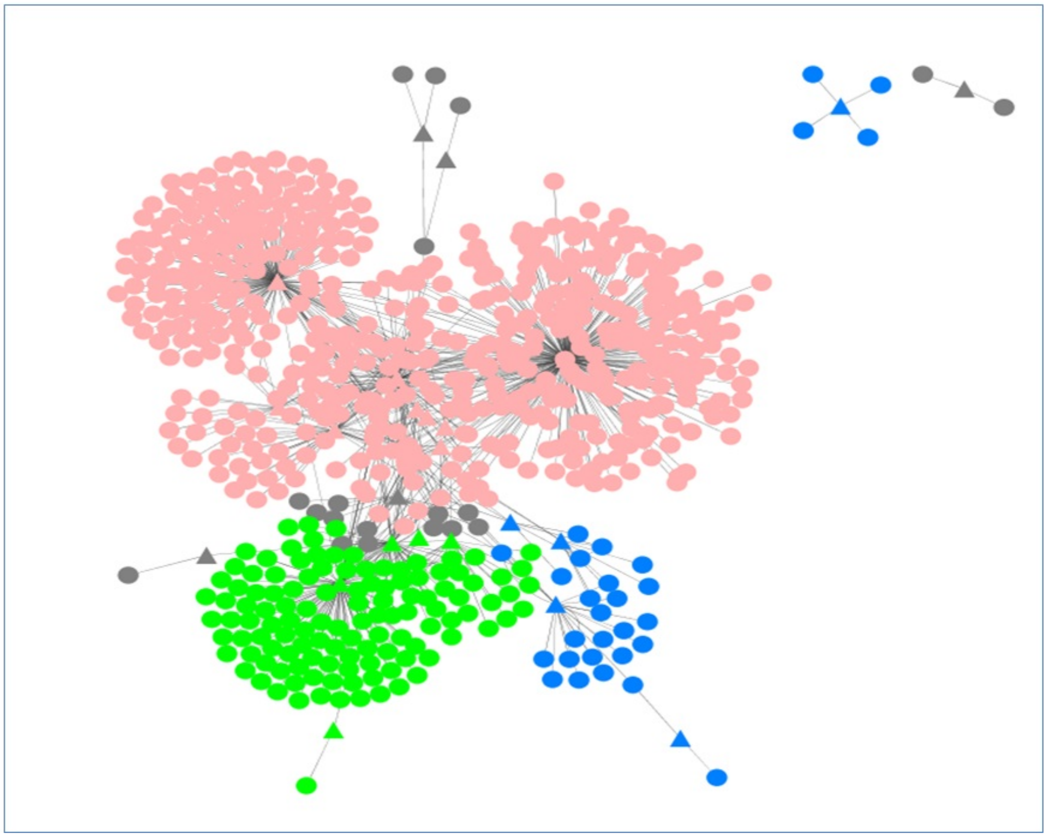


Figure 89: Detailed Information for Topic 6

Topic7

Domains: ABSCESS, ILEUS

# of Predicates: 11

Top Predicates: inv:discontinued, inv:packed,
ilv:clamp, ilv:fluid, ilv:diurese

Top Unique Predicates: None

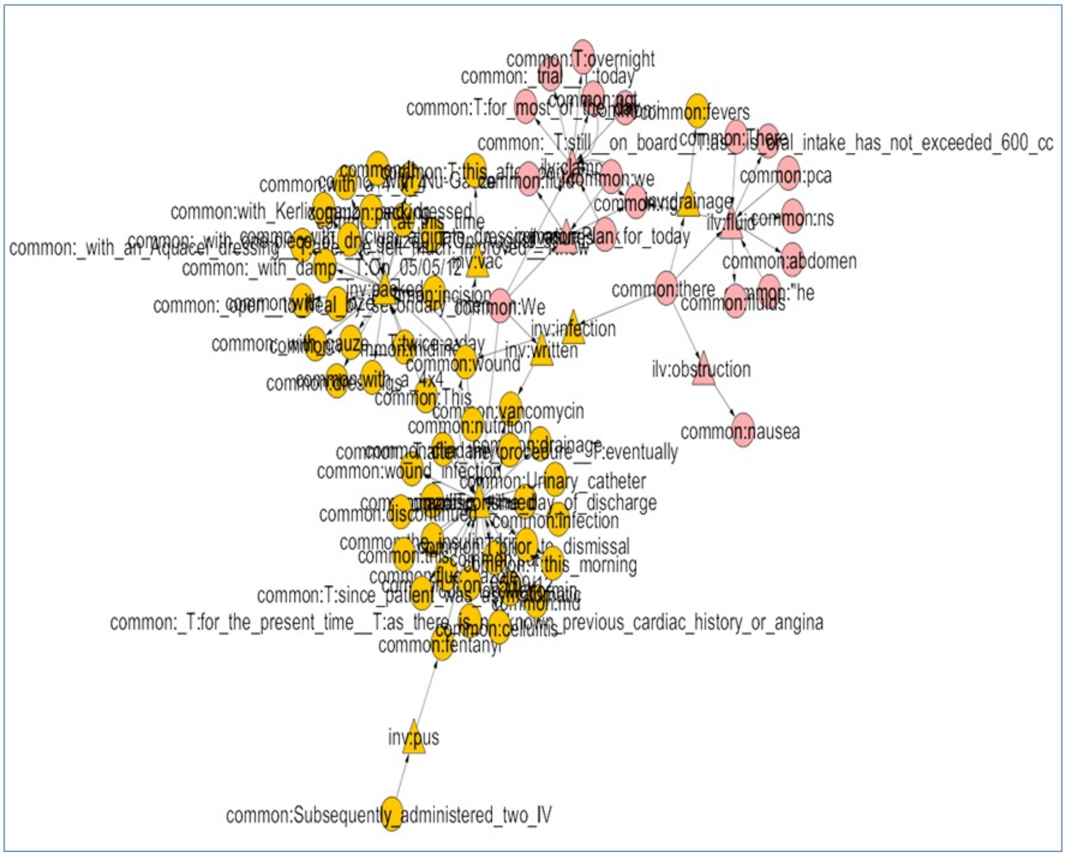


Figure 90: Detailed Information for Topic 7

Topic8

Domains: ABSCESS, BLEED, MI

# of Predicates: 16

Top Predicates: bv:bleed, abv:pelvis,
bv:sedated, abv:transferred, abv:read

Top Unique Predicates: mv:intubated, bv:extubated

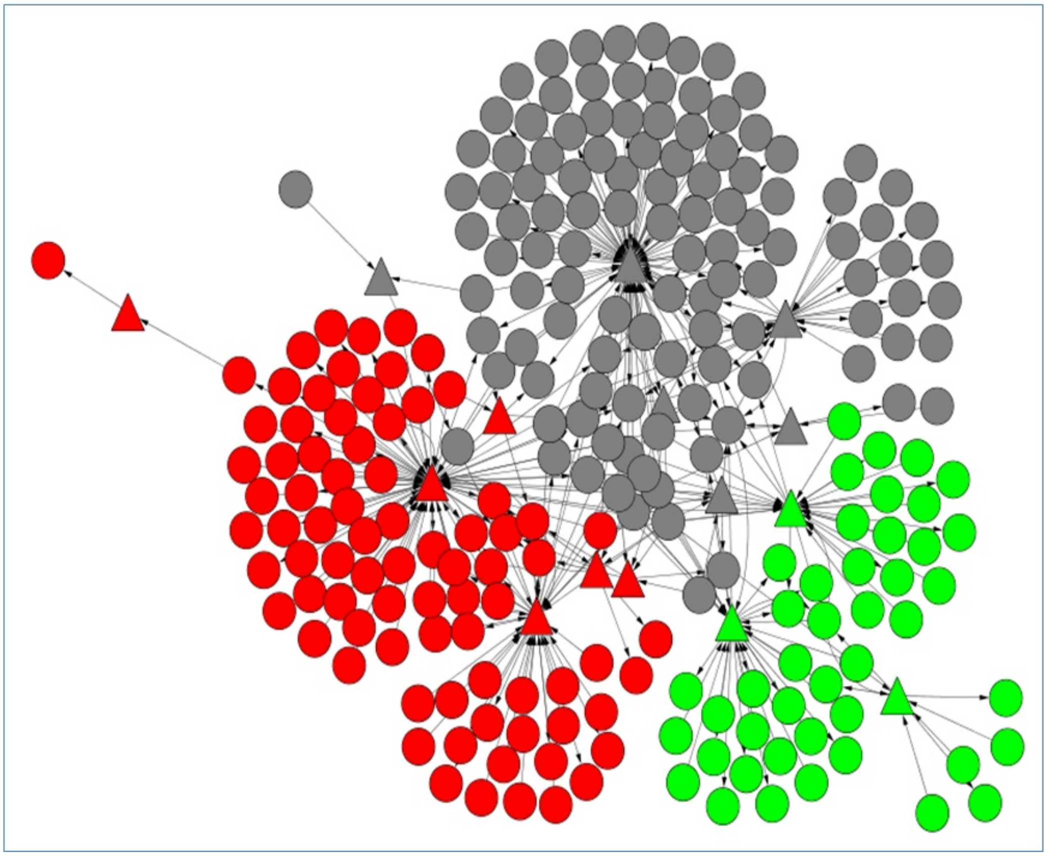


Figure 91: Detailed Information for Topic 8

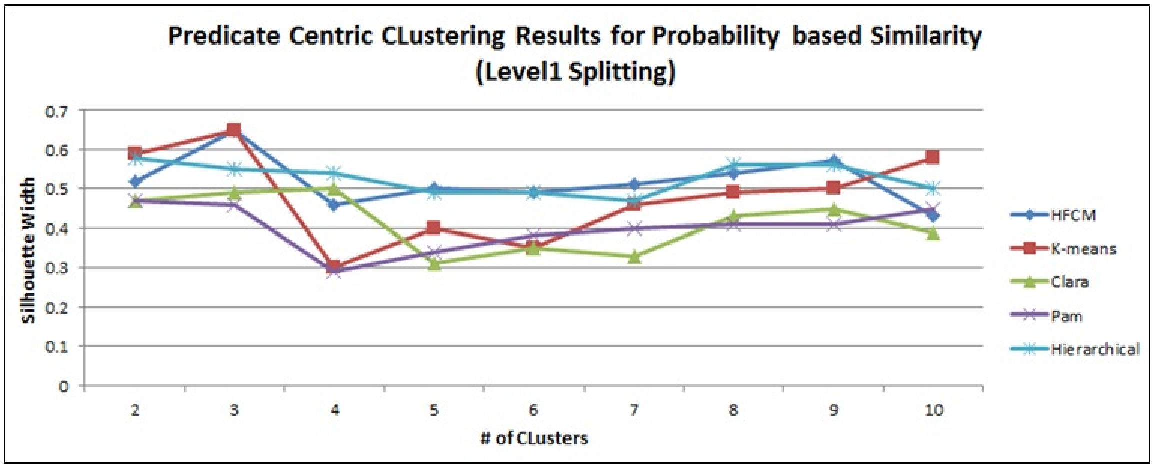


Figure 92: Predicate Oriented Clustering Decision Making on Different Levels (A)

highest Silhouette Width for level 2-1, level 2-2 and level 2-3 splitting, which are 0.52, 0.55 and 0.58 with HFCM, respectively. This explains the reason why the Intermediate 1 Cluster is split into 3 clusters, Intermediate 2 Cluster is split into three clusters and Intermediate 3 Cluster is split into two clusters.

5.4.4 Validation of Clustering Results with Golden Standard

For those eight generated topics, we use a golden standard file provided by a medical expert, Dr. David W. Larson, in the Colon and Rectal Surgery department at the Mayo Clinics to validate our clustering output. This file lists indications of seven types of complications for 1505 patients after colorectal surgery from 2005 to 2013. In our study, we consider six types of complications by treating $ABSCESS$ and $LEAK$ as the same complication (unlike the golden standard) for the sake of simplicity. A patient may have had no complications or up to seven complications as the golden standard specified. We build correlation metrics based on this benchmark to find out which complications showed

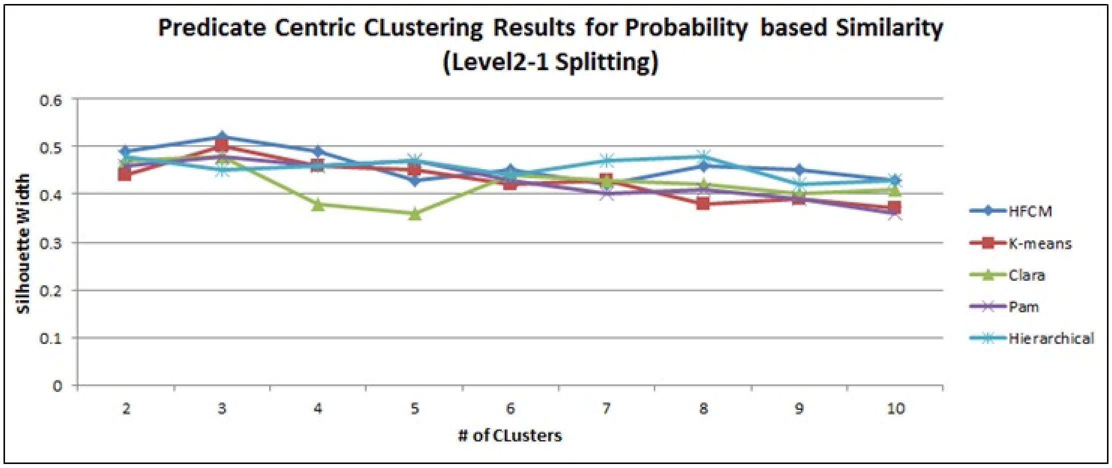


Figure 93: Predicate Oriented Clustering Decision Making on Different Levels (B)

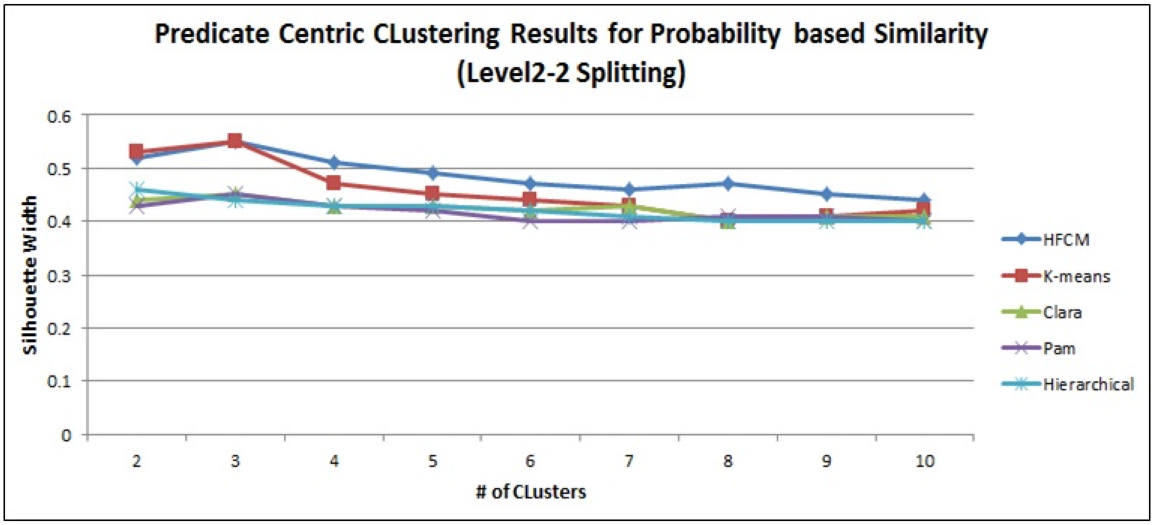


Figure 94: Predicate Oriented Clustering Decision Making on Different Levels (C)

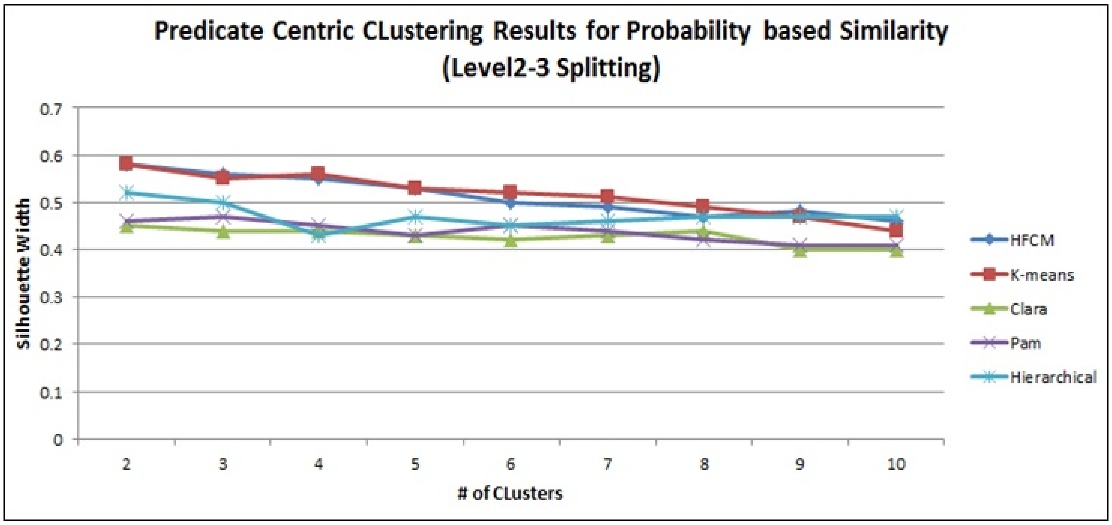
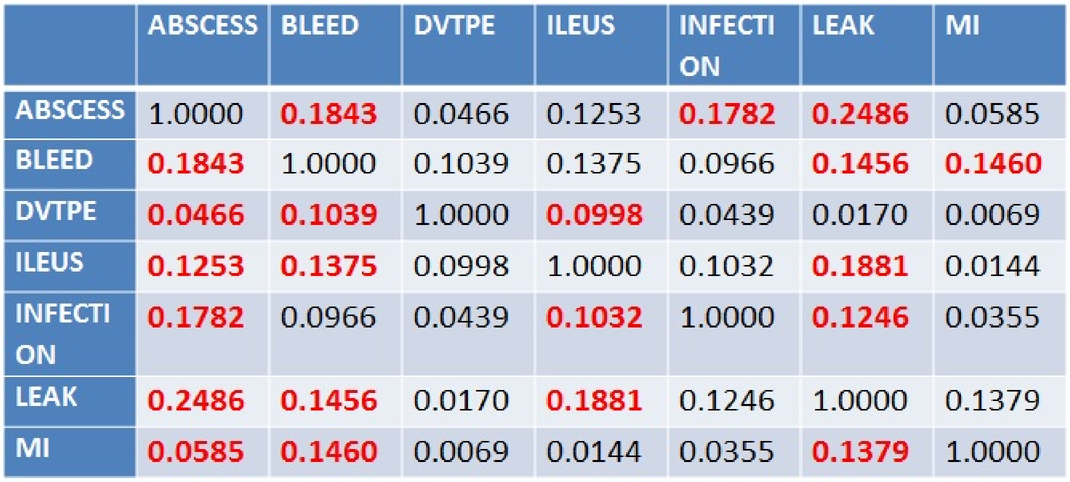


Figure 95: Predicate Oriented Clustering Decision Making on Different Levels (D)

a strong positive correlation. Fig. 96 represents the matrices with visualization. The number in red represents the top 3 relative strongest correlations for each complication. It is obvious to see that the complications $ABCESS$, $BLEED$ and $INFECTION$ have a relative stronger correlation than other complications that verifies that Topic 1 and Topic 2 are valid. $ILEUS$ has a relative stronger relationship with $ABSCESS$ and $BLEED$, verifying that Topic 6 is valid. $LEAK$ and $ILEUS$ are also strongly associated, verifying that Topic 7 is valid. MI is strongly related to $ABSCESS$ and $BLEED$, and we can also verify that Topic 8 is valid. $DVTPE$ does not have a very strong relationship with other complications, but this weak correlation with $ILEUS$, $BLEED$ and $ABSCESS$ is captured by Topics 3, 4, 5 and 6. Therefore, the clusters generated by the HFCM follow the same correlation provided by the golden standard benchmark.



	ABSCESS	BLEED	DVTPE	ILEUS	INFECTION	LEAK	MI
ABSCESS	1.0000	**0.1843**	0.0466	0.1253	**0.1782**	**0.2486**	0.0585
BLEED	**0.1843**	1.0000	0.1039	0.1375	0.0966	**0.1456**	**0.1460**
DVTPE	**0.0466**	**0.1039**	1.0000	**0.0998**	0.0439	0.0170	0.0069
ILEUS	**0.1253**	**0.1375**	0.0998	1.0000	0.1032	**0.1881**	0.0144
INFECTION	**0.1782**	0.0966	0.0439	**0.1032**	1.0000	**0.1246**	0.0355
LEAK	**0.2486**	**0.1456**	0.0170	**0.1881**	0.1246	1.0000	0.1379
MI	**0.0585**	**0.1460**	0.0069	0.0144	0.0355	**0.1379**	1.0000

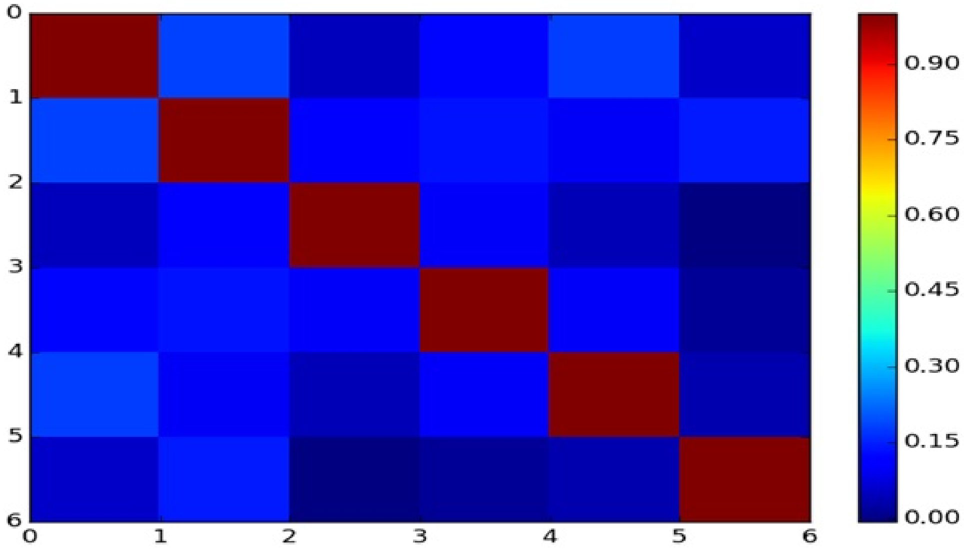


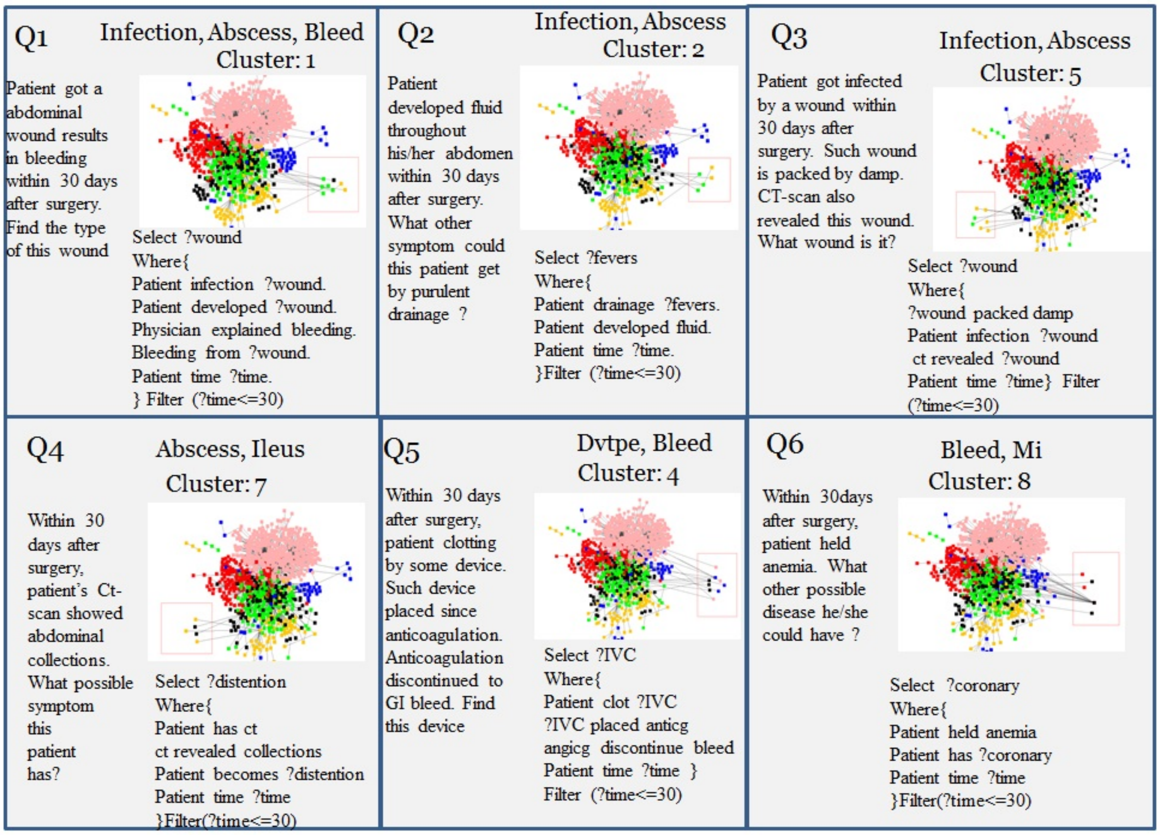
Figure 96: Correlation Matrices for Golden Standard

5.4.5 Query Generation and Visualization

The SPARQL queries generated for each cluster are shown in Fig. 97 and 98. For the predicate graphs across six domains, we use a rectangle to identify the query boundary out of the whole predicate graph. Queries 1 to 6 are cross domain queries that are automatically generated from each cluster. These queries identify the relationships among different post-surgical complications. For example, $INFECTION$, $ABSCESS$ and $BLEED$ are closely related to each other through the predicates of wound, bleeding or fever. $DVTPE$ and $BLEED$ are usually related through the predicate clot. $ABSCESS$ and $ILEUS$ are usually related to each other through abdominal collections and distention. MI and $BLEED$ are closely related to each other through anemia and coronary. Queries 7 to 12 are about queries within a single complication. We also find some interesting query patterns for each of the six complications. For instance, in $ABSCESS$, sepsis usually comes with drainage. In $BLEED$, transfusion connects to anemia and hemoglobin. In $DVTPE$, coumadin and clot occur together. In $ILEUS$, ct scan and pelvis have a close relationship. In $INFECTION$, patients discontinue wound after the wound be packed. In MI, pressure and volume overload can be a good treatment for a problem exacerbated by radiation.

5.5 Summary

In this chapter, we have presented the idea of predicate based pattern analysis, investigated the use of ontology and applied an unsupervised machine learning approach to integrate a heterogeneous unstructured resource with a semi-structured knowledge base.



Q1 Infection, Abscess, Bleed Cluster: 1

Patient got a abdominal wound results in bleeding within 30 days after surgery. Find the type of this wound

```
Select ?wound
Where{
Patient infection ?wound.
Patient developed ?wound.
Physician explained bleeding.
Bleeding from ?wound.
Patient time ?time.
} Filter (?time<=30)
```

Q2 Infection, Abscess Cluster: 2

Patient developed fluid throughout his/her abdomen within 30 days after surgery. What other symptom could this patient get by purulent drainage ?

```
Select ?fevers
Where{
Patient drainage ?fevers.
Patient developed fluid.
Patient time ?time.
}Filter (?time<=30)
```

Q3 Infection, Abscess Cluster: 5

Patient got infected by a wound within 30 days after surgery. Such wound is packed by damp. CT-scan also revealed this wound. What wound is it?

```
Select ?wound
Where{
?wound packed damp
Patient infection ?wound
 ct revealed ?wound
Patient time ?time} Filter
(?time<=30)
```

Q4 Abscess, Ileus Cluster: 7

Within 30 days after surgery, patient's Ct-scan showed abdominal collections. What possible symptom this patient has?

```
Select ?distention
Where{
Patient has ct
ct revealed collections
Patient becomes ?distention
Patient time ?time
}Filter(?time<=30)
```

Q5 Dvtpe, Bleed Cluster: 4

Within 30 days after surgery, patient clotting by some device. Such device placed since anticoagulation. Anticoagulation discontinued to GI bleed. Find this device

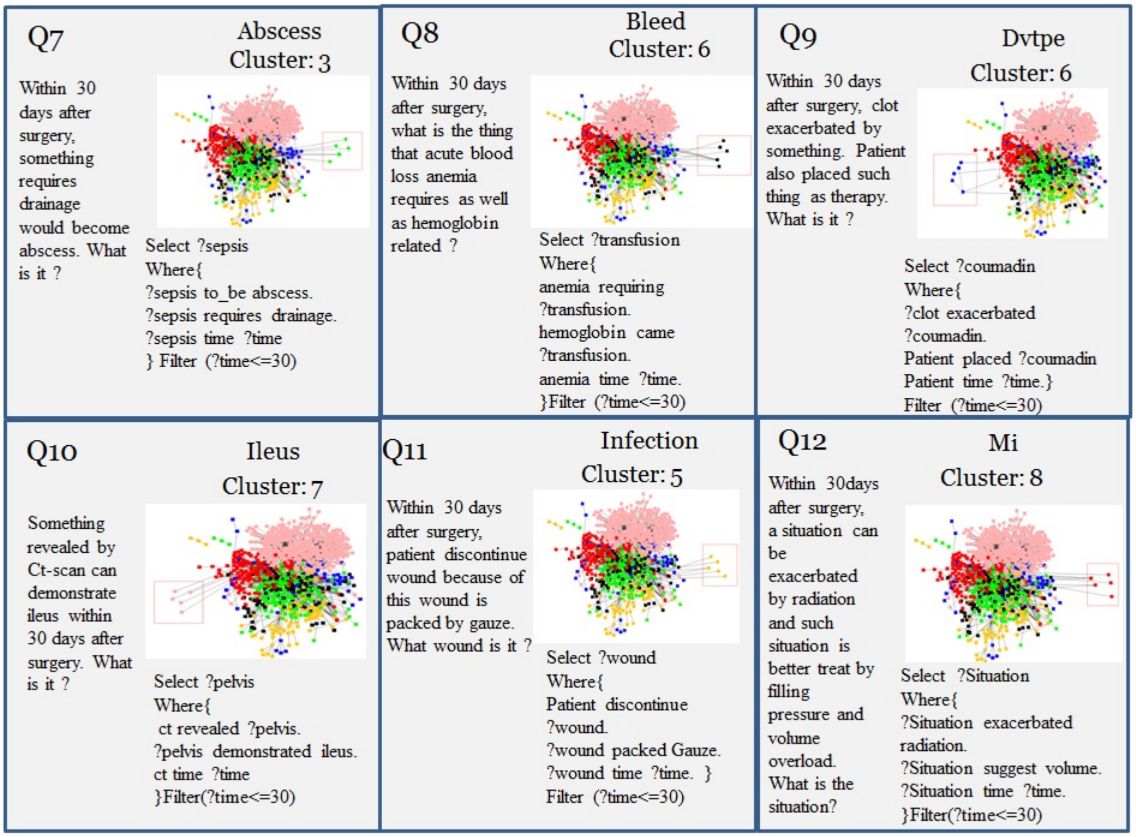
```
Select ?IVC
Where{
Patient clot ?IVC
?IVC placed anticg
angicg discontinue bleed
Patient time ?time }
Filter (?time<=30)
```

Q6 Bleed, Mi Cluster: 8

Within 30days after surgery, patient held anemia. What other possible disease he/she could have ?

```
Select ?coronary
Where{
Patient held anemia
Patient has ?coronary
Patient time ?time
}Filter(?time<=30)
```

Figure 97: Cross Complications Queries



Q7 Abscess Cluster: 3

Within 30 days after surgery, something requires drainage would become abscess. What is it ?

```
Select ?sepsis
Where{
?sepsis to_be abscess.
?sepsis requires drainage.
?sepsis time ?time
} Filter (?time<=30)
```

Q8 Bleed Cluster: 6

Within 30 days after surgery, what is the thing that acute blood loss anemia requires as well as hemoglobin related ?

```
Select ?transfusion
Where{
anemia requiring
?transfusion.
hemoglobin came
?transfusion.
anemia time ?time.
}Filter (?time<=30)
```

Q9 Dvtpe Cluster: 6

Within 30 days after surgery, clot exacerbated by something. Patient also placed such thing as therapy. What is it ?

```
Select ?coumadin
Where{
?clot exacerbated
?coumadin.
Patient placed ?coumadin
Patient time ?time.}
Filter (?time<=30)
```

Q10 Ileus Cluster: 7

Something revealed by Ct-scan can demonstrate ileus within 30 days after surgery. What is it ?

```
Select ?pelvis
Where{
 ct revealed ?pelvis.
?pelvis demonstrated ileus.
ct time ?time
}Filter(?time<=30)
```

Q11 Infection Cluster: 5

Within 30 days after surgery, patient discontinue wound because of this wound is packed by gauze. What wound is it ?

```
Select ?wound
Where{
Patient discontinue
?wound.
?wound packed Gauze.
?wound time ?time. }
Filter (?time<=30)
```

Q12 Mi Cluster: 8

Within 30days after surgery, a situation can be exacerbated by radiation and such situation is better treat by filling pressure and volume overload. What is the situation?

```
Select ?Situation
Where{
?Situation exacerbated
radiation.
?Situation suggest volume.
?Situation time ?time.
}Filter(?time<=30)
```

Figure 98: Single Complication Queries

In application level, we have achieved specific topic based pattern analysis as well as query generation for cross domain knowledge discovery. The GraphKDD framework is proposed to process any RDF/OWL datasets from heterogeneous resources. For the evaluation purpose, we have adopted a case study with colorectal post-surgical complications and demonstrated that the GraphKDD framework is capable of extracting a cohesive structure and semantics, as well as interesting patterns from structured/unstructured complication datasets. By using the colorectal surgical reports from the Mayo Clinic and golden standard, we have successfully validated our clustering results, thereby providing solid evidence for automatic query generation.

Table 36: Definition of Colorectal Post-surgical Complication

Post-surgical Complication	**Description**
Abscess/Leak (ABSCESS)	An abscess is a painful collection of pus, usually caused by a bacterial infection. Coloanal anastomoses have the highest rates.
Bleeding (BLEED)	Minor and major bleeding is common in anastomotic complications. Epinephrine and saline retention enemas are used to manage serious bleeding. Surgical intervention is necessary if situation is getting worse.
Deep vein thrombosis (DVT)/pulmonary embolism (PE) (DVTPE)	DVT is a condition wherein a blood clot forms in a vein of the deep system. A piece of the clot can break off and travel through the lung, which can cause heart failure, known as PE.
Ileus (ILEUS)	Ileus is defined as bowel obstruction. For small bowel obstruction, 90-100% sensitivity can be achieved by a CT scan of the abdomen and pelvis.
Myocardial infraction (MI)	Myocardial infarction is commonly known as a heart attack. It occurs during surgery or within 30 days after surgery.
Wound infection,(INFECTION)	Wound infections commonly present around the fifth post-surgical day and 5-15% of patients have such complication after colorectal surgery.

Table 37: Statistics of Six Complication Ontologies

	# of Subjects	**# of Predicates**	**# of Objects**	**# of Unique Triples**
ABSCESS	63	13	89	220
BLEED	58	13	73	142
DVTPE	19	10	26	32
ILEUS	227	21	204	624
MI	52	12	53	132
INFECTION	26	14	37	60
Total	**445**	**83**	**482**	**1210**

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

We presented an innovative predicate oriented pattern analysis and knowledge discovery framework called the GraphKDD. Contributions are summarized as follows.

- A Predicate Oriented neighborhood Patterns (PONP) analysis model to quantify the close relationship among different RDF predicates with cross domains knowledge bases;
- A Hierarchical Predicate oriented K-Means clustering (HPKM) and a Predicate oriented Hierarchical Agglomerative clustering (PHAL) approach to partition graphs into small semantically related sub-graphs with different purposes;
- A dynamic query generation algorithm from outputs of topic discovery;
- A source and target reachable topic aware link discovery algorithm to efficiently find paths with context between the source and target nodes;
- An ontology learning integrated framework is proposed to extract key words from unstructured data with a natural language processing technique and build an ontology based on retrieved words for further analysis;
- Comprehensive experimental evaluations to validate proposed methodologies of the

framework using DBPedia [12], Yago [115] and Bio2RDF [10] datasets specifically. The GraphKDD is proposed based on a Predicate Oriented Neighborhood Patterns (PONP). PONP is an innovative pattern analysis methodology that performs a similarity measurement for pairwise RDF predicates. PONP applies a predicate based neighborhood similarity strategy to measure the close relationship among predicates in order to partition the RDF graph based on such a relationship. Specifically, PONP adopts a dynamic neighborhood weightage association measurement to quantify similarity scores between each pair of predicates in an RDF graph. If two predicates are closer, they maintain a higher similarity score; otherwise, they keep a lower similarity score. Evaluation of PONP shows the predicate oriented approach gives a better average similarity score than a concept oriented one. Analysis results of PONP also show the advantage of using a predicate pattern based approach especially for cross domains data sets.

Based on similarity matrices generated by PONP, the GraphKDD is able to apply top-down and bottom-up unsupervised clustering on it to partition the RDF graph into different topics, each topic holds a group of RDF predicates and their related concepts to form a specific collection of knowledge. For top-down unsupervised learning, we proposed a Hierarchical Predicate oriented K-Means clustering algorithm. This approach combines the advantages of K-Means clustering and hierarchical clustering to provide a topic hierarchy solution with global similarity optimization. For bottom-up unsupervised learning, we proposed a Predicate oriented Hierarchical Agglomerative clustering algorithm. This approach focuses more on local diversity optimization that is able to find relationships among cross domains predicates in addition to group heterogeneous resources. Evaluations showed the best predicate oriented neighborhood radius boundary and branching factor with two different approaches. In addition, our experiment showed the comparison between a predicate oriented approach and concept oriented approach with two clustering algorithms. Both show the optimal solution with the predicate oriented pattern based approach. In addition, to validate the good performance of the GraphKDD framework on a context aware graph partition, we also conducted a comparison study between the GraphKDD and four random graph partition algorithms (Random Vertex Cut, Canonical Random Vertex Cut, Edge Partition 1D and Edge Partition 2D). Results also showed the meaningful and reasonable partition results coming from the GraphKDD when compared to random graph partition algorithms. In addition, as the comparison result from the GraphKDD with the famous knowledge discovery tool Latent Dirichlet allocation (LDA) showing, the GraphKDD is better to handle graph data knowledge discovery. Topics generated by the GraphKDD can also be used as a classifier to categorize new predicates into the right buckets.

The automatic query generation tool and topic aware linking discover tool are two applications derived from the GraphKDD framework. The automatic query generation tool is able to compose specific SPARQL queries from each topic that describes several scenarios within the context of each topic. The topic aware linking discovery tool is capable of finding all the paths between any source and target nodes with topic context awareness. We demonstrated two applications with different data sets. Results showed that both applications are able to discover knowledge in a context aware manner.

As an extension of the GraphKDD framework, we added an ontology learning feature to enable the GraphKDD to retrieve keywords from the natural language and convert such keywords to ontology for further semantic analysis. In evaluation, we used a Mayo Clinic colorectal patients cohort that contains seven types of complications for 1505 patients after colorectal surgery from 2005 to 2013. We applied mutual information to extract keywords from the cohort and built an ontology based on them. Then we used PONP to analyzed ontology graph and used Hierarchical Fuzzy-Cmeans (HFCM) clustering algorithm to find topics. Each topic explained the correlation and association among certain types of post-surgical complications.

6.2 Future Work

For future work, we will add more semantic analysis features into the GraphKDD framework. RDF predicates like $sameAs$, $seeAlso$, $subclassOf$, etc. will be added to construct the semantic analysis pattern to build the bridge between two predicates that don't share any k level neighborhood. This semantic analysis could help to find a hidden relationship. In addition, we will use the GraphKDD schema level topic discovery results to guide the information extraction process in the RDF/OWL instance level. We will also provide a big data processing platform to handle large among of RDF instances. Interactive query generation and query processing tool will also be extended on the current framework. We will analyze the users' frequent query to guide the future query generation and provide a topic/query repository for different users.

REFERENCE LIST

[1] Alani, H., and Brewster, C. Ontology ranking based on the analysis of concept structures. In *Proceedings of the 3rd international conference on Knowledge Capture* (2005), ACM, pp. 51–58.

[2] Alexander, K., Cyganiak, R., Hausenblas, M., and Zhao, J. Describing Linked Datasets. In *LDOW* (2009).

[3] Anyanwu, K., Maduko, A., and Sheth, A. Sparq2l: towards support for subgraph extraction queries in rdf databases. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 797–806.

[4] Araujo, R. F. *Getting Started with Eclipse Juno*. Packt Publishing Ltd, 2013.

[5] Assadi, H. Construction of a regional ontology from text and its use within a documentary system. In *Proceedings of the international conference on Formal Ontology and Information Systems (FOIS-98)* (1998).

[6] Aussenac-Gilles, N., and Sörgel, D. Text analysis for ontology and terminology engineering. *Applied Ontology 1*, 1 (2005), 35–46.

[7] Baorto, D., Li, L., and Cimino, J. J. Practical experience with the maintenance and auditing of a large medical ontology. *Journal of Biomedical Informatics 42*, 3 (2009), 494–503.

[8] Bauer, F., and Kaltenböck, M. Linked open data: The essentials. *Edition mono/monochrom, Vienna* (2011).

[9] Bechhofer, S. OWL: Web ontology language. In *Encyclopedia of Database Systems*. Springer, 2009, pp. 2008–2009.

[10] Belleau, F., Nolin, M.-A., Tourigny, N., Rigault, P., and Morissette, J. Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics 41*, 5 (2008), 706–716.

[11] Bezdek, J. C., Ehrlich, R., and Full, W. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences 10*, 2 (1984), 191–203.

[12] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. DBpedia-A crystallization point for the Web of Data. *Web Semantics: science, services and agents on the World Wide Web 7*, 3 (2009), 154–165.

[13] Bizer, C., and Schultz, A. The berlin sparql benchmark, 2009.

[14] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *the Journal of Machine Learning Research 3* (2003), 993–1022.

[15] Cai, M., and Frank, M. RDFPeers: a scalable distributed RDF repository based on a structured peer-to-peer network. In *Proceedings of the 13th international conference on World Wide Web* (2004), ACM, pp. 650–657.

[16] Califf, M. E., and Mooney, R. J. Bottom-up relational learning of pattern matching rules for information extraction. *The Journal of Machine Learning Research 4* (2003), 177–210.

[17] Callahan, A., Cruz-Toledo, J., Ansell, P., Klassen, D., Tumarello, G., and Dumontier, M. Improved Dataset Coverage and Interoperability with Bio2RDF Release 2. In *SWAT4LS* (2012).

[18] Caraballo, S. A. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* (1999), Association for Computational Linguistics, pp. 120–126.

[19] Chen, B., Ding, Y., and Wild, D. J. Assessing drug target association using semantic linked data. *PLoS Comput Biol 8*, 7 (2012), e1002574.

[20] Chen, B., Dong, X., Jiao, D., Wang, H., Zhu, Q., Ding, Y., and Wild, D. J. Chem2Bio2RDF: a semantic framework for linking and data mining chemogenomic and systems chemical biology data. *BMC Bioinformatics 11*, 1 (2010), 255.

[21] Chen, L., Zeng, W.-M., Cai, Y.-D., Feng, K.-Y., and Chou, K.-C. Predicting anatomical therapeutic chemical (ATC) classification of drugs by integrating chemical-chemical interactions and similarities. *PloS one 7*, 4 (2012), e35254.

[22] Chen, M.-S., Han, J., and Yu, P. S. Data mining: an overview from a database perspective. *Knowledge and data Engineering, IEEE Transactions on 8*, 6 (1996), 866–883.

[23] Cheung, K., and Marshall, M. HCLSIG BioRDF Subgroup. Query Federation. Use case 2-microarray.

[24] Church, K. W., and Hanks, P. Word association norms, mutual information, and lexicography. *Computational Linguistics 16*, 1 (1990), 22–29.

[25] Cimiano, P., Hotho, A., and Staab, S. Comparing conceptual, divise and agglomerative clustering for learning taxonomies from text. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004* (2004).

[26] Ciravegna, F., Chapman, S., Dingli, A., and Wilks, Y. Learning to harvest information for the semantic web. In *The Semantic Web: Research and Applications*. Springer, 2004, pp. 312–326.

[27] Dagan, I., Marcus, S., and Markovitch, S. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics* (1993), Association for Computational Linguistics, pp. 164–171.

[28] Das, S., Agrawal, D., and El Abbadi, A. G-store: a scalable data store for transactional multi key access in the cloud. In *Proceedings of the 1st ACM symposium on Cloud Computing* (2010), ACM, pp. 163–174.

[29] Dembélé, D., and Kastner, P. Fuzzy C-means method for clustering microarray data. *Bioinformatics 19*, 8 (2003), 973–980.

[30] Detwiler, L., Suciu, D., and Brinkley, J. F. Regular paths in SparQL: querying the NCI Thesaurus. In *AMIA* (2008), Citeseer.

[31] Dijkstra, F., van der Ham, J., Grosso, P., and de Laat, C. A path finding implementation for multi-layer networks. *Future Generation Computer Systems 25*, 2 (2009), 142–146.

[32] Doan, A., Domingos, P. M., and Levy, A. Y. Learning Source Description for Data Integration. In *WebDB (Informal Proceedings)* (2000), pp. 81–86.

[33] Dos Santos, D. A., and Deutsch, R. The Positive Matching Index: A new similarity measure with optimal characteristics. *Pattern Recognition Letters 31*, 12 (2010), 1570–1576.

[34] Dumontier, M., Baker, C. J., Baran, J., Callahan, A., Chepelev, L. L., Cruz-Toledo, J., Nicholas, R., Rio, D., Duck, G., Furlong, L. I., et al. The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery. *J. Biomedical Semantics 5* (2014), 14.

[35] Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam, M. Open Information Extraction: The Second Generation. In *IJCAI* (2011), vol. 11, pp. 3–10.

[36] Fader, A., Soderland, S., and Etzioni, O. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2011), Association for Computational Linguistics, pp. 1535–1545.

[37] Faure, D., and Nédellec, C. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *LREC workshop on adapting lexical and corpus resources to sublanguages and applications* (1998), vol. 707, p. 30.

[38] Faure, D., and Nedellec, C. Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM. In *Knowledge Acquisition, Modeling and Management*. Springer, 1999, pp. 329–334.

[39] Formica, A. Ontology-based concept similarity in formal concept analysis. *Information Sciences 176*, 18 (2006), 2624–2641.

[40] Gerbessiotis, A. V., and Valiant, L. G. Direct bulk-synchronous parallel algorithms. *Journal of Parallel and Distributed Computing 22*, 2 (1994), 251–267.

[41] Glitho, R. Mapping function and method of transmitting signaling system 7 (SS7) telecommunications messages over data networks, Jan. 23 2001. US Patent 6,178,181.

[42] Glover, E., Pennock, D. M., Lawrence, S., and Krovetz, R. Inferring hierarchical descriptions. In *Proceedings of the eleventh international conference on Information and Knowledge Management* (2002), ACM, pp. 507–514.

[43] Godoy, M. J. G., López-Camacho, E., Navas-Delgado, I., and Aldana-Montes, J. F. Sharing and executing linked data queries in a collaborative environment. *Bioinformatics* (2013), btt192.

[44] Gotz, D., Wang, F., and Perer, A. A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data. *Journal of Biomedical Informatics 48* (2014), 148–159.

[45] Guo, Y., Pan, Z., and Heflin, J. LUBM: A benchmark for OWL knowledge base systems. *Web Semantics: Science, Services and Agents on the World Wide Web 3*, 2 (2005), 158–182.

[46] Hahn, U., and Schnattinger, K. Ontology engineering via text understanding. In *Proceedings of the 15th World Computer Congressâ The Global Information Society on the Way to the Next Milleniumâ(IFIPâ98)* (1998), Citeseer.

[47] Hahn, U., and Schnattinger, K. Towards text knowledge engineering. *Hypothesis 1*, 2 (1998).

[48] Hamp, B., Feldweg, H., et al. Germanet-a lexical-semantic net for german. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications* (1997), Citeseer, pp. 9–15.

[49] Hartigan, J. A., and Wong, M. A. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 28*, 1 (1979), 100–108.

[50] Hassanzadeh, O., Kementsietsidis, A., Lim, L., Miller, R. J., and Wang, M. Linkedct: A linked data space for clinical trials. *arXiv preprint arXiv:0908.0567* (2009).

[51] Hearst, M. A. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational Linguistics-Volume 2* (1992), Association for Computational Linguistics, pp. 539–545.

[52] Hearst, M. A. Automated discovery of WordNet relations. *WordNet: an electronic lexical database* (1998), 131–153.

[53] Hindle, D. Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics* (1990), Association for Computational Linguistics, pp. 268–275.

[54] Hornik, K. The comprehensive R archive network. *Wiley Interdisciplinary Reviews: Computational Statistics 4*, 4 (2012), 394–398.

[55] Huang, Z., Dong, W., Ji, L., Gan, C., Lu, X., and Duan, H. Discovery of clinical pathway patterns from event logs using probabilistic topic models. *Journal of Biomedical Informatics 47* (2014), 39–57.

[56] Inkpen, D. Z., and Hirst, G. Automatic sense disambiguation of the near-synonyms in a dictionary entry. In *Computational Linguistics and Intelligent Text Processing*. Springer, 2003, pp. 258–267.

[57] Jaccard, P. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

[58] Janik, M., and Kochut, K. BRAHMS: a workbench RDF store and high performance memory system for semantic association discovery. In *The Semantic Web–ISWC 2005*. Springer, 2005, pp. 431–445.

[59] Jannink, J., and Wiederhold, G. Thesaurus entry extraction from an on-line dictionary. In *Proceedings of Fusion* (1999), vol. 99, Citeseer.

[60] Jeh, G., and Widom, J. SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (2002), ACM, pp. 538–543.

[61] Jena, A. semantic web framework for Java, 2007.

[62] Johnson, S. C. Hierarchical clustering schemes. *Psychometrika 32*, 3 (1967), 241–254.

[63] Kaufman, L., and Rousseeuw, P. *Clustering by means of medoids*. North-Holland, 1987.

[64] Khayyat, Z., Awara, K., Alonazi, A., Jamjoom, H., Williams, D., and Kalnis, P. Mizan: a system for dynamic load balancing in large-scale graph processing. In

Proceedings of the 8th ACM European Conference on Computer Systems (2013), ACM, pp. 169–182.

[65] Kietz, J.-U., Maedche, A., and Volz, R. A method for semi-automatic ontology acquisition from a corporate intranet. In *EKAW-2000 Workshop âOntologies and Textâ, Juan-Les-Pins, France, October 2000* (2000).

[66] Klyne, G., and Carroll, J. J. Resource description framework (RDF): Concepts and abstract syntax.

[67] Kokkinakis, D. MEDLEX: Technical report. *Department of Swedish Language, Språkdata, Göteborg University.[www].¡ http://demo. spraakdata. gu. se/svedk/pbl/MEDLEX work2004. pdf¿ Retrieved January 9* (2004), 2007.

[68] Kompella, K., and Rekhter, Y. OSPF extensions in support of generalized multiprotocol label switching (GMPLS).

[69] Lang, J., and Lapata, M. Similarity-driven semantic role induction via graph partitioning. *Computational Linguistics 40*, 3 (2014), 633–669.

[70] Lasko, T. A., Denny, J. C., and Levy, M. A. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PloS one 8*, 6 (2013), e66341.

[71] Lee, Y., Supekar, K., and Geller, J. Ontology integration: Experience with medical terminologies. *Computers in Biology and Medicine 36*, 7 (2006), 893–919.

[72] Li, H., and Abe, N. Clustering words with the MDL principle. *Natural Language Processing (China) 4*, 2 (1997), 71–88.

[73] Lin, D. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational Linguistics-Volume 2* (1998), Association for Computational Linguistics, pp. 768–774.

[74] Liu, H., Sohn, S., Murphy, S., Lovely, J., Burton, M., Naessens, J., and Larson, D. W. Facilitating post-surgical complication detection through sublanguage analysis. *AMIA Summits on Translational Science Proceedings 2014* (2014), 77.

[75] Loizou, A., Angles, R., and Groth, P. On the formulation of performant sparql queries. *Web Semantics: Science, Services and Agents on the World Wide Web 31* (2015), 1–26.

[76] Lorey, J., and Naumann, F. Detecting SPARQL query templates for data prefetching. In *The Semantic Web: Semantics and Big Data*. Springer, 2013, pp. 124–139.

[77] Luciano, J. S., Andersson, B., Batchelor, C., Bodenreider, O., Clark, T., Denney, C. K., Domarew, C., Gambet, T., Harland, L., Jentzsch, A., et al. The Translational Medicine Ontology and Knowledge Base: driving personalized medicine by bridging the gap between bench and bedside. *Journal of Biomedical Semantics 2*, 2 (2011), 1.

[78] Maedche, A., and Staab, S. Mining ontologies from text. In *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*. Springer, 2000, pp. 189–202.

[79] Malewicz, G., Austern, M. H., Bik, A. J., Dehnert, J. C., Horn, I., Leiser, N., and Czajkowski, G. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (2010), ACM, pp. 135–146.

[80] Meymandpour, R., and Davis, J. G. A Semantic Similarity Measure for Linked Data: An Information Content-Based Approach.

[81] Missikoff, M., Navigli, R., and Velardi, P. Integrated approach to web ontology learning and engineering. *Computer 35*, 11 (2002), 60–63.

[82] Momtchev, V., Peychev, D., Primov, T., and Georgiev, G. Expanding the pathway and interaction knowledge in linked life data. *Proc. of International Semantic Web Challenge* (2009).

[83] Nekrutenko, A., and Taylor, J. Next-generation sequencing data interpretation: enhancing reproducibility and accessibility. *Nature Reviews Genetics 13*, 9 (2012), 667–672.

[84] Neumann, T., and Weikum, G. RDF-3X: a RISC-style engine for RDF. *Proceedings of the VLDB Endowment 1*, 1 (2008), 647–659.

[85] Ngomo, A.-C. N., and Auer, S. Limes-a time-efficient approach for large-scale link discovery on the web of data. *integration 15* (2011), 3.

[86] Nikolić, M. Measuring similarity of graph nodes by neighbor matching. *Intelligent Data Analysis 16*, 6 (2012), 865–878.

[87] Papailiou, N., Konstantinou, I., Tsoumakos, D., Karras, P., and Koziris, N. H 2 RDF+: High-performance distributed joins over large-scale RDF graphs. In *Big Data, 2013 IEEE International Conference on* (2013), IEEE, pp. 255–263.

[88] Pereira, F., Tishby, N., and Lee, L. Distributional clustering of English words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics* (1993), Association for Computational Linguistics, pp. 183–190.

[89] Prud, E., Seaborne, A., et al. Sparql query language for rdf.

[90] Quilitz, B., and Leser, U. *Querying distributed RDF data sources with SPARQL.* Springer, 2008.

[91] Rafiq, M. I., O'Connor, M. J., and Das, A. K. Computational method for temporal pattern discovery in biomedical genomic databases. In *Computational Systems Bioinformatics Conference, 2005. Proceedings. 2005 IEEE* (2005), IEEE, pp. 362–365.

[92] Reinberger, M.-L., and Daelemans, W. Unsupervised text mining for ontology learning. *Machine Learning for the Semantic Web, Schloss Dagstuhl, Waden, Germany* (2005).

[93] Rekhter, Y., and Li, T. A border gateway protocol 4 (BGP-4).

[94] Riloff, E., et al. Automatically constructing a dictionary for information extraction tasks. In *AAAI* (1993), pp. 811–816.

[95] Riloff, E., and Shepherd, J. A corpus-based approach for building semantic lexicons. *arXiv preprint cmp-lg/9706013* (1997).

[96] Roark, B., and Charniak, E. Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. In *Proceedings of the 17th international conference on Computational Linguistics-Volume 2* (1998), Association for Computational Linguistics, pp. 1110–1116.

[97] Ruge, G. Experiments on linguistically-based term associations. *Information Processing & Management 28*, 3 (1992), 317–332.

[98] Rydin, S. Building a hyponymy lexicon with hierarchical structure. In *Proceedings of the ACL-02 workshop on Unsupervised Lexical Acquisition-Volume 9* (2002), Association for Computational Linguistics, pp. 26–33.

[99] Sanderson, M., and Croft, B. Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval* (1999), ACM, pp. 206–213.

[100] Schlobach, S. Assertional mining in description logics. In *In Proceedings of the 2000 International Workshop on Description Logics (DL2000* (2000), Citeseer.

[101] Schmidt, M., Görlitz, O., Haase, P., Ladwig, G., Schwarte, A., and Tran, T. Fedbench: A benchmark suite for federated semantic data query processing. In *The Semantic Web–ISWC 2011*. Springer, 2011, pp. 585–600.

[102] Schmidt, M., Hornung, T., Lausen, G., and Pinkel, C. SPˆ 2Bench: a SPARQL performance benchmark. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on* (2009), IEEE, pp. 222–233.

[103] Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research 13*, 11 (2003), 2498–2504.

[104] Shaw, M., Detwiler, L., Brinkley, J. F., and Suciu, D. Generating application ontologies from reference ontologies. In *AMIA* (2008).

[105] Shaw, M., Detwiler, L. T., Noy, N., Brinkley, J., and Suciu, D. vSPARQL: A view definition language for the semantic web. *Journal of Biomedical Informatics 44*, 1 (2011), 102–117.

[106] Shen, F., and Lee, Y. Knowledge Discovery from Biomedical Ontologies in Cross Domains. In *(Manuscript submitted for publication. March 7, 2016) Available at https://www.dropbox.com/s/0dzhgqiqneokop0/MedKDD.pdf?dl=0.*

[107] Shen, F., and Lee, Y. MedTQ: Dynamic Topic Discovery and Query Generation for Medical Ontologies. In *(Manuscript submitted for publication. Jan. 10, 2016) Available at https://www.dropbox.com/s/9dcpidh5fjdxv1t/MedTQ.pdf?dl=0*.

[108] Shen, F., Liu, H., Sohn, S., Larson, D. W., and Lee, Y. BmQGen: Biomedical query generator for knowledge discovery. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on* (2015), IEEE, pp. 1092–1097.

[109] Shen, F., Liu, H., Sohn, S., Larson, D. W., and Lee, Y. Predicate Oriented Pattern Analysis for Biomedical Knowledge Discovery. *Journal of Intelligent Information Management* (2016).

[110] Shi, B., and Weninger, T. Fact Checking in Large Knowledge Graphs-A Discriminative Predicate Path Mining Approach. *arXiv preprint arXiv:1510.05911* (2015).

[111] Simmhan, Y., Kumbhare, A., Wickramaarachchi, C., Nagarkar, S., Ravi, S., Raghavendra, C., and Prasanna, V. Goffish: A sub-graph centric framework for large-scale graph analytics. In *Euro-Par 2014 Parallel Processing*. Springer, 2014, pp. 451–462.

[112] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology 25*, 11 (2007), 1251–1255.

[113] Stuckenschmidt, H., and Klein, M. Structure-based partitioning of large concept hierarchies. In *The Semantic Web–ISWC 2004*. Springer, 2004, pp. 289–303.

[114] Sturn, A., Quackenbush, J., and Trajanoski, Z. Genesis: cluster analysis of microarray data. *Bioinformatics 18*, 1 (2002), 207–208.

[115] Suchanek, F. M., Kasneci, G., and Weikum, G. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 697–706.

[116] Tan, P.-N., Steinbach, M., Kumar, V., et al. *Introduction to data mining*, vol. 1. Pearson Addison Wesley Boston, 2006.

[117] Tari, Z., Bukhres, O., Stokes, J., and Hammoudi, S. The reengineering of relational databases based on key and data correlations. In *Data Mining and Reverse Engineering*. Springer, 1998, pp. 184–216.

[118] Thanopoulos, A., Fakotakis, N., and Kokkinakis, G. Automatic extraction of semantic relations from specialized corpora. In *Proceedings of the 18th conference on Computational Linguistics-Volume 2* (2000), Association for Computational Linguistics, pp. 836–842.

[119] Torii, M., Wagholikar, K., and Liu, H. Using machine learning for concept extraction on clinical documents from multiple data sources. *Journal of the American Medical Informatics Association 18*, 5 (2011), 580–587.

[120] Van der Laan, M., Pollard, K., and Bryan, J. A new partitioning around medoids algorithm. *Journal of Statistical Computation and Simulation 73*, 8 (2003), 575–584.

[121] van Leeuwen, M. Interactive data exploration using pattern mining. In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*. Springer, 2014, pp. 169–182.

[122] Viswanathan, V., and Krishnamurthi, I. Finding relevant semantic association paths through user-specific intermediate entities. *Human-centric Computing and Information Sciences 2*, 1 (2012), 1–11.

[123] Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G. Silk-A Link Discovery Framework for the Web of Data. *LDOW 538* (2009).

[124] Wang, H., Azuaje, F., and Black, N. An integrative and interactive framework for improving biomedical pattern discovery and visualization. *Information Technology in Biomedicine, IEEE Transactions on 8*, 1 (2004), 16–27.

[125] Warrender, J. D., and Lord, P. A pattern-driven approach to biomedical ontology engineering. *arXiv preprint arXiv:1312.0465* (2013).

[126] Wiemer-Hastings, P., Graesser, A., and Wiemer-Hastings, K. Inferring the meaning of verbs from context. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (1998), pp. 1142–1147.

[127] Williams, A. J., Harland, L., Groth, P., Pettifer, S., Chichester, C., Willighagen, E. L., Evelo, C. T., Blomberg, N., Ecker, G., Goble, C., et al. Open PHACTS: semantic interoperability for drug discovery. *Drug Discovery Today 17*, 21 (2012), 1188–1198.

[128] Xin, R. S., Gonzalez, J. E., Franklin, M. J., and Stoica, I. Graphx: A resilient distributed graph system on spark. In *First International Workshop on Graph Data Management Experiences and Systems* (2013), ACM, p. 2.

[129] Xu, F., Kurz, D., Piskorski, J., and Schmeier, S. Term extraction and mining of term relations from unrestricted texts in the financial domain. *Business Information Systems, Poznan, Poland* (2002).

[130] Yang, S., Yan, X., Zong, B., and Khan, A. Towards effective partition management for large graphs. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (2012), ACM, pp. 517–528.

[131] Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., and Soderland, S. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations* (2007), Association for Computational Linguistics, pp. 25–26.

[132] Zeng, K., Yang, J., Wang, H., Shao, B., and Wang, Z. A distributed graph engine for web scale RDF data. *Proceedings of the VLDB Endowment 6*, 4 (2013), 265–276.

[133] Zhou, L. Ontology learning: state of the art and open issues. *Information Technology and Management 8*, 3 (2007), 241–252.

[134] Zhou, Y., Liu, L., and Buttler, D. Integrating Vertex-centric Clustering with Edge-centric Clustering for Meta Path Graph Analysis. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), ACM, pp. 1563–1572.

[135] Ziegler, P., and Dittrich, K. R. User-specific semantic integration of heterogeneous data: The sirup approach. In *Semantics of a Networked World. Semantics for Grid Databases*. Springer, 2004, pp. 44–64.

VITA

Feichen Shen was born on Nov 2nd, 1987, in Nanjing, Jiangsu Province, China. He was educated in local public schools and graduated from Nanjing No.9 Senior High School in China, Jiangsu. After graduating from the public school, he studied as an undergraduate student in Computer Science and received Bachelor degree from Nanjing University, China in 2010.

After his undergraduate study, Mr. Shen moved to the United States and joined University of Missouri - Kansas City for pursing his Masters in Computer Science. He received his Masters degree of Computer Science at the University of Missouri - Kansas City on August 2012. At the same year, he was enrolled as an interdisciplinary Ph.D. student in University of Missouri - Kansas City supervised by Dr. Yugyung Lee with Computer Science as his coordinating discipline and Telecommunication & Computer Networking as his Co-discipline, respectively.

During Mr. Shen's Ph.D. study, he received research grants from UMKC School of Graduate Studies (SGS) to support his academia research in 2014 and 2015. He was also selected for the Graduate Student Professional Development Program, a leadership training program, by UMKC SGS as top 10 Ph.D. students. Mr. Shen was also awarded for the Outstanding Doctoral Student in 2014 by UMKC School of Computing and Engineering (SCE).

Mr. Shen worked as a Ph.D. intern in Department of Health Science Research at the Mayo Clinic on 2012, 2013 and 2015 summer.

After the completion of his degree requirement, Mr. Shen will join Mayo Clinic Department of Health Science Research as a research associate in Rochester, MN.