Research Article

Understanding Large-Scale Social Relationship Data by Combining Conceptual Graphs and Domain Ontologies

Zhao Huang\(^1\,^2\) and Liu Yuan\(^2\)

\(^1\)Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi’an, China
\(^2\)School of Computer Science, Shaanxi Normal University, Xi’an, China

Correspondence should be addressed to Liu Yuan; yuanliu@snnu.edu.cn

Received 27 April 2021; Revised 22 June 2021; Accepted 19 July 2021; Published 31 July 2021

1. Introduction

The popularity of social media involves real-time participation via the Internet, which empowers the connection of numerous users worldwide and promotes content and information sharing [1]. Such user activities generate valuable repositories of information [2], forming relationships between (or among) users online, which are called social relationships. Broadly, social relationships are the connections between users who have recurring interactions that are perceived by the participants to have personal meaning, including relationships of family members, friends, neighbours, coworkers, and other associates [3]. From a data perspective, these human-generated data have diverse social uses and rich meanings (for example, communication text, videos for entertainment and self-representation, and sharing of news and other 3rd party content in social media). Such unstructured and semistructured yet semantically rich data, also called big social data, have been argued to constitute 95% of all big data [4]. Understanding, analysing, and mining these social relationship data are powerful means to acquire new information for investigating social relationship-based applications, such as relationship prediction on social networks [4] and recommendations on social commerce [5].

A precondition of analysing and using these relationship data intensively is to understand and represent social relations accurately, completely, and effectively. From the perspective of knowledge representation in computer science, social relationship representation needs formal syntax and semantics for implementing regular browsing, querying, and reasoning and should provide mechanisms of machine readability and human readability in addition to data consistency, reliability, and high quality [6]. From the application perspective, social relationship representations should describe key elements of relationship, including relationship context, attributes of individuals, and attributes of relationship. Applications, such as recommendation
systems and fraud detection, have different information needs on using social relationship data; it will be handy if social data are organized in multiple granularity level and represented in multiple forms. For example, the ontology structure is useful for presenting individual semantics, but graph structure is more effective in discovering and predicting relationships [7]. Therefore, the representation of social relationships in an expressive and flexible way to adaptively meet the requirements of different applications is an issue to be dealt with intensively.

In this work, we propose a new way to represent social relationships according to the application requirements by exploiting conceptual graph structure and ontology terminologies. The contributions of this study are twofold:

A novel social relationship representation model that adapts to various application environments is developed with different levels of expressivity and inference capability to meet applications’ needs.

Our study also increases the understanding of social relationship query by designing a new hybrid query answering mechanism for the social relationship model on the basis of graph-based algorithms and ontology-based reasoning.

The rest of this paper is organized as follows: Section 2 presents background on social relationship from the sociology and computer science perspectives. Section 3 describes the semantic social relationship model in this study. Section 4 defines the formal representation of the proposed model. Section 5 implements the social relationship model and performs relationship discovery task. The application scenarios and experimental results are discussed in Section 6. Conclusion, implications, and future work are detailed in Section 7.

2. Related Work

2.1. Social Relationships in Sociology. The investigation of social relationships has been an active topic in both sociology and computer science research areas. In sociology, social relationships have been studied from the following different perspectives: problems including factors influencing social relations, relation categories, and the development of relations [8, 9], and there is much work focusing on the relationships and their effects on behaviour [10–12]. The concept of social representation was developed as a social psychological approach articulating individual thinking and feeling with collective interaction and communication [13], and different types of methods and techniques are employed to analyse social relationships [14, 15]. There are a large number of studies available focusing on analysing social relationships and their effect on human behaviours [16–18].

2.2. Social Relationship Representations in Computer Science. Social relationships have become a popular research area in the fields of computer and information science since the success of Web 2.0 techniques, and social media contributes a huge magnitude of daily data [18]. Social media is a typical example of big data sources. In particular, big data technologies that combine the conventional social media analytics demonstrate remarkable potential enabling processing of the flood of social media data [19–21]. As such, the new terms social big data (SBD) or big social data (BSD) are emerged, which designates the joint of big data technologies and frameworks with the traditional analysis techniques targeted to process and analyze social media data and with the aim of deriving useful value [22]. However, the alliance between the social media analytics and big data does not seem to explicitly discuss the representation of social media data.

Social data represent a precious mine of information that has attracted researchers from different domains. In this context, most of the literature studies focused on the social network topology or the communications between individuals, which represents the linkage information [23, 24]. Data fusion based on semantic linking aims at providing data integration and fusion between heterogeneous semantic data [25, 26]. This can be done in considerations of the semantic conflict detection and reconsolidation between the heterogeneous semantic data. Some studies (e.g., [27]) adopt neural network to help predict new information based on the individuals’ features and relationships between individuals. Some focus on semantical modelling interaction, collaboration, and communication between key elements in an intelligent environment [28, 29], and semantic data mining is often used to deal with the semantic relationship data [30]. However, it can be argued that there still lacks a unified, effective, and interpretable social relationship representation form that can integrate benefits from different forms. Because of the success of graph theory and its application, graphs are an intuitive way to represent relationships between social entities [31]. Graph representation allows the relational knowledge of interacting entities to be stored and accessed efficiently [32]. The analysis of graph data can provide significant insights into community detection [33], behaviour analysis [34], and other applications (e.g., node classification [35], link prediction [36], clustering [37], and recommendations [38, 39]).

The descriptions and formats of graphs stored in a computer vary, and graph embedding is the most popular numeric-based graph structure description. Various graph embedding techniques have been developed to convert raw graph data into a high-dimensional vector while preserving intrinsic graph properties [40–42]. This process is known as graph representation learning [43]. With a learned graph representation, one can adopt machine learning tools to conveniently perform downstream tasks [44]. Some research has focused on the complex interactions among a group of people, which are another informative resource to analyse patterns of social behaviours and characteristics affected by evolving dynamic social relationships [45, 46]. Although graph embedding techniques work well on analysing and mining social relationships, problems of integrating relations from different applications and visualizing, querying, and interpreting them remain. These issues require semantic-level relationship processing.
For a long time, the following two main types of graph models supporting semantic description and inference exist: resource description frameworks (RDFs) and labelled property graphs (LPFs) [47, 48]. RDF graphs are either triple or quad stores, and their chief benefits are interoperable standards that eliminate silos, so the ease of use of their data models and their inference support machine intelligence [5, 49]. The principal benefit of LPFs is the richness of detail they give data [50, 51]. These graphs allow users to quickly add properties to nodes to describe them with any additional information, so properties can easily enhance data with colourful descriptions. RDF * extends the RDF triple store graph to include the level of detail of LPFs that leverage properties [52]. RDF and LPF are basically fundamental semantic social relationship data models. The ontological representation of social networks such as Friend of a Friend (FOAF) is a very popular method in applications, but these ontologies still lack the ability to support the automated integration of social information on a semantic basis and capture established concepts in social network analysis. As a popular knowledge representation form, a knowledge graph can also be used to represent social relationships [53], and the large number of methods and techniques developed in the knowledge graph research area can be adapted to social relationship applications [54, 55].

An ontology that expresses the shared concepts and their relationships in a specific field can be used as a semantic framework for social media data analytics. Adopting ontology as background knowledge during data analysing will help in improving the performance. For example, an ontology-based name entity recognition approach was used to detect a traffic event and its location [56]. Moreover, Ali et al. [57] proposed an ontology-based recommendation system that sends text to the patient based on the monitored results from the healthcare domain. More importantly, the combination of the neural network with ontology has proved the efficiency in solving content and sentiment related problems (e.g., [58, 59]). However, it can be argued that ontology is not as flexible as the graph structure for representing relationship.

According to the discussion above, it is feasible to make expressive and flexible social relationship representations based on research achievements from different areas available. As fundamental metadata to understand social networks, social relationships provide distinctive features in many applications. In personal content applications, social ontology is used to annotate, browse, and retrieve documents; the social ontologies for online communities and networks help to manage the local data and interoperate semantically between different systems; in the domain of enterprise knowledge management, the categories of enterprise document management, and collaboration software are rapidly merging into integrated solutions where metadata regarding the personal profiles and social networks of experts is combined with metadata about the documents and the other content of the enterprise. We focus in this work to propose a novel social relationship representation by analysing and exploiting the current related work broadly and intensively.

3. Semantic Model of Social Relationships

The goal of constructing a semantic model of social relationships is to provide a comprehensive view of social relationship representation and application. The relationship users are usually concerned with come from more than one social network application. The information that users seek could be diverse, such as a person, an online document, relationship between people, or artefacts in any form generated on the social network platform. When users browse, query, search, or mine these relationships, in addition to accuracy and efficiency, they expect machines that can provide the information they need in a human-understandable way. Therefore, it is necessary to represent social relationships in an integrated, accurate, interpretable, and computable way. The features of social relationship representation lie in two aspects. One is from the users’ perspective, where all valuable information is included or implied in the integrated social representation, and the representation should support a user-friendly graphical information browse. Another is from the knowledge representation perspective, where machines can understand and deal with this representation form well, and the information retrieved or inferred from relationship datasets is provable and interpretable.

A model of social relationship representation is presented in Figure 1. The social relationship representation model consists of 4 layers. Layer 1 is the fundamental, where each social network application has its relationship data represented by ontology and conceptual graph. In layer 2, relevant social relationship datasets or user concerns are integrated into a united representation form based on the context. For example, the people connections in different social network applications could be integrated. The query about the relationships between two people should not be restricted to a single platform, but to the integrated connections data. Layer 3 (user interface) provides different views of the integrated social relationship representation. The benefits of offering multiple views are twofold: one is to reduce the complexity of the integrated representation by narrowing down the large amount of relationship information in an application. Another is to summarize the information from different perspectives. Within layer 4, users can access the relationships they need in their own applications. According to the discussion above, it is obvious that the social relationship semantic model illustrated in Figure 1 is a unified, flexible, and adaptive semantic model by integrating different social relationship information into a coherent whole, which provides multiple information access views for applications.

Social relationship features make graph-based representations, such as a conceptual graph (CG), a good fit. The power and flexibility of the graph-based representation formats has been rediscovered by modern data management, and the shift towards graphs is motivated by the need to integrate, explore, and exploit resources semantically. Meanwhile, the increasing prosperity of web ontology language (OWL) and linked open data has become one of the major sources of social media information. It will be
convenient and beneficial if we can make full use of the services and techniques provided by the OWL/LOD (linked open data) community. Therefore, both CG- and OWL-based techniques support representation of social relationships formally and semantically. The ontology technology is capable of describing semantics and context. However, this technology lacks the graphical aspect of social relationship representation and makes information visualization more difficult. A conceptual graph can fulfill the graph features of social relationships. Nevertheless, unlike ontology, it is not direct and convenient in representing concepts. To provide a more flexible and adaptive representation, we develop a new way that combines ontology and a conceptual graph, as shown in Figure 1. This combination is the core of the model in Figure 1 and can inherently satisfy the requirement of social relationship representation.

4. Combination of Conceptual Graph and Ontology for Social Relationship Representation

This proposed social relationship representation format combines the essential properties of both CG and ontology. The feasibility of this combination comes from the relationship between CG and ontology. Description logics (DLs) are the logic foundation of ontology, and conceptual graphs are rooted in both frames and semantic networks [60]. They both remedy some critiques on their ancestors. For example, addressing the shortages of distinction between the factual and ontological knowledge as well as increasing precise formal semantics. In conceptual graphs, there is a clear distinction between the vocabulary representing basic ontological knowledge and the sets of graphs representing basic facts. In description logics, a knowledge base is split into the terminological component, namely, the TBox, which can be seen as the ontological part of the knowledge base, and an assertional component (i.e., the ABox), which contains assertions about individuals. Both formalisms are provided with set and first-order logical semantics.

The distinctive DL properties we focus on in this study include the model-theoretical semantics that are compatible with first order logic (FOL) semantics, the constructors that are used for new synthesis concepts, and the OWL ontology resources [61]. The essential properties of CG representing our method are listed as follows:

1. Objects are labelled graphs (mathematically defined with graph-theoretical notions)
2. Reasoning mechanisms are based on graph-theoretical operations, mainly relying on graph homomorphism
3. Efficient reasoning algorithms exist for important specific cases
4. Objects and operations have graphical representations, which make them easily understandable by users (limitation of the semantic gap)
5. The CG model is logically founded, with the inference mechanism being sound and complete with respect to FOL semantics.

4.1. Augmentation of Conceptual Graph Nodes with Ontology Concepts. We first prove the possibility of augmenting CG nodes with ontology concepts. For simplicity, basic conceptual graphs (BGs, the basis of other more complicated CGs) are adopted in the following discussion and proof.

**Definition 1** (vocabulary [46]). A BG vocabulary, or simply a vocabulary, is a triple $(TC, TR, I)$, where:

TC and TR are finite pairwise disjoint sets.

TC, the set of concept types, is partially ordered by a relation $\leq$ and has the greatest element denoted.

TR, the set of relation symbols, is partially ordered by a
relation $\leq$ and is partitioned into subsets $TR_1, ..., TR_k$ of relation symbols of arity 1, ..., $k$. The arity of a relation $r$ is denoted arity ($r$). Any two relations with different arities are not comparable.

$I$ is the set of individual markers, which is disjoint from $TC$ and $TR$. Furthermore, $*$ denotes the generic marker, $M = I \cup \{\ast\}$ denotes the set of markers, and $M$ is ordered as follows: $*$ is greater than any element in $I$ and elements in $I$ are pairwise incomparable.

In some studies, it is assumed that $TC$ has a specific structure, such as a tree, lattice, or semilattice.

**Definition 2** (basic conceptual graph [62]). A basic conceptual graph (BG) defined over a vocabulary $V = (TC, TR, I)$ is a 4-tuple $G = (C, R, E, l)$ satisfying the following conditions:

$(C, R, E)$ is a finite, undirected, and bipartite multigraph called the underlying graph of $G$, denoted graph $(G)$. $C$ is the concept node set, and $R$ is the relation node set (the node set of $G$ is $N = C \cup R$). $E$ is the family of edges.

$l$ is a labelling function of the nodes and edges of graph $(G)$ that satisfies:

1. A concept node $c$ is labelled by a pair (type $(c)$, marker $(c)$), where type $(c)\in TC$ and marker $(c)\in I \cup \{\ast\}$.
2. A relation node $r$ is labelled by $l(r)\in TR$. $l(r)$ is also called the type of $r$ and is denoted by type $(r)$.
3. The degree of a relation node $r$ is equal to the arity of type $(r)$.
4. Edges incident to a relation node $r$ are totally ordered and labelled from 1 to arity (type $(r)$)

There is an important requirement of CG in Definition 1 and Definition 2: there exists a certain order on vocabulary a BG defined over. This means that if we use DL concepts for CG node description, the DL must have similar order relationships between concepts, and the DL concepts should be somehow structured. Fortunately, the order between concepts in $TC$ can be established based on the following refinement operation in DL [63]: starting from a root concept $C_r$, applying a refinement operator $\rho$ to $C_r$, a sequence of concepts as $C_r$’s descendants are generated; then applying operator $\rho$ to each $C_r$’s descendent concept, a new set of concepts will be constructed, and so on.

The definition and proof above make it clear that we can augment CG nodes with DL concepts because of the sequence property that DL concepts have. Adopting the structure of the conceptual graph and the term defined by description logics, a social relationship graph is defined as follows.

**Definition 3** (social relationship graph, SRG). A social relationship graph (SRG) defined over a description logics knowledge base $K$ is a 4-tuple $G = (C, R, E, l)$ defined on a vocabulary triple $(TC, TR, I)$ satisfying the following conditions:

$TC$ is the set including atomic concepts defined in the TBox of $K$ and the complex concepts constructed using atomic concepts and constructors supported by $K$; and $TR$ is the set of roles defined in TBox of $K$.

$I$ is individual sets defined in ABox of $K$.

$(C, R, E)$ is a finite, undirected, and bipartite graph, denoted graph $(G)$. $C$ is the concept node set and $R$ is the relation node set (the node set of $G$ is $N = C \cup R$). $E$ is the family of edges.

$l$ is a labelling function of the nodes and edges of graph $(G)$ that satisfies:

1. A concept node $c$ labelled by a pair (type $(c)$, marker $(c)$), where type $(c)\in TC$ and marker $(c)\in I \cup \{\ast\}$.
2. A relation node $r$ labelled by $l(r)\in TR$. $l(r)$ is also called the type of $r$ and is denoted by type $(r)$.
3. According to the $TR$ defined in $K$, the degree of a relation node $r$ is 2.

Social relationship graphs in Definition 3 has three distinguishable features. First, the concept vocabulary set includes not only atomic concepts but also complex concepts constructed by legal constructors that support the knowledge base. Second, it has a simpler structure by restricting the underlying graph to an undirected and bipartite graph and the relations to binary relations. This simplification makes the operation on social relationship graphs easier and more efficient, especially for applications with the high-performance demand. Third, social relationship graphs can describe document content in addition to the relationships.

4.2. Expressivity of Social Relationship Graphs. This section will discuss the way that the social relationship graph meets the features of social relationship representation, covering the application of social relationship graphs and formal semantics of social graph.

4.2.1. Social Relationship Graph and Applications. Graph properties of social relationship representation not only provide the basic mechanism of browsing social relationship information in a human-readable way but also support graph-based reasoning. Since the conceptual graph is a typical representative of graph-based knowledge representation, the graph structure is an inherent part of Definition 3. Graph algorithms can be easily applied to this structure for data analysis and mining. Graph homomorphism is a totally different way to perform reasoning tasks in knowledge graphs compared with description logic reasoners. Simple conceptual graph properties make it possible to visualize a part of the knowledge graph user concern without losing any semantic information.

Social relationship graphs can play different roles in different applications. Labelling function $l$ of the nodes and edges just meets the application requirements of explicitly showing the graph structure. When the vocabulary triple $(TC, TR, I)$ in Definition 1 varies, the knowledge base and the
knowledge expressivity of the knowledge graph will also change, which can be adapted to different knowledge service requirements, including lightweight knowledge service and inference on a very expressive knowledge base.

4.2.2. Formal Semantics of Social Graph. Formal semantics is the foundation of implementing machine-understandable data processing. The employment of description logics-based ontology in describing terminologies makes the concepts and relations contained in social relationship graphs well defined in a formal way. The context of the social relationship graphs is clear and explicit because of the formal semantics of description logics. Based on Definition 3, a semantic model can be applied to a social relationship graph, which is the fundamentals of implementing graph-based reasoning. First, a model of a vocabulary is defined as the model of a description logic. It consists of a set, the set of entities also called a universe, upon which the concept types, the relation types, and the individuals are interpreted. A concept type is interpreted as a subset of the universe, a relation type is interpreted as a set of tuples of elements of the universe, and an individual is interpreted as an element of the universe. Second, a model of a social relationship graph over a vocabulary \( V \) is defined. It is a model of \( V \) enriched by an interpretation of the concept nodes as elements of the universe. Then, an entailment relation can be defined, i.e., what it means that a social relationship graph \( G \) entails a social relationship graph \( H \), or equivalently, what it means that \( G \) is a consequence of \( H \). This canonical model of a social relationship graph \( G \) plays an important role in reasoning and computing on \( G \).

4.3. Reasoning on the Social Relationship Graphs. Based on the canonical model of the social relationship graph, the reasoning mechanism can be implemented in a combinational way. Since the detection subsumption is the most fundamental reasoning task and the foundation of query answering in DLs and CG, this study analyzes and designs a practical subsumption detection method for social relationship graphs.

In CG, let \( G \) and \( H \) be two BGs over the same vocabulary. Intuitively, \( G \) subsumes \( H \) (noted \( G \sqsubseteq H \)) if the fact—or the information—represented by \( H \) entails the fact represented by \( G \), or in other words, if all information contained in \( G \) is also contained in \( H \). “\( G \) subsumes \( H \)” is equivalent to “\( H \) is subsumed by \( G \),” denoted \( H \sqsubseteq G \). BGs are structured by a subsumption relation defined by the homomorphism notion between BGs, as well as by elementary specialization and generalization operations.

A homomorphism from a BG \( G \) to a BG \( H \) is a mapping from the nodes of \( G \) to the nodes of \( H \), which preserves the relationships between entities of \( G \) and may specialize the labels of entities and relationships. In graph-theoretical terms, it is a labelled graph homomorphism that will be precisely described later. For the moment, it is only necessary to know that a generalization/specialization relation (or subsumption) over BGs can be defined with this notion: \( G \) is more general than \( H \) (or \( G \) subsumes \( H \) or \( H \) is more specific than \( G \)), if there is a homomorphism from \( G \) to \( H \). The soundness and completeness of homomorphism with respect to entailment relation and FOL deduction is proven.

With BGs and subsumption, we can build a basic query answering mechanism. Let us consider a KB (knowledge base) \( B \) composed of a set of BGs, representing some assertions about a modelled world. A query made to this base is itself a BG, say \( Q \). Elements answering \( Q \) are intuitively defined as the elements in \( B \) that entail \( Q \), or equivalently, elements that are specializations of \( Q \), or also, elements that are subsumed by \( Q \).

5. Social Relationship Representation Implementation

There are two main tasks in implementing SRG as follows: describing structure and determining vocabularies. Both the logic structure and graph topology of social relationship graphs can be implemented by conceptual graphs; and the vocabularies of social relationships depend on the content and information described. Adopting widely used open standard vocabularies about social media, such as FOAF and Dublin core metadata initiative (DCMI), makes the social relationship graph more general and applicable to describe social network data. For terminologies in specific domains that standard vocabularies were not covered, users can define new vocabularies as concepts in SRG. Either single social network information or integrated social information involving more than one application can be represented by a social relationship graph.

5.1. Social Relationship Graph for a Single Social Network

Taking stackoverflow.com as an example to show that the social relationship graph represents a question on the Web (see Figure 2), rectangles and ovals represent the concepts and relationships, respectively. The vocabularies shown in Figure 2 are either open standard or user-defined, which are fairly self-explained and human readable. There are eight different relationships, which contain the question’s title, topics, tags, persons who asked the questions, persons who answered the questions, persons who commented on the questions, and persons who commented on the answers.

Figure 3 is another example of social relationship representation on zhihu.com. As presented in the figure, although the resource described is not a question, a technical document and the document node have a more complex structure. Moreover, the inner structure of the node is a social relationship graph. Such a feature enhances the representation capability of the social relationship graph. Thus, the information can be described in detail, such as the content of the document. By addressing concept nodes with inner social relationship structures, we can organize the graph and view it as a complex concept.

5.2. Integrating Related Social Relationship Graph

Based on the social relationship graph structure, we can integrate the separated social network data into a larger scale graph.
according to user demand. An easy way to integrate is to merge isolated social relationship graphs into a single graph. During the merging procedure, different graphs are connected by a common node. For example, in Figure 4, resource question q1 and document d2 with the same topic “Java Programming” are integrated (other information related to q1 and d2 is omitted). Figure 5 shows a different way of integration. More specifically, two graphs are connected by “knows” relationship between people since this relationship is used to connect two separate graphs. Additional information (such as relation definition and pairs of individuals with this relation defined; in this example, the relation is “knows” and the fact about the relation is p3 knows p4) is required to perform this kind of graph integration.

5.3. Social Relationship Graph Views. The social relationship graphs can be viewed from different perspectives, including node-centric or relationship-centric. Basically, the information a social relationship graph presents is summarized as concept information and relationship information, as shown in Figure 2 and Figure 3.
in Figure 6. People, resources, and tags/topics are three types of concept users who are most concerned about the social network, and the relationship between them are the key information accordingly. Only binary relationships are discussed in this work, and the two elements of a relationship could be different concepts, such as people and resources, people and tags, or the same concepts, such as friend/follower/coworker relationships between people and comment/referencerelationships between resources.

Based on the information model shown in Figure 6, we established different social relationship graph views to organize relationships between different concepts. A social relationship graph view is a subgraph structure of a social relationship graph, which aims to capture the relationships between two concept types. However, it is noted that a relationship search task would not have to traverse the whole social relationship graph. Indeed, it only focuses on the views of the contained information that users need.

By analysing the search tasks on the social network, three kinds of social relationship graph views are defined in the study, including people-centric view, resource-centric view, and people-resource centric view.

5.3.1. People-Centric View. The people-centric view focuses on the relationship between people. On social networks, people are usually connected by activities or resources. It notes that the artefacts of an activity (such as text, image, and video) are considered web resources. Hence, a resource index is initially constructed by listing relevant people who connect with a resource. Moreover, to support quick queries of relationships between people, a people-centric view can enable the server to retrieve queried relationships of people in real time. Therefore, this study creates a people-indexed database by analysing resources and inferring a social graph based upon people’s mutual activities across social resources.

It is complex to extract the relationships between people from all relevant resources. This study mainly focuses on two dimensions of abstraction for such relationships. First, the people relationships are classified into six categories: covering organization, friend, tag, comment, co-authorship, and co-membership. Second, we distinguish between relationships that are likely to reflect familiarity between two individuals (e.g., tagging each other or having a common manager) and relationships that are likely to reflect similarity between the individuals (e.g., using the same tag or commenting on the same blog entry) (see Table 1).

5.3.2. Resource-Centric View. Having a built people-centric view, this study creates a resource index for listing people who connect with the resource. This index is efficient for finding certain resources. However, for queries about a set of relevant resources, an additional index structure is required to achieve
better performance. Therefore, a resource-to-resource relationship view is constructed to provide direct links to relevant resources. The relevance between resources can be determined by both the resources and the people (see Table 2). The following four types of resource relationships are included: topic, authorship, tagging, and commenting.

5.3.3. People-Resource-Centric View. Having indicated the relationships between people and resources in people-centric and resource-centric views, it is necessary to build a people-resource-centric view to support fast people-resource relationship queries. Four people-resource relationship categories are contained in the study, including create, comment, tag, and share, to summarize people-resource relationships (see Table 3).

5.4. Social Relationship Visualization. Social relationship visualization is a basic requirement for users. Therefore, it is necessary to provide the support mechanism for visualizing the detailed relationship information and explaining the meaning of the relationships. Such a way is helpful to discover new information hidden in the relationships.

5.4.1. Relationship Visualization. The most important focus of relationship discovery is people. In other words, the people-centric view plays a vital role within social relationship graphs. However, in this view, people are not simply represented as a textual list but instead are displayed using social graph visualization. Social graph visualizations have a tendency to be complex, and visualization techniques should attempt to make the information legible. Nodes and links are positioned using an advanced force-directed, stress majorization algorithm to minimize node overlaps and edge crossings [15].

Visualization can highlight a pivotal type of information relevant to relationship discovery, namely, social position. Social position is important because users normally are not familiar with others who match their analytic queries. For
example, by exploring the connection of people, their peers, or known individuals, users can gauge which people are better suited for their relationship tasks. Social position can be used as a barometer by determining whether a matched user might be willing to communicate with the user. Prior work shows that “social software participation” is a significant signal of the likelihood of contact. Finding a matched person with few social connections may be adequate, but finding a well-connected individual might better meet the user’s needs. Social position can be conveyed via social graph visualization (see Figure 7; note that the colour of a node, degree of a node, and thickness of the links between nodes can show meanings of related relationships). Within the figure, the nodes represent the top people who match the user’s topic. The edges represent the types of relationships that connect various people (see the people relationship categories in Table 1). In the visualization interface, the node can also present the user’s features, such as name and image. As there can be multiple categories of relationships connecting two individuals, bands are added for each edge representing each category, producing a “rainbow” when multiple categories are present.

5.4.2. Relationship Explanation. The relationship explanation can help users to better understand the reasons for user connections in the social graph, including users, resources, and tags. As three types of information (three views defined in Section 5.3) are connected in the analytic system, it is possible for users to freely pivot from one data type to another to find the information they need. We organize the three types of information into three separate tabs as evidence of the existence of the relationship. This functionality is powerful under consideration that a list of names is often not enough to be of practical use when results include people the user is unfamiliar with. By providing coordinated explanations that the user can explore, users can become acquainted enough to judge whether the person is a useful result.

6. Application

6.1. Social Relationship Graph Application. Social relationship graphs can be used in many applications where relationships between entities are of great concern. We provide an overview of typical applications of SRGs.

6.1.1. Link Prediction. A natural problem for a graph is to predict if there is a link between two nodes \( v \) and \( u \). Example applications include graph completion, friend recommendation, and finding biological connections among species. For SRG representation, link prediction can use the semantics of relationships to reduce the computing cost by ignoring the node pairs irrelevant in semantics.

6.1.2. Entity Prediction. One of the fundamental problems in graphs is to predict missing entities or relations. It is a classical problem where we want to predict either a missing entity \((?; r; u)\) or \((v; r; ?)\), where \((v; r; u)\) denotes that there exists relationship \( r \) between \( v \) and \( u \). Similar to link prediction, entity prediction can also make use of the entity semantics implied in SRG to determine the type of missing node and reduce the computing scale.

6.1.3. Recommender Systems. The design of recommender systems is an important applied graph problem faced by a myriad of e-commerce companies. As a coarse approximation, we can view a recommender-system problem as one

| Vocabularies | Relevance by document | Relevance by people |
|--------------|----------------------|---------------------|
| Topic        | Documents with the same topic | Documents shared by the same person |
| Authorship   | Documents created by the same author | Documents created by similar authors |
| DTagging     | Documents tagged with same tags | Documents tagged by the same person |
| DCommenting  | Documents commented by the same people | Documents commented by similar people |

| Vocabularies | Relevance                                      |
|--------------|-----------------------------------------------|
| createDoc    | People create a document                       |
| commentDoc   | People comment a document                      |
| tagDoc       | People tag a document                          |
| shareDoc     | People share a document                        |

| Table 2: Document-document relationships. |
|-------------------------------------------|
| Table 3: People-resource relationships. |

**Figure 7:** Social graph visualization of social positions.
of link prediction (or entity prediction) and attempt to recommend to users the items they are likely to autonomously choose. Enhanced by the expressivity of SRG, the recommender system can provide explainable and interpretable recommendation results to both users and business owners.

6.1.4. Node Classification. Node classification is the problem of classifying graph nodes into different classes. An example of a node classification problem is to predict the political affiliation of the users of a social network based on their attributes, connections, and activities. It will be convenient to classify nodes by meanings, if every node in the graph has a specific type or semantics. The nodes belonging to the same type are an inherent classification. The SRG structure representation makes this type-based classification easier to be accomplished.

6.1.5. Question/Query Answering. The way search engines respond to our questions has evolved in the last few years. There are an increasing number of studies focusing on semantic-based query answering because it can provide more relevant results.

The use of SRG in the abovementioned applications consists of three steps: (1) modelling problem domain, including searching or developing ontology for constructing the vocabulary triple \( (TC, TR, I) \) in Definition 1; (2) establishing conceptual graph structure based on \( (TC, TR, I) \), and augmenting relationship with more property ontology; this procedure involves much human work, so it is hard to construct SRG for an application automatically; (3) choosing the proper algorithms to perform tasks. Taking query answering as an example, we will discuss the detailed query answering based on SRG in the following subsections.

6.2. Relationship Discovery Algorithms. Relationship discovery aims to find the relationships that satisfy user’s query request in datasets. Query answering is a special case of relationship discovery task. This section describes the efficiency of the relationship discovery algorithm. The core task of the SRG-based query answering is to find and discover relationships asked by the user’s query. The implementation of social relationship discovery largely relies on the description form of the social relationship graph. Two description formats are used for the relationship discovery task in this work. They are cogitant social relationship graphs (see Figure 8(a)) and RDF social relationship graphs (see Figure 8(b)). Because we adopt the cogitant library to implement the SRG, the SRG complies with the cogitant conceptual graph format.

By cogitant, we can perform the relationship query through graph homomorphism. To overcome the computing complexity, the query condition is required and represented in a tree structure [42]. Algorithm 1 shows a graph homomorphism-based query answer procedure.

For social relationship graphs described by ontology language, such as RDF/OWL, the relationship discovery task can be performed under the support of reasoners. As an RDF query language, SPARQL works well for the RDF-compatible data format. Thus, it is simple and easy to perform the social relationship query tasks using SPARQL when the relationship information is presented in the RDF format. For more complicated queries, more capable query answering mechanism will be used to perform query processing. This study adopts a reasoner-based query-answering mechanism for exploiting the capacity of deep reasoning and the knowledge expressivity of RDF/OWL.

Usually, the homomorphism on basic conceptual graph is NP-complete. Within social information query answering circumstance, it is feasible to solve homomorphism problems in polynomial time by restricting the form of query without affecting the expressivity. Reasoning efficiency on OWL highly depend on the expressivity of OWL. Although reasoning with OWL has a high worst case complexity (\( N^2 \text{EXPTIME} \)), practical reasoning optimism techniques have made its applicability to large semantic datasets. Thus, the algorithms used for relationship discovery is applicable to solve real problems. In the following, a detailed description of how the two algorithms perform will be presented.

6.3. Efficiency of Social Relationship Graph-Based Query Answering. To illustrate the efficacy of the social relationship graphs, an experimental study is conducted by testing relationship discovery tasks with different complexities. LUBM is used to generate a relationship dataset with different scales in the university domain for this experiment. Note that although LUBM is not a dataset that is directly collected from social network data, the concepts and properties defined in LUBM are highly similar to those of social relationship. It is because LUBM has designed 14 standard query tasks based on the dataset. For the algorithm performance evaluation purposes, it is convenient to conduct experiments on these artificial datasets with different scales, and analyse the results intensively. To carry out the experiments, both relationship discovery algorithms above are used to complete the query task. Then, the performance of the algorithms is verified by the LUBM benchmark.

To test the performance of the social relationship graph-based relationship discovery, it is important to develop a benchmark dataset with graph properties supported by well-defined ontology and standard query tasks. According to the philosophy of ontology engineering, reusability is a main objective during the process of developing an ontology. Instead of constructing a brand-new ontology for evaluating the proposed method in this study, we found an ontology that can meet requirement of experiment for better efficiency. Therefore, the LUBM ontology and the queries designed in the LUBM benchmark are employed to construct a social relationship graph \( G \) and queries on \( G \). The LUBM ontology involves different types of queries that are similar to search of different relationships (see the descriptions in Section 4.3). The instances of concepts and relations generated by the LUBM data generator UBA are the entities knowledge graph \( G \) covered. For each LUBM
query, both the description logics reasoning mechanism and conceptual graph homomorphism algorithm are used to execute the query and compare the performance of the two methods. This demands both OWL-based data representation and storage and conceptual graph format of the LUBM test data. There are two parts that convert the OWL data representation to a conceptual graph knowledge base, including converting the instances generated in graph format and expressing the query in a conceptual graph. A knowledge base is composed of a large number of relative conceptual graphs, and each graph corresponds to an instance generated by UBA. A query about a knowledge graph is represented in the conceptual graph format, and graph homomorphism is used to implement query processing.

The pellet and cognitant libraries are employed to perform the queries. The storage of LUBM data has two versions. One is the .owl file for pellet. Another is the .bcs/.bcg file for cognitant library. Both types of storage are plain disk files, and

![Figure 8: Description formats for relationship discovery task. (a) Cognito social relationship graph. (b) RDF social relationship graph.](image-url)

**Algorithm 1:** Query answering on social relationship graph by graph homomorphism algorithm.

| Input: a social relationship graph G, query graph C_qi |
|------------------------------------------------------|
| **Output:** answers, the result of the C_qi |
| (1) If G is a large knowledge base |
| (2) viewG = createGraphView (G); |
| (3) if C_qi is not tree structure |
| (4) treeCq_qi = treeStrucConvert (C_qi); |
| (5) Choose a suitable view sg in viewG |
| (6) answers = homomorphism (treeCq_qi, sg); |
| (7) return answers. |

**Algorithm 2:** Query answering on social relationship graphs by DL reasoner.

| Input: a social relationship graph G, query concept C_qi |
|--------------------------------------------------------|
| **Output:** answers, the result of the C_qi |
| (1) If G is a large knowledge base |
| (2) viewG = createGraphView (G); |
| (3) Choose a suitable view sg in viewG |
| (4) answers = instanceChecking (C_qi, sg); |
| (5) return answers; |
Table 4: Queries used in the test.

| Query Number | Query                                                                                           |
|--------------|-------------------------------------------------------------------------------------------------|
| 1            | Answer (X) ← takes Course (X, http://www.Department0.University0.edu/GraduateCourse0)          |
| 2            | Answer (X, Y) ← GraduateStudent (X) ∧ University ∧ Department (Z) ∧ memberOf (X, Z) ∧ subOrganizationOf (Z, Y) ∧ undergraduateDegreeFrom (X, Y) |
| 3            | Answer (X) ← Publication (X) ∧ publicationAuthor (X, ttp://http://www.Department0.University0.edu/AssistantProfessor0) |
| 4            | Answer (X, Y1, Y2, Y3) ← Professor (X) ∧ telephone (X, Y3) ∧ worksFor (X, X, http://www.Department0.University0.edu) ∧ Name (X, Y1) ∧ emailAddress (X, Y2) |
| 5            | Answer (X) ← Person (X) ∧ memberOf (X, http://www.Department0.University0.edu)                  |
| 6            | Answer (X) ← Student (X)                                                                          |
| 7            | Answer (X, Y) ← Student (X) ∧ Course (Y) ∧ takesCourse (X, Y) ∧ teacherOf (http://www.Department0.University0.edu/AssociateProfessor0, Y) |
| 8            | Answer (X, Y, Z) ← Student (X) ∧ Department (Y) ∧ memberOf (X, Y) ∧ subOrganizationOf (Y, http://www.University0.edu) ∧ emailAddress (X, Z) |
| 9            | Answer (X, Y, Z) ← Student (X) ∧ Faculty (Y) ∧ Course (Z) ∧ advisor (X, Y) ∧ takesCourse (X, Z) ∧ teacherOf (Y, Z) |
| 10           | Answer (X) ← Student (X) ∧ takesCourse (X, http://www.Department0.University0.edu/GraduateCourse0) |
| 11           | Answer (X) ← ResearchGroup (X) ∧ subOrganizationOf (X, http://www.University0.edu)              |
| 12           | Answer (X, Y) ← Chair (X) ∧ Department (Y) ∧ worksFor (X, Y) ∧ subOrganizationOf (Y, http://www.University0.edu) |
| 13           | Answer (X) ← Person (X) ∧ hasAlumnus (http://www.University0.edu, X)                             |
| 14           | Answer (X) ← UndergraduateStudent (X)                                                             |

Figure 9: Continued.
no special data storage systems are required. Table 4 lists the 14 LUBM queries that are tested in this study. The graph representations for the 14 queries are shown in Figures 9(a)–9(n). The 14 queries can be divided into three types. First, the query does not need the relationship information to perform query answering (e.g., Query 6 and Query 14). Only single individual is considered in the query condition and the query results. Second, the query needs the relationship information to express query condition with only single individual who uses the query results (e.g., Query 1, Query 3, Query 5, Query 10, Query 11, and Query 13). Last, the query needs the relationship information to express query condition with only single individual who uses the query results (e.g., Query 1, Query 3, Query 5, Query 10, Query 11, and Query 13). These three types of queries cover the social relationship understanding and the discovery task discussed in Sections 5 and 6.2. Thus, it is reasonable to adopt the LUBM queries to evaluate the query answering performance of the SRG-based method. For Query 2 and Query 9, the direct graph forms are cyclic. This study divided the cyclic structure into a few smaller acyclic graphs to keep the complexity of graph homomorphism-based reasoning tractable for applications.

Table 5 shows the time cost of the 14 queries under description logics reasoner pellet and graph homomorphism. The results show that both DL inference and graph homomorphism can complete query answering on the LUBM knowledge base. The performance of DL inference-based query answering is highly related to factors such as the query type, DL reasoner, and dataset scale. Some queries consume almost the same time on very different data scales, such as Query 1, Query 3, Query 4, and Query 5; however, the time cost of some queries grows exponentially with the dataset scale, such as Query 8 and Query 9. The reasoner used (pellet in this work) also determines the query results because different reasoners (such as pellet, Fact++, and Racer) work on different mechanisms and has different capabilities. In terms of graph homomorphism-based inference, it is obvious that the only explicit factor that affects the performance is the data scale, the time cost of using the cogitant library grows almost linearly, and each query has a similar trend.

Figure 10 visualizes the time cost of the 14 LUBM Queries on pellet and graph homomorphism reasoning mechanism. The scale of the Y axis is logarithmic to improve the presentation. It seems that DL-based reasoning is highly related to queries, but graph-based reasoning performance is proportional to data volume. Since query complexity and dataset volume are two of the most important factors influencing query answering performance, it is necessary to choose an appropriate reasoning mechanism according to the features of queries and datasets. A comparison between DL inference-based query answering and graph-based query answering is conducted from different focuses (see Table 6). As shown in the table, it is not easy to draw a simple conclusion on which query answering mechanism is better.

It is obvious that the time cost of graph homomorphism is lower than that of the DL reasoner when the data volume is not extremely large. However, this does not mean that graph homomorphism-based query processing is superior. Although the time cost of DL reasoners is higher, it is still acceptable for some knowledge-based applications. For
applications supported with well-defined background knowledge and high-performance reasoners, reasoning-based query processing will be more convenient. Generally, we prefer DL reasoning-based query processing for simple queries, and graph homomorphism-based reasoning is fit for very large datasets.

We mentioned that a distinctive feature of the conceptual graph representation is the division of a

| Query | Pellet | Cogitant |
|-------|--------|----------|
| 1     | 1.114295 | 0.39     |
|       | 1.124605 | 0.83     |
|       | 1.129796 | 1.26     |
|       | 1.1324   | 1.79     |
|       | 1.137627286 | 3.58    |
| 2     | 2.187762 | 2.24     |
|       | 3.235937 | 4.31     |
|       | 4.168694 | 6.73     |
|       | 4.570882 | 10.91    |
|       | 5.754399373 | 23.54   |
| 3     | 1.000691 | 0.49     |
|       | 1.002305 | 0.86     |
|       | 1.004847 | 1.47     |
|       | 1.006236 | 2.76     |
|       | 1.009485302 | 5.44    |
| 4     | 1.011114 | 0.39     |
|       | 1.011579 | 1.12     |
|       | 1.012512 | 1.69     |
|       | 1.013678 | 3.08     |
|       | 1.0167168 | 6.31    |
| 5     | 1.076465 | 0.31     |
|       | 1.081434 | 0.68     |
|       | 1.08893 | 0.92     |
|       | 1.099006 | 1.52     |
|       | 1.119437883 | 3.16    |
| 6     | 1.129796 | 0.16     |
|       | 1.318257 | 0.39     |
|       | 2.137962 | 0.52     |
|       | 3.235937 | 0.93     |
|       | 7.413102413 | 1.88    |
| 7     | 1.011579 | 0.44     |
|       | 1.08893 | 0.93     |
|       | 1.124605 | 1.48     |
|       | 1.158777 | 2.37     |
|       | 1.244514612 | 4.84    |
| 8     | 3.090295 | 0.21     |
|       | 1621.81 | 2.79     |
|       | 58884.37 | 2.92     |
|       | 870963.6 | 8.09     |
|       | 1071519305 | 21.53   |
| 9     | 2.884032 | 0.39     |
|       | 478.6301 | 1.0067   |
|       | 3890.451 | 1.011347 |
|       | 10.21    | 16.77    |
|       | 34.24    | 3.41     |
| 10    | 1.005773 | 0.27     |
|       | 7.09     | 0.59     |
|       | 10.011347 | 0.92    |
|       | 10.030386 | 1.66    |
|       | 1.076465214 | 3.17    |
| 11    | 1.056818 | 0.19     |
|       | 1.073989 | 0.41     |
|       | 1.093956 | 0.63     |
|       | 1.127197 | 1.22     |
|       | 1.150800389 | 2.58    |
| 12    | 1.086426 | 0.09     |
|       | 1.09144 | 0.41     |
|       | 1.106624 | 0.63     |
|       | 1.129796 | 1.22     |
|       | 1.193988104 | 23.76   |
| 13    | 1.001844 | 0.03     |
|       | 1.001913 | 0.71     |
|       | 1.002098 | 1.18     |
|       | 1.002144 | 1.56     |
|       | 1.002305238 | 3.31    |
| 14    | 1.099006 | 0.11     |
|       | 1.101539 | 0.26     |
|       | 1.106624 | 0.41     |
|       | 1.129796 | 0.81     |
|       | 1.180320636 | 1.72    |
knowledge base without losing any semantic relationships. This makes us apply, divide, and conquer strategies to deal with large datasets. For example, LUBM (20, 0) contains more than 200,000 instances. If we divide this dataset into 10 small parts, execute the queries on each small part, and merge the query results on each part after finishing the query processing, the time cost will be dramatically reduced. However, it is difficult to divide the dataset to perform DL inference-based query processing because any division may lead to important missing relationships. Under this circumstance, graph homomorphism is a suitable reasoning mechanism for tasks with a large volume of data, as we have discussed above.

Our experiment on LUBM has showed the possibility and efficiency of query answering mechanism proposed for the large social data. However, note that as an artificial dataset designed for ontology and semantic query research, this work is an intensive pilot study that employs the proposed social relationship model to real social network applications.

7. Conclusion and Further Work

Representing social relationship formally and semantically is the fundamental of exploiting social data in applications where social relationship information play an important role. The variety of application purposes makes a unified and explicit relationship representation difficult. The graph based relationship model can benefit from the graph theory and high-performance graph algorithms for processing social relationship efficiently, and ontology-based semantic model of relationship makes operations on relationship understandable and reasonable. Although social relationships have been studied and exploited much and both graph and ontology knowledge representations does not constitute a new technology, there is still a lack of an explicit definition and a formal representation of social relationships extracted from large-scale social data.

The contribution of this work lies in a comprehensive analysis of social relationship representation and its application. This work proposed social relationship representation by combining the conceptual graphs and domain ontologies. This representation can take advantage of the knowledge representation features of both conceptual graphs and description logics and provide two different ways to perform query processing. Both reasoner-based description logics and graph homomorphism mechanisms are available for query processing on this representation, which makes social relationship data adaptive to applications with different scales and requirements.

In this work, we consider a semantic-level social relationship representation form and its corresponding query processing mechanism. Many other important problems remain undecided and open for extended social relationship applications, such as social relationship storage, graph visualization, and systematic knowledge derivation. The performance of social relationship graph techniques needs to be further increased to satisfy more extensive and large applications. A more general social relationship dataset with versatile relationship types needs to be constructed for better experimental settings. Moreover, representing real social relationship data with the SRG model needs semantic annotation work, which is hard to be implemented automatically in this study. Therefore, more work needs to be done to construct a real social relationship dataset for a more intensive study. All of these are research problems, which will be focused in future work.

Data Availability
The data used to support the findings of this study are available upon request to the corresponding author.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Acknowledgments
This study was supported by research grants funded by the National Natural Science Foundation of China (Grant nos. 61771297 and 61907029) and Natural Science Foundation of Shaanxi, China (Grant no. 2020JM-307).

References

[1] A. G. Sutcliffe, J. F. Binder, and R. I. M.R. I. Dunbar, “Activity in social media and intimacy in social relationships,” Computers in Human Behavior, vol. 85, pp. 227–235, 2018.
[2] A. C. A. C Orben and R. I. M.R. I. M Dunbar, “Social media and relationship development: the effect of valence and intimacy of posts,” Computers in Human Behavior, vol. 73, pp. 489–498, 2017.
[3] S. Jaffali, S. Jamoussi, N. Khelifi, and A. B. Hamadou, Survey on Social Networks Data Analysis, International
[36] S. Mallek, I. Boukhris, Z. Elouedi, and E. Lefevre-lefevre, "Evidential link prediction in social networks based on structural and social information," Journal of Computational Science, vol. 30, pp. 98–107, 2019.

[37] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: aA survey," Knowledge-Based Systems, vol. 151, pp. 78–94, 2018.

[38] Y.-S.Y. Lu and J.-L.J. Huang, "GLR: a graph-based latent representation model for successive POI recommendation," Future Generation Computer Systems, vol. 102, pp. 230–244, 2020.

[39] Y. Qiao, X. Luo, C. Li, H. Tian, and J. Ma, "Heterogeneous graph-based joint representation learning for users and POIs in location-based social network," Information Processing & Management, vol. 57, no. 2, Article ID 102151, 2020.

[40] A. Salim, S. S. Shiju, and S. Sumitra, "Design of multi-view graph embedding using multiple kernel learning," Engineering Applications of Artificial Intelligence, vol. 90, Article ID 103534, 2018.

[41] R. Agrawal, M.Md. Arquam, and A. Singh, "Community detection in networks using graph embedding," Procedia Computer Science, vol. 173, pp. 372–381, 2020.

[42] S. Fu, G. Wang, S. Xia, and L. Liu, "Deep multi-granularity graph embedding for user identity linkage across social networks," Knowledge-Based Systems, vol. 193, Article ID 105301, 2020.

[43] F. Li, Z. Zhu, X. Zhang, J. Cheng, and Y. Zhao, "Diffusion induced graph representation learning," Neurocomputing, vol. 360, pp. 220–229, 2020.

[44] L. Zhang, Q. Yan, and L. Zhang, "A text analytics framework for understanding the relationships among host self-description, trust perception and purchase behavior on airbnb," Decision Support Systems, vol. 133, Article ID 113288, 2020.

[45] K. Y. Lee and H. Choi, "Predictors of electronic word-of-mouth behavior on social networking sites in the United States and Korea: cultural and social relationship variables," Computers in Human Behavior, vol. 94, pp. 9–18, 2019.

[46] S. Ahmadian, N. Joorabloo, M. Jalili, Y. Ren, M. Meghdadi, and M. Afsharchi, "A social recommender system based on reliable implicit relationships," Knowledge-Based Systems, vol. 192, Article ID 105371, 2020.

[47] V. S. Chooralil and E. Dr. E. Gopinathan, "A semantic web query optimization using resource description framework," Procedia Computer Science, vol. 70, pp. 723–732, 2015.

[48] C. Song, T. Ge, Y. Ge, H. Zhang, and X. Yuan, "Labeled graph sketches: kKeeping up with real-time graph streams," Information Sciences, vol. 503, pp. 469–492, 2019.

[49] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger, "Linked data quality of DBpedia, Freebase, OpenCyc, wWikipedia, and YAGO," Semantic Web, vol. 9, no. 1, pp. 77–129, 2017.

[50] L. Zou and M. T. TsuOtsu, "Graph-based rdf data management," Data Science and Engineering, vol. 2, no. 1, pp. 56–70, 2017.

[51] M. Wyllot, M. Hauswirth, P. Cadré-Mauroux, and S. Sakr, "RDF data storage and query processing schemes: a surveyRDF dData sStorage and qQuery pProcessing sSchemes," ACM Computing Surveys, vol. 51, no. 4, pp. 1–36, 2018.

[52] G.G Marzano, "Using resource description framework (RDF) for description and modeling place identity," Procedia Computer Science, vol. 77, pp. 135–140, 2015.

[53] E. Amador-Dominguez, E. Serrano, D. Manrique, P. Hohenecker, and T. Lukasiewicz, "An ontology-based deep learning approach for triple classification with out-of-

knowledge-base entities," Information Sciences, vol. 564, pp. 85–102, 2021.

[54] C. Chen, F. Cai, X. Hu, J. Zheng, Y. Ling, and H. Chen, "An entity-graph based reasoning method for fact verification," Information Processing & Management, vol. 58, no. 3, Article ID 102472, 2021.

[55] W. Liu, J. Liu, M. Wu et al., "Representation learning over multiple knowledge graphs for knowledge graphs alignment," Neurocomputing, vol. 320, pp. 12–24, 2018.

[56] A. Fa, B. Aa, C. Mi, D. Ran, E. Mhs, and F. Ksk, "Traffic accident detection and condition analysis based on social networking data," Accident Analysis & Prevention, vol. 151, Article ID 105973, 2021.

[57] F. Ali, S. El-Sappagh, S. M. R. Islam et al., "An intelligent healthcare monitoring framework using wearable sensors and social networking data," Future Generation Computer Systems, vol. 114, pp. 23–43, 2021.

[58] F.F. Ali, S.S. El-Sappagh, and D.D. Kwak, "Fuzzy ontology and LSTM-based text mining: a transportation network monitoring system for assisting travel," Sensors (Basel, Switzerland), vol. 19, no. 2, pp. 1–23, 2019.

[59] F. Ali, D. Kwak, P. Khan et al., "Transportation sentiment analysis using word embedding and ontology-based topic modellingTransportation sentiment analysis using word embedding and ontology-based topic modeling," Knowledge-Based Systems, vol. 174, pp. 27–42, 2019.

[60] S. Steffen and S. Rudi, Handbook on Ontologies, Springer, Berlin, Heidelberg, Germany, 2009.

[61] S. Rudolph, "Foundations of description logics, reasoning Web, semantic technologies for the Web of data," Lecture Notes in Computer Science, vol. 6848, pp. 23–27, 2011.

[62] M. Chein and M. L. Mugnier, Crossroads: Knowledge representation: Computational Ffoundations of Conceptual Graphs, pp. 1–428, Springer, Berlin, Heidelberg, Germany, 2009.

[63] M. K. Sarker and P. Hitzler, "Efficient concept induction for description logics," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3036–3043, 2019.