Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A multi-agent framework for mining semantic relations from Linked Data*

Hua-jun CHEN†1, Tong YU†‡1, Qing-zhao ZHENG[1], Pei-qin GU[1], Yu ZHANG[2]
([1]*School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)
([2]*College of Information, Zhejiang Sci-Tech University, Hangzhou 310018, China*)
†E-mail: {huajunsir, ytcs}@zju.edu.cn

Received Aug. 13, 2011; Revision accepted Feb. 13, 2012; Crosschecked Feb. 27, 2012

Abstract: Linked data is a decentralized space of interlinked Resource Description Framework (RDF) graphs that are published, accessed, and manipulated by a multitude of Web agents. Here, we present a multi-agent framework for mining hypothetical semantic relations from linked data, in which the discovery, management, and validation of relations can be carried out independently by different agents. These agents collaborate in relation mining by publishing and exchanging inter-dependent knowledge elements, e.g., hypotheses, evidence, and proofs, giving rise to an evidentiary network that connects and ranks diverse knowledge elements. Simulation results show that the framework is scalable in a multi-agent environment. Real-world applications show that the framework is suitable for interdisciplinary and collaborative relation discovery tasks in social domains.

Key words: Semantic Web, Linked open data, Semantic association discovery
doi:10.1631/jzus.C1101010 **Document code:** A **CLC number:** TP311

1 Introduction

Berners-Lee *et al.* (2001) envisioned the Semantic Web (SW) as a Web of data that is meaningful and understandable to any computer. The Web of data, when fully realized, will enable us to share structured data (e.g., spreadsheets and databases) as easily as we share documents, photos, and videos today. Conceptually speaking, the Web of data can be viewed as a graph layer that emerges on top of the current Web (Ayers, 2008). According to Berners-Lee *et al.* (2008), the Web of data has two faces: (1) the 'Graph of Things' (also called the Giant Global Graph, GGG), which encapsulates the semantic relations under investigation, with nodes representing concepts and edges representing relations that are annotated with evidence, and (2) the 'Web of documents', which contains a set of interlinked documents that serve as evidence. The Semantic Web will ignite a revolution of intelligent agents, which operate directly on the 'Graph of Things' and collaborate with each other to solve complex problems and accomplish intelligent tasks (Berners-Lee *et al.*, 2001; Hendler, 2001; 2007). This technical trend will lead to the emergence of intelligent applications that take advantage of the Web of data to augment the underlying Web system's functionalities, such as information retrieval and knowledge sharing (Mukherjea, 2005).

The concept of the Semantic Web is best manifested in the prosperous movement of Linked Data, which was initiated by Berners-Lee (2006). According to Heath and Bizer (2011), Linked Data "provides a publishing paradigm in which not only documents, but also data, can be a first class citizen of the Web, thereby enabling the extension of the Web with

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (Nos. 61070156 and 61100183) and the Natural Science Foundation of Zhejiang Province, China (No. Y1110477)

a global data space based on open standards—the Web of data". Inspired by Linked Data principles, the Semantic Web community has published a large set of datasets, covering a broad range of domains such as life sciences, healthcare, social networking, and e-commerce (Bizer *et al.*, 2009). With the rapid growth of datasets on the Web, how to distill knowledge and insights from this wealth of data becomes an important problem.

In this paper, we focus on a particular knowledge discovery problem called relation discovery (also called link discovery, link predication, relation mining, etc.), which typically means to find interesting relations (expressed as meaningful paths, subgraphs, patterns, etc.) from large-scale datasets (Tarjan, 1981; Deerwester *et al.*, 1990; de Raedt *et al.*, 2007). The Web of data, which is essentially a graphical data model, has provided excellent vehicle for the representation, mapping, and analysis of complex relations. With this background, semantic association discovery (SAD) is proposed to infer implicit or latent relations between arbitrary resources based on patterns discovered from the Web of data (Anyanwu and Sheth, 2003; Aleman-Meza, 2005; Anyanwu, 2007; Anyanwu *et al.*, 2007).

Here, we illustrate the usefulness of SAD through a motivating story of mining social networks (Mika, 2005; Aleman-Meza *et al.*, 2006). As shown in Fig. 1, the news contents from Web pages are important sources for mining social relations between public figures. The extracted relations, however, typically lack accurate semantic labels. Linked data can be used to discover direct or indirect evidence that annotates the extracted relations. For example, the frequent co-occurrences of 'Obama' and 'Michelle Robinson' can be annotated with the triple ⟨Obama, spouse, Michelle Robinson⟩ that can be queried from linked data.

This story illustrates a genre of 'connect-the-dots' applications, in which knowledge analysts typically use analytical tools to gather a set of interlinked intelligence resources to discern hidden and important relations, which often involves cross-domain knowledge integration and collaboration. A 'connect-the-dots' application requires that the 'Graph of Things' should be extracted from the 'Web of documents', which is then navigable and editable by multiple parties, and also capable of answering domain-specific complex problems.

To support the above genre of applications, semantic relations should be derived from intelligence resources and aggregated into a graph while connecting to their evidence for justification and validation. Accordingly, we propose a hypothesis-driven

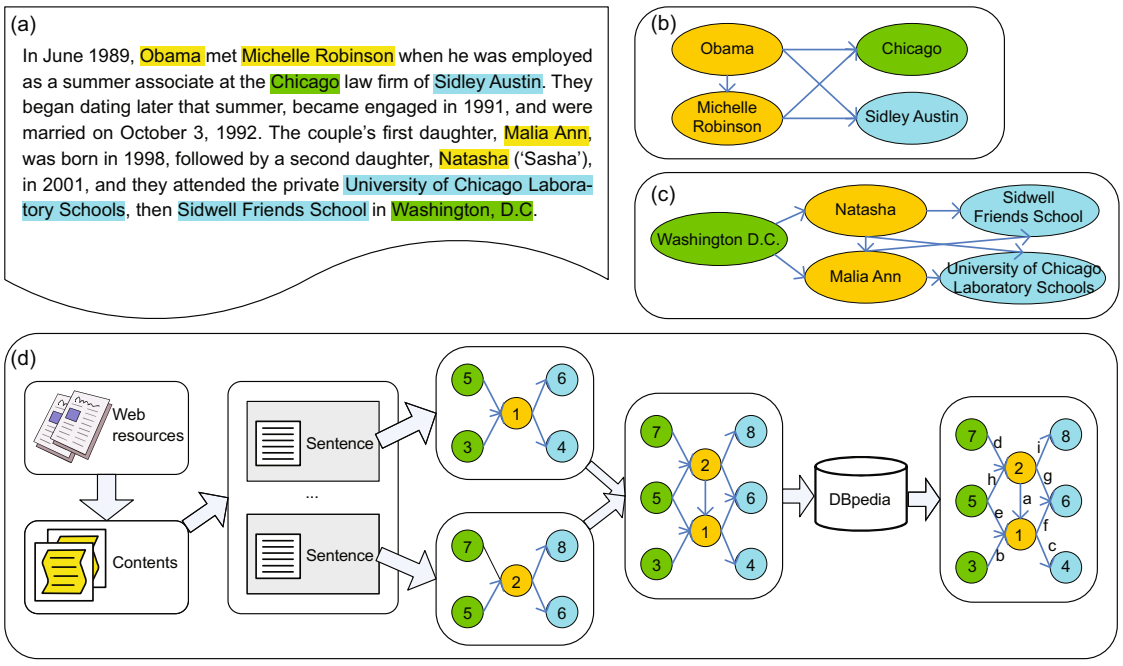


Fig. 1 An example of mining semantic relations from documents with the help of Linked Data. (a) A snippet of text from which entities and relations are to be extracted; (b) The graph extracted from the first sentence; (c) The graph extracted from the third sentence; (d) The mining process

framework for a multitude of agents to collaborate in discovering and validating latent semantic relations. A semantic relation is published as a hypothesis, which is an identified and attributable RDF statement whose truthfulness depends on the existence of certain relevant evidence. On the other hand, an evidence is open or uncertain if its own reliability depends on the validity of some open hypotheses. The mutual dependency between hypotheses and (open) evidence gives rise to an evidentiary network.

In the scenario of multi-agent collaboration, an agent can publish a predicted yet unproved relation as a hypothesis H, send H to a selective set of neighbors who might be interested in it and devoted to solving it, and then periodically search for the evidence of H that other agents have published. Each agent, upon solving a hypothesis, can make its own decision on which (partial) evidence to provide and which (derivative) hypotheses to propose, to influence the direction of mining and contribute to the final solution to the hypothesis. This framework allows an evidentiary network to emerge through the communication of hypotheses and evidence by a multitude of agents.

2 Related work

The activities of Semantic Web are led by the World Wide Web Consortium (W3C), which focuses on promoting the standardization and development of the World Wide Web (Berners-Lee *et al.*, 2006). According to the W3C Semantic Web Activity (http://www.w3.org/2001/sw/), "the Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries".

The Semantic Web builds on the Resource Description Framework (RDF) (W3C RDF Working Group, 2004). RDF extends the use of the uniform resource identifier (URI) (Berners-Lee *et al.*, 1998) to the identification of any significant object, including concrete things and abstract concepts (Berners-Lee *et al.*, 2008). An RDF document, as a set of subject-predicate-object triples, is also a graph with each node corresponding to a URI or a literal, and an edge corresponding to a triple. For example, a triple $\langle s, p, o\rangle$ is represented as $s \xrightarrow{p} o$. Therefore, a set of RDF triples is called an RDF graph. Also, a named graph is an RDF graph that is associated with a URI (Carroll *et al.*, 2005), and a set of named graphs forms a dataset that can be published on the Web. In addition, the major Semantic Web technologies include the Simple Knowledge Organization System (SKOS) (Semantic Web Deployment Working Group, 2009), Web Ontology Language (OWL) (W3C OWL Working Group, 2009), and SPARQL query language (W3C SPARQL Working Group, 2008).

The basic idea of Linked Data is to apply Semantic Web technologies to the task of sharing structured data on global scale (Heath and Bizer, 2011). Berners-Lee (2006) first coined the term 'Linked Data', and proposed the following Linked Data principles:

1. Use URIs as names for things.
2. Use HTTP URIs, so that people can look up those names.
3. When someone looks up a URI, provide useful information, by using the standards (RDF, SPARQL).
4. Include links to other URIs, so that they can discover more things.

In 2007, W3C initiated the Linking Open Data (LOD) community project to realize the Semantic Web vision by publishing various open datasets according to Linked Data principles. As of September 2011, the resulting Web of data, also known as the LOD cloud, contains 295 datasets, 31 634 213 770 RDF triples, and 503 998 829 RDF links (Bizer, 2006). The resulting Web of data allows humans and Web agents to look up, navigate, and edit RDF graphs using the HTTP protocol, and provides an elegant solution to connect data from different sources and domains. This data utility has been applied in various domains, such as genetics (Feigenbaum *et al.*, 2007), drug discovery (Stephens *et al.*, 2006), neuroscience (Ruttenberg *et al.*, 2009), and social networking (Mika, 2005; Aleman-Meza *et al.*, 2006). A rich set of technologies and tools, such as semantic browsers and semantic search engines, have been created to consume linked data for these applications.

Here, we focus on discussing the major works in SAD, which means mining relations from the Web of data. Anyanwu and Sheth (2003) defined a set of ρ-queries for SAD, which were further developed and extended by Aleman-Meza (2005) and Mukherjea *et al.* (2005). Sabou *et al.* (2008) presented a

demo of SCARLET, a technique for discovering relations between two concepts by harvesting the Semantic Web, i.e., automatically finding and exploring multiple and heterogeneous online ontologies. Volz *et al.* (2009) presented the Silk Linking Framework, a toolkit for discovering and maintaining data links between Web data sources. Anyanwu (2007) integrated SAD with SPARQL query language, to support a more powerful analysis of linked data.

These works share the characteristic of representing relations in terms of a domain ontology, which enables agents to reason with the relations (Anyanwu and Sheth, 2003; Aleman-Meza, 2005; Mukherjea *et al.*, 2005; Anyanwu, 2007). The related works show that the Semantic Web can improve the accuracy, application-relevance, and actionability of relation mining, through the following means: (1) formally representing the domain knowledge, the data semantics, and the problem-solving context (Mukherjea, 2005); (2) facilitating the automated extraction of semantic relations from texts (Mukherjea *et al.*, 2005); (3) facilitating data integration to map a comprehensive network of relations (Aleman-Meza *et al.*, 2006).

In terms of practicality, SAD has been applied in a series of use cases, e.g., social network analysis (Aleman-Meza *et al.*, 2006), national security (Anyanwu, 2007), patent retrieval (Mukherjea *et al.*, 2005), and biomedicine (Mukherjea, 2005). However, the effectiveness of the proposed techniques in real-world applications needs to be further investigated (Anyanwu and Sheth, 2003; Mukherjea, 2005).

In addition, these works are limited by the fact that the mining function is typically conducted by one agent in a centralized manner (Anyanwu and Sheth, 2003; Mukherjea *et al.*, 2005; Aleman-Meza *et al.*, 2006; Anyanwu, 2007). Designing a scalable infrastructure for relation mining in distributed environments is an open and important problem (Mukherjea *et al.*, 2005; Anyanwu, 2007). Therefore, our major concern is to propose decentralized and collaborative mechanisms that are scalable to multiple distributed sources.

3 Formulation and methods

3.1 Formulation

As shown in Fig. 2a, we define some fundamental concepts in knowledge publication and exchange. Here, every published knowledge element is defined as a knowledge resource that must have at least one author; therefore, we reuse the dc:Creator (from Dublin Core Metadata, http://dublincore.org/documents/dces/) and foaf:Agent (from Friend of a Friend (FOAF), http://xmlns. com/foaf/spec/) to define the Class KnowledgeResource as the subClass of 'dc:Creator some foaf:Agent'. A knowledge resource can be annotated with numeric or literal values delivering its semantic information. In particular, a knowledge resource can be annotated with numerical weights, such as 'confidence', 'utility', and 'importance'.

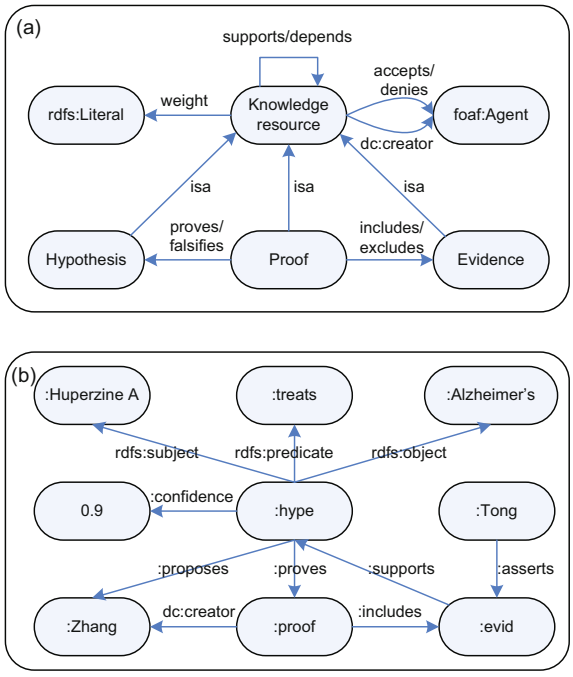
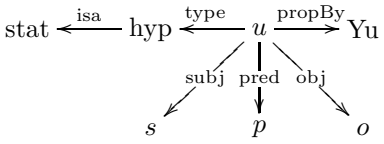


Fig. 2 An overview of the ontology used in this study. (a) The major classes and properties used for knowledge exchange; (b) The graph related to the hypothesis "Huperzine A seems to treat Alzheimer's disease"

We define a hypothesis as a published RDF statement whose truthfulness is under investigation. A hypothesis h is a quintuple $\langle u, s, p, o, a\rangle$, in which u is the URI for h, s, p, and o are the subject, predicate, and object of h, respectively, and a is a set of agents that 'propose' h. A hypothesis $h = \langle u, s, p, o, a\rangle$ means that the URI u identifies a problem regarding whether $\langle s, o\rangle$ belongs to the binary relation denoted by p, and is represented as $h\langle u\rangle$ or $h: s \xrightarrow{?p} o$ (the notion '?' is used to denote uncertainty). A hypothesis is a knowledge resource that is proposed by

(isProposedBy [propBy]) at least one agent; therefore, we define the class Hypothesis [hyp] as the subClass of 'KnowledgeResource and isProposedBy some foaf:Agent'. A hypothesis is represented in two ways:

1. RDF statement. A HypotheticalStatement is a hypothesis that is represented as an rdf:Statement. As illustrated in the following graph, we reuse the reification vocabulary in RDFS, such as rdf:Statement[stat], and the properties rdf:subject[subj], rdf:predicate[pred], and rdf:object[obj] to represent an instance of HypotheticalStatement:



2. Named (singleton) graph. A HypotheticalGraph contains a set of hypotheses proposed by the same authors and is represented as a named graph. A HypotheticalGraph is a singleton graph if it has exactly one hypothesis.

While an 'evidence' in general can be any resource (documents, images, videos, or even persons) that facilitates an agent to achieve a belief (or denial) of a hypothesis, we here focus on explicit and formal evidence that is expressed in Semantic Web languages. An evidence e is a triple $\langle u, p, a\rangle$, in which u is e's URI, p is a graph that specifies e's pattern, and a is a set of agents that 'assert' e. Such an evidence is in itself a pattern seen as a subgraph extracted from an agent's knowledge base, and thus evidence discovery can be seen as a subgraph extraction problem.

In addition to the publication of hypotheses and evidence, agents also need to make assertions about their mutual dependency: (1) If an evidence e is asserted to support a hypothesis h, then the statement $\langle h, \text{depends}, e\rangle$ can be asserted; (2) If an evidence e' depends on a hypothesis h', then the statement $\langle e', \text{depends}, h'\rangle$ can be asserted. Therefore, an evidence e may be related with two kinds of hypotheses: (1) the hypotheses that e is intended to support; (2) the hypotheses that e depends on (if e is uncertain). We propose an evidentiary graph to wrap up these dependency relationships. For two mutually exclusive sets H and E that contain hypotheses and evidence respectively, an evidentiary graph is a directed bipartite graph $\langle H, E, D\rangle$ where H and E are two sets of vertices, and D is a set of edges corresponding to the mutual dependency relationships among members of H and members of E, so that for $h \in H, e \in E$,

$$\langle h, \text{depends}, e\rangle \quad \Leftrightarrow \quad h \xrightarrow{\text{dep}} e \ \in D,$$

$$\langle e, \text{depends}, h\rangle \quad \Leftrightarrow \quad e \xrightarrow{\text{dep}} h \ \in D.$$

In essence, it is the authors' intent that determines whether a knowledge resource is a hypothesis or an evidence. An agent proposes a hypothesis and asserts an evidence. In addition, an agent accepts a KnowledgeResource if the agent believes it to be true. It accepts resources with the following recursive process:

1. Selectively accept an evidence that does not depend on other hypotheses.

2. Selectively accept a hypothesis that is supported by accepted evidence.

3. Selectively accept an evidence that depends on only accepted hypotheses.

We define a proof as an evidentiary graph in which every knowledge resource is accepted. We say that a proof proves the hypotheses and includes the evidence within its evidentiary graph.

We present an example of our formulation in Fig. 2b. The hypothesis ':hype', proposed by ':Zhang', states that the drug ':Huperzine A' seems to 'treats' the disease ":Alzheimer's", with a confidence of 0.9. To prove the hypothesis, the evidence ':evid' is asserted by ':Tong'. The hypothesis ':hype' and the evidence ':evid' are connected by 'supports' and 'depends' relations. The ':evid' itself may contain open hypotheses which set goals for further investigation. Finally, ':Zhang' creates a proof ':proof' that includes the ':evid' to prove the ':hype'.

3.2 Semantic relation mining

We propose a recursive problem division strategy, in which one agent attempts to solve a published hypothesis with evidence that might depend on other hypotheses, which in themselves may be proved or be under investigation. In this strategy, an agent executes the following process recursively:

1. Select a new hypothesis to solve from a pool of published hypotheses.

2. Extract a subgraph from its factual graph that can support the hypothesis.

3. Formalize the (possible) gap between the pattern (as what is proved) and the hypothesis (as what needs to be proved) into a set of hypotheses.

4. Publish the pattern as an evidence and possibly the newly-derived hypotheses.

We first use an example in Fig. 3 to illustrate this process. Suppose the client proposes a hypothesis $H_1 = 1 \xrightarrow{?p} 2$, and an engine E and three agents collaborate in producing the evidence E for the hypothesis. The engine notifies the hypothesis to agent A, who then discovers an evidence $E_1 = 1 \xrightarrow{f} 3 \xrightarrow{?p} 5 \xrightarrow{g} 2$, which is partial in that it depends on a hypothesis $H_2 = 3 \xrightarrow{?p} 5$. Agent A publishes the evidence and its dependency $H_1 \xleftarrow{\text{supports}} E_1 \xrightarrow{\text{depends}} H_2$. Agent B then provides an evidence $E_2 = 3 \xrightarrow{h} 4 \xrightarrow{?p} 5$ for H_2, which depends on $H_3 = 4 \xrightarrow{?p} 5$. Agent C provides a complete evidence $E_3 = 4 \xrightarrow{h} 6 \xrightarrow{a} 5$ for H_3. Since the resources are distributed among agents, a crawler is needed to recursively crawl the available evidence and index them in the directory. Through the collaboration between E, A, B, C: $E \xrightarrow{H_1} A \xrightarrow{H_2} B \xrightarrow{H_3} C$, the proof generator finds a proof for H_1: $H_1 \xrightarrow{\text{depends}} E_1 \xrightarrow{\text{depends}} H_2 \xrightarrow{\text{depends}} E_2 \xrightarrow{\text{depends}} H_3 \xrightarrow{\text{depends}} E_3$, and an accepted evidence

$$E = \begin{array}{ccccc} 1 & \xrightarrow{f} & 3 & \xrightarrow{h} & 4 \\ \downarrow{\scriptstyle ?p} & & & & \downarrow{\scriptstyle h} \\ 2 & \xleftarrow{g} & 5 & \xleftarrow{a} & 6 \end{array}$$

is returned to the client.

Given a hypothesis $a \xrightarrow{?p} b$, the generic strategy for an agent is:

1. Return available triples $\{ a \xrightarrow{p_i} b \}_{i=1}^{n}$ as direct evidence.

2. Propose a path $a \dashrightarrow b \equiv a \xrightarrow{?p_1} r_1 \xrightarrow{?p_2} r_2 \dashrightarrow r_n \xrightarrow{?p_{n+1}} b$.

3. For each sub-problem $r \xrightarrow{?p} s \in P$, replace it with an available triple $r \xrightarrow{p} s$ if possible.

4. Return P as a new evidence.

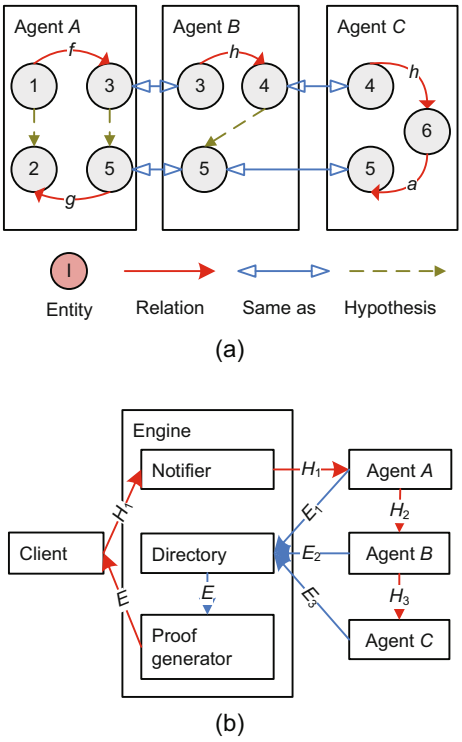


Fig. 3 An overview of the system architecture and the knowledge flow. (a) The architecture of the proposed system which contains a set of agents (called miners) for evidence discovery and hypothesis generation and an engine for registration and notification; (b) The knowledge flow between the client, engine, and agents

5. Publish the newly-generated hypotheses within P.

6. Go to step 2 if more paths exist.

In this process, the basic task of probabilistic decision-making (PDM) for agents to perform is to substitute the hypothesis $a \xrightarrow{?p} b$ with a primary evidence $a \xrightarrow{p} a'$ and a new hypothesis $a' \xrightarrow{?p} b$. Given a set of candidate primary evidence with a as the subject, $\{ a \xrightarrow{p_i} a_i \}_{i=1}^{n}$, the decision of choosing $a \xrightarrow{p} a'$ over others is based on the expected complexity of solving $a' \xrightarrow{?p} b$.

We will move on to implement the engine and the miner, and also investigate how alternative local collaboration and evidence discovery strategies could affect the global results.

3.2.1 Engine

The engine coordinates the collaboration through the agent-capacity registration, the notifi-

cation of knowledge resources, and through proof generation. The directory maintains a global schema for a domain, and each agent can register their local schema describing their hypothesis-answering capacity with the directory. Therefore, the engine can provide a notification of the knowledge resources to the relevant agents who might be able to handle them.

Routing table: The possibility of an agent A reaching resource o is measured by $P = \max_i \text{Similarity}(a_i \in A, o)$ ($a_i \in A$ means A hosts a_i as a resource). In particular, if $o \in A$, then $P = 1$. For a set of agents $\{A_i\}_{i=1}^n$ and a set of resources $\{R_j\}_{j=1}^m$, the matrix $M[n][m] = (P_{ij})$ is called the probabilistic routing table.

Notifier: The notifier notifies a hypothesis $h\langle u\rangle = s \xrightarrow{?p} o$ to the agent A that is chosen from a set of candidates by the following criteria: (1) A has not be notified of h; (2) h belongs to A's local schema; (3) s belongs to A's resource list; (4) A has the maximum possibility of reaching o (by consulting the routing table) among the candidates that satisfy criteria 1, 2, and 3. The candidates for global notification include all agents, and the candidates for local notification include neighbors of h's creator.

Proof generator: As specified in Algorithm 1, the proof generator is an intelligent agent that works on the evidentiary graph in search of an optimized set of proofs for a hypothesis or hypothetical graph specified by users. It includes a checker and a crawler:

1. The checker, a procedure triggered by the generation of evidence e with no depending hypothesis, recursively checks if the hypotheses and evidence that (indirectly) depend on e are closed. Here, an evidence is closed if (1) it does not depend on any hypothesis, or (2) it depends only on hypotheses that are closed. A hypothesis is closed if it is supported by closed evidence.

2. The crawler, a procedure triggered by the client proposing a hypothesis h, navigates on the global evidentiary graph to generate a subgraph that proves h.

3.2.2 Miner

Evidence discovery is implemented as semantic graph routing (Algorithm 2). Within each iteration of the recursive routing procedure, the router (which hosts a graph FG) is given a hypothesis $h = s \xrightarrow{?h} o$ as the original problem, and a trace

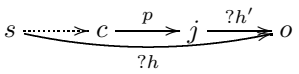
Algorithm 1 Proof generation

```
Checker(e ∈ Evidence)
for all h --dep--> e do
  setTrue(h.closed);
  for all ( e' --dep--> h ) ∧ (isClosed(e') = True) do
    Checker(e');
  end for
end for

isClosed(e ∈ Evidence)
if (e.closed = False) ∧ ((∀ e --dep--> h ) h.closed = True)
then
  setTrue(e.closed);
end if
return e.closed;

Crawler(h^c ∈ ClosedHypothesis)
eg = Graph();
for all e^c, h'^c : h^c --dep--> e^c --dep--> h'^c do
  eg.add( h^c --dep--> e^c --dep--> h'^c );
  eg.add(Crawler(h'^c));
end for
return eg;
```

(which is a path) $t = s \dashrightarrow c$ as the current solution, which is shown as follows:



It first chooses j (as the next 'jumping point') from the neighbors of c and updates the trace t as $s \dashrightarrow j$, and then invokes the next iteration with h and t.

The choice of j from neighbors of c is performed by the Next function, which optimizes the possibility of j associated with o using the following procedure:

1. Choose j if FG entails $j \xrightarrow{p} o$.

2. Choose j which optimizes the estimated value $I \times S$, in which I is j's importance as measured by its degree (the number of triples that contain j as the subject or the object), and S is the similarity between j and o.

The routing procedure terminates within an agent's local boundary under the following conditions: (1) the remaining problem $c \xrightarrow{?h} o$ is an identified hypothesis; (2) $j = o$; (3) j is null (c is the locally-optimized point for h).

In addition, the routing procedure can be relayed to the agent's neighbors through local

Algorithm 2 Semantic graph routing

Router($s, c, o \in$ Resource, $t \in$ Graph)
if $c \xrightarrow{?p} o \in$ HG **then**
  return new Evidence ($s \overset{t}{\cdots\!\rightarrow} c \xrightarrow{?p} o$);
end if
if $c \xrightarrow{p} o \in$ FG **then**
  t.add($c \xrightarrow{p} o$);
  return new Evidence ($s \overset{t}{\cdots\!\rightarrow} o$);
end if
Resource j = Next(c, o, t);
if (j = null) **then**
  new Hypothesis ($c \xrightarrow{?p} o \in$ HG);
  return new Evidence ($s \overset{t}{\cdots\!\rightarrow} c \xrightarrow{?p} o$);
else
  t.add($c \xrightarrow{p} o$);
  return Router (s, j, o, t);
end if

Next($c, o \in$ Resource, $t \in$ Graph)
for all $\{j_i\}_{i=1}^n : (c \to j_i \in \text{FG}) \wedge (j_i \notin t)$ **do**
  $j : I(j, o) = \max_{i=1}^n I(j_i, o)$;
end for
if $(j \neq \text{null}) \wedge (I(j, o) \geqslant I(c, 0))$ **then**
  return j;
else
  return null;
end if

notification. If agent A fails to find such a neighboring agent to relay h, then it will invoke the global notification mechanism for h.

4 Experimental evaluation

We established a multi-agent environment to evaluate the feasibility and scalability of our approach. The experiment setting contains the following components: (1) the schema base (SB) which contains a set of RDF schemas, (2) the resource list (RL) which contains a list of resources, (3) the similarity matrix (SM) which captures the pair-wise similarity between resources, and (4) the factual graph (FG) which is the global graph containing all the triples under investigation.

We then generated an engine and K agents, and assigned agents with data. The scalability parameter K is the number of agents among which the graph is distributed. We used clustering methods to divide the FG into K inter-related communities, which are hosted by K inter-linked agents, and analyzed the performance with respect to K.

We started the process by proposing a set of seeding hypotheses. The original hypothesis generator Agent_{hg} created a set of seeding valid hypotheses: $\{s_i \xrightarrow{?p_i} o_i\}_{i=1}^n$. The engine then performed the global notification mechanism for the seeding hypotheses to initiate the evidence discovery process.

The engine and agents performed evidence discovery through routing and notification until all seeding hypotheses were solved or the time T ran out.

We first present an example (Fig. 4) to explain the process. The factual graph FG contained 21 vertices, 72 edges, and 4 obvious communities with the overlapping node as '0'. The similarity between node i and node j was defined as $1/(D(i,j)+1)$, where $D(i,j)$ is the distance between i and j (with $D(i,j) = 0$ for $i = j$). We assigned each community to one agent, and started the process P with a hypothesis $h_1 = 20 \xrightarrow{?} 10$:

1. The engine notifies h_1 to Agent A_0, the only agent that contains h_1's subject '20'.

2. Agent A_0 routes through the path $20 \to 4 \to 0$ to optimize the similarity between the current node and the target '10'. Agent A_0 publishes the evidence $e_1 = 20 \longrightarrow 0 \xrightarrow{?} 10$ which depends on $h_2 = 0 \xrightarrow{?} 10$.

3. Agent A_2 is notified of h_2 since the possibility P of A_2 reaching '10' ($P = 1$ since $10 \in A_2$) is the highest among the neighbors of A_0.

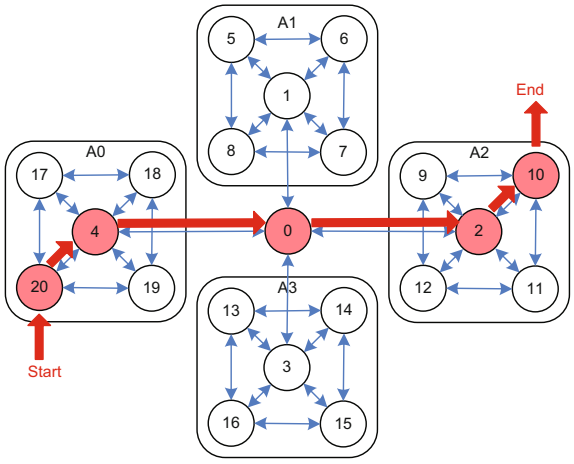


Fig. 4 An example of semantic relation mining by four agents that hold four communities connected by node 0

4. Agent A_2 routes through the path $0 \to 2 \to 10$ based on the same strategy as A_0, and publishes the path as an evidence e_2 for h_2.

5. The publication of e_2 triggers the proof checker to accept h_2, e_1, and h_1 in turn.

Now that h_1 is closed, the discovery process is terminated. Upon the client's request, the crawler can generate an evidentiary graph $h_1 \xrightarrow{\text{dep}} e_1 \xrightarrow{\text{dep}} h_2 \xrightarrow{\text{dep}} e_2$, which entails the evidence $20 \to 4 \to 0 \to 2 \to 10$.

We compared P with an alternative process P', in which the same graph was assigned to only one agent, who found one evidence $20 \to 4 \to 0 \to 2 \to 10$ for $h_1 = 20 \xrightarrow{?} 10$ by using the same strategy as Agent A_0, with the following measurements:

1. Coverage ratio c: The coverage ratio c, as the ratio of the number of traversed statements to the number of all statements in FG, is a major factor for measuring agents' efficiency. Process P has $c = 4/72$, equal to the c of P', which means P is efficient.

2. Average evidence size $\overline{|e|}$: supposing a process generates $\{e_i\}_{i=1}^n$, $\overline{|e|} = \frac{1}{n}\sum_{i=1}^n |e_i|$ is a critical factor for measuring agent's responsiveness. For P, the average evidence size is $(|e_1| + |e_2|)/2 = 2$; for P', the average evidence size is 4—through multi-agent collaboration, the responsiveness of each agent increases.

3. Participation ratio w: the ratio of the number of working miners (who participate in the collaboration) to the number of all miners. For P, the collaboration graph shows $w = 2/4 \equiv 0.5$; for P', $w \equiv 1$.

4. Total computational time t: For the engine E and miners $\{A_i\}_{i=1}^n$, t is the sum of computational time of all agents: $t = T(E) + \sum_{i=1}^n A_i$. The change of factor t over a different number of agents reflects the scalability of the mechanism.

5. Success ratio s and serendipitous discovery ratio d: suppose a process that is given a set S of seeding hypotheses actually solves the set R of hypotheses. The success ratio $s = |S \cap R|/|S|$ and the serendipitous discovery ratio $d = |R - S|/|S|$ are two factors that determine the quality of mining. For P, $s = 1/1$ and $d = 1/1$; for P', $s = 1/1$ and $d \equiv 0$. This means that the multi-agent collaboration has the advantage of discovering some extra knowledge.

4.1 Result analysis

As shown in Fig. 5, we generated a graph with 100 nodes and 861 edges, and performed each iteration with 10 hypotheses. We examined the scalability by increasing the number of agents K from 1, 2, 4, 6, 8, to 10, and the change of

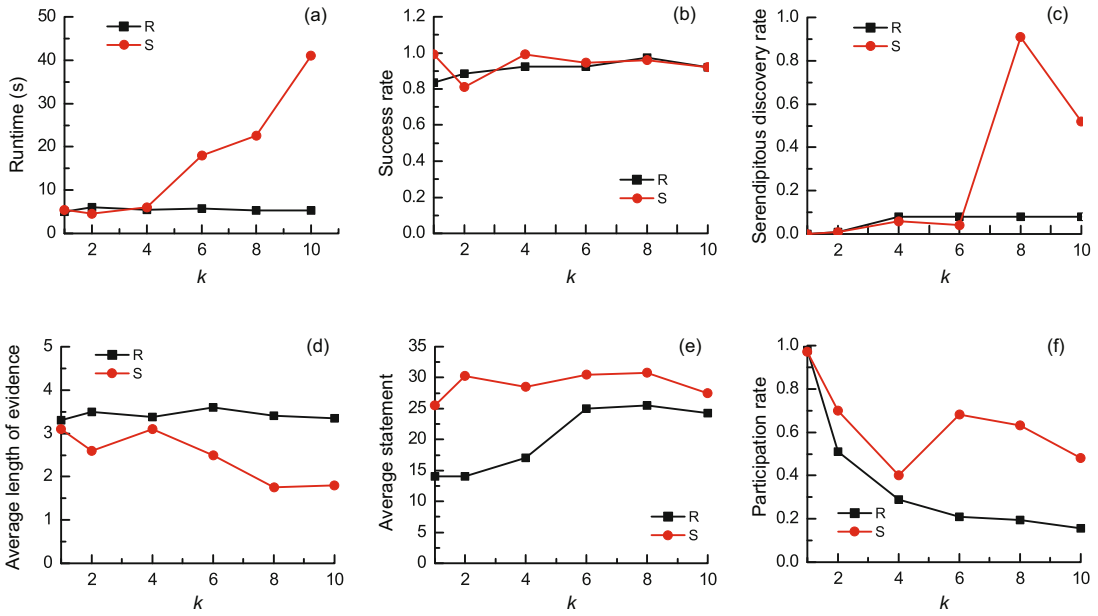


Fig. 5 Simulation results for evaluating the scalability of the proposed mechanism: (a) runtime; (b) success rate; (c) serendipitous discovery rate; (d) average length of evidence; (e) average statement; (f) participation rate. As for the legend, R represents the communication strategy 'a hypothesis is notified to a preferred agent according to the routing table', and S represents 'a hypothesis is notified to a random agent'

major performance factors. For each K, we compared two communication strategies: (1) a hypothesis is notified to a random agent; (2) a hypothesis is notified to a preferred agent according to the routing table.

We performed the process for five iterations. Overall, we proposed 600 hypotheses and finally solved 556 of them with an overall success ratio of 92.7%, which demonstrates that our approach is very effective. The 7.3% unsolved hypotheses indicated the fact that semantic similarity does not necessarily translate to connectivity in a graph. Fig. 5a shows that as the number of agents increased, the total runtime t did not show a large increase in the similarity-based routing strategy, in comparison with the random strategy. Fig. 5b shows that the success rate remained high as the number of agents increased. Fig. 5c highlights that the serendipitous discovery rate (as the rate of the number of derivative hypotheses versus the number of original hypotheses) increased as the number of agents increased. Fig. 5d shows that as the number of agents increased, the average length of evidence decreased, and hypotheses were more frequently solved by the combination of evidence. Fig. 5e demonstrates that as the number of agents increased the covering of statements by agents increased. Finally, Fig. 5f shows that as the number of agents increased, the rate of agents that participated in evidence discovery decreased.

These results show that the proposed routing-table-based approach can scale up to a large number of agents while sustaining the quality and performance of discovery, thus outperforming the random approach. In both approaches, as the number of agents increases, serendipitous discovery can also increase. The results suggest that decentralized decision-making can be effective in discovering hidden semantic relations from distributed information sources.

5 Application in social relationship discovery

In this section, we elaborate on the motivating story in Section 1. In this application, we aim to discover and validate interesting social relations between political figures from textual documents. This application demonstrates that Linked Data can be used to annotate hypothetical relations with evidence, and our approach is suitable particularly for cross-domain knowledge exchange and integration.

5.1 Experiment description

In this experiment, we used mainly the following two datasets: (1) news on the Web—as shown in Fig. 1a, a collection of news describing political figures, news, and events, crawled from news Web sites (Heritrix, http://crawler.archive.org/), was used to crawl these Web resources; (2) DBpedia (http://www.dbpedia.org/), a large-scale Semantic Web dataset extracted from Wikipedia.

In the relation discovery process, we aimed to use linked data to generate semantic annotations for frequent patterns extracted from textual documents. As illustrated in Fig. 1d, we first extracted entities (persons, organizations, locations, etc.) from the content of Web pages. Then, we discerned semantic relations and merged them into graphs (one graph for each document), from which a set of frequently-occurring subgraphs was learned. We then searched linked data for the information that is used to annotate semantics to the edges of the frequent subgraphs.

1. Sentence extraction: Use language processing and regular expression techniques to extract sentences from contents (HTML parser at http://htmlparser.sourceforge.net/ was used to parse pages).

2. Named entity extraction: Recognize different types of named entities (Fig. 1a) (Stanford Named Entity Recognizer at http://nlp.stanford.edu/software/CRF-NER.shtml was used to extract entities from texts).

3. Hypothetical relation discovery in this process takes the following steps:

(1) Semantic relation extraction: For each document, extract hypothetical relations from its content. As illustrated in Figs. 1b and 1c, we asserted the proper hypothetical relations between two entities co-occuring in the same sentence. We focused on triples with an instance of foaf:Person as the subject, i.e., relations between persons (denoted by foaf:Person) and organizations (denoted by foaf:Organization), and between persons and locations (denoted by geo:Feature). For instance, in Fig. 1b, 'Obama' is related with 'Michelle Robinson', 'Sidley Austin', and 'Chicago'. These relations are represented as hypotheses, e.g., ⟨Obama,

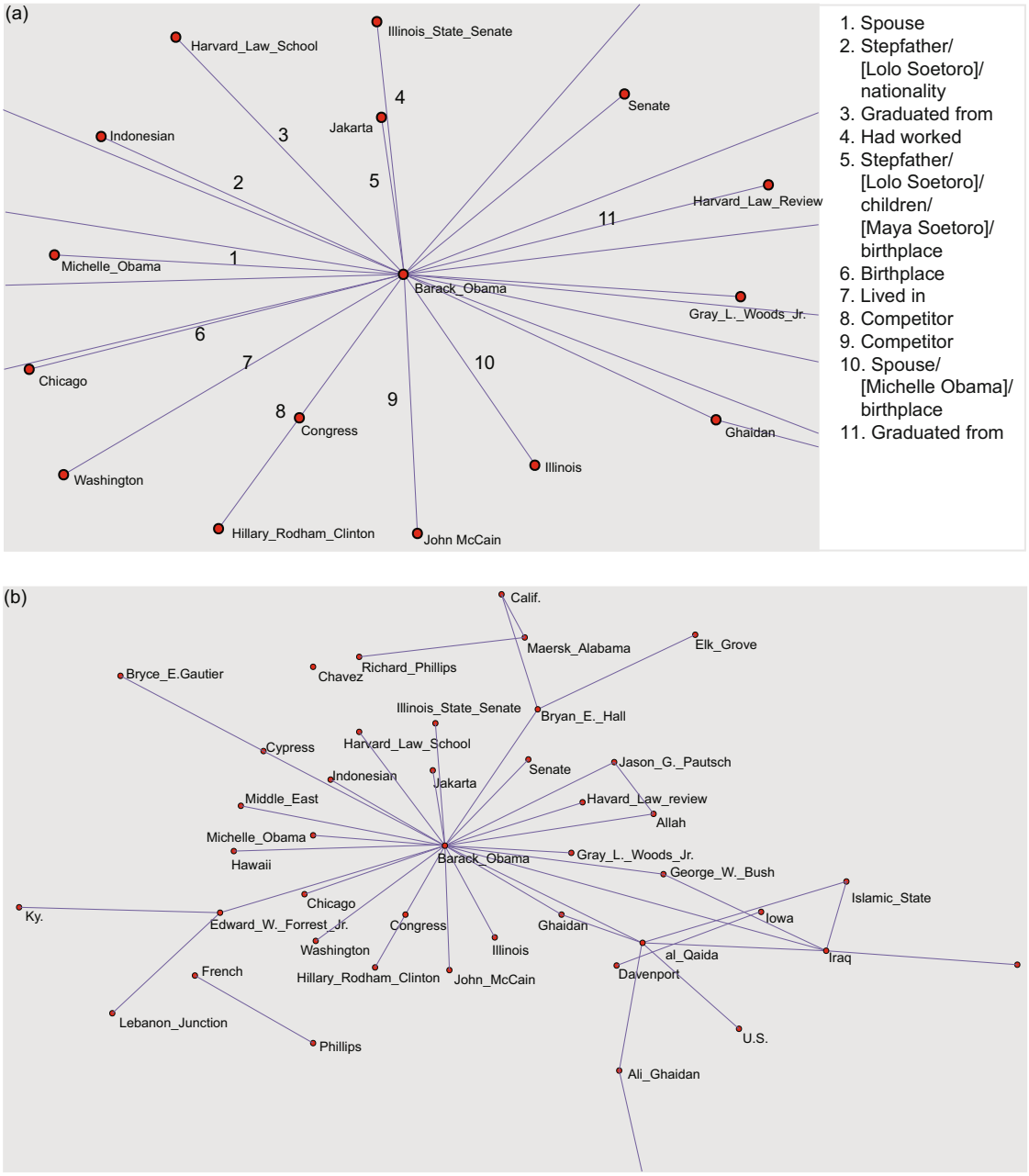


Fig. 6 An example of social relationship discovery. (a) Hypothetical relations of the entity 'Barack Obama' extracted from news; (b) The extension of (a) with the relations discovered from linked data (partially shown)

?foaf:knows, Michelle Obama⟩.

(2) Graph generation: Merge all the sentence graphs into one named graph that represents the content of a textual document.

(3) Pattern recognition: Apply the MARGIN algorithm (Thomas *et al.*, 2006) on the graphs to discover frequently-occurring subgraphs, which reveal the patterns of close or persistent relations between entities.

4. Evidence discovery: Search linked data for evidence that annotates the hypothetical semantic relations within the discovered patterns. We searched two types of evidence for each hypothesis $\langle s, ?p, o\rangle$:

(1) Direct evidence: Search linked data for triple in the form of $\langle s, p', o\rangle$, where p' is a subproperty of p. For instance, we used ⟨Obama, hasSpouse, Michelle Obama⟩ as the evidence for the hypothesis ⟨Obama, ?foaf:knows, Michelle Obama⟩.

(2) Indirect evidence: Search linked data for a subgraph that supports the hypothesis. For instance, for the hypothetical relation of ⟨Obama, ?foaf:knows, Carl Levin⟩ (Carl Levin is a senator from Michigan), we have the evidence {⟨Obama, almaMater, Harvard Law School⟩, ⟨Carl Levin, almaMater, Harvard Law School⟩} (meaning Obama and Carl Levin are schoolmates of Harvard Law School).

5. Pattern visualization: Visualize discovered patterns for human interpretation.

5.2 Results

Fig. 6a displays the results of frequent subgraph mining with a support of 0.04 from our Web resources. Fig. 6b shows that the patterns were annotated with evidence derived from DBpedia. Users can understand the relation between two persons if there are edges between them. Through this approach, we can not only obtain an effective application for text mining, but also enrich linked data with relations learned from textual documents.

6 Conclusions

The Semantic Web, when fully realized, will enable data to be shared and reused in a machine-understandable format, and foster multiple agents to collaborate in knowledge discovery (Mukherjea, 2005). To realize this potential, we have proposed a novel approach for discovering hidden semantic relation from linked data in a human-agent collaborative manner. This approach emphasizes the publication of (partial) knowledge resources that can be interlinked into an evidentiary network. Based on this model, agents can make local decisions as to which hypotheses to propose and work on. Simulation results suggest that certain decentralized decision-making strategies, such as the semantic-based routing tables, can be effective and scalable in a distributed environment. Our approach can be applied in such tasks as social network analysis and medical network analysis.

References

Aleman-Meza, B., 2005. Ranking complex relationships on the Semantic Web. *IEEE Internet Comput.*, **9**(3):37-44. [doi:10.1109/MIC.2005.63]

Aleman-Meza, B., Nagarajan, M., Ramakrishnan, C., Ding, L., Kolari, P., Sheth, A.P., Arpinar, I.B., Joshi, A., Finin, T., 2006. Semantic Analytics on Social Networks: Experiences in Addressing the Problem of Conflict of Interest Detection. Proc. 15th Int. Conf. on World Wide Web, p.407-416. [doi:10.1145/1135777.1135838]

Anyanwu, K., 2007. Supporting Link Analysis Using Advanced Querying Methods on Semantic Web Datasets. PhD Thesis, University of Georgia, Athens, Georgia.

Anyanwu, K., Sheth, A., 2003. ρ-Queries: Enabling Querying for Semantic Associations on the Semantic Web. Proc. 12th Int. Conf. on World Wide Web, p.690-699.

Anyanwu, K., Maduko, A., Sheth, A., 2007. SPARQ2L: Towards Support for Subgraph Extraction Queries in RDF Databases. Proc. 16th Int. Conf. on World Wide Web, p.797-806. [doi:10.1145/1242572.1242680]

Ayers, D., 2008. Graph farming. *IEEE Internet Comput.*, **12**(1):80-83. [doi:10.1109/MIC.2008.13]

Berners-Lee, T., 2006. Linked Data—Design Issues. Available from http://www.w3.org/DesignIssues/LinkedData.html [Accessed on Feb. 19, 2012].

Berners-Lee, T., Fielding, R.T., Masinter, L., 1998. Uniform Resource Identifiers (URI): Generic Syntax. IETF RFP 3986 (Standards Track). Available from www.ietf.org/rfc/rfc3986.txt

Berners-Lee, T., Hendler, J., Lassilia, O., 2001. The Semantic Web. *Sci. Am.*, **284**(5):34-44. [doi:10.1038/scientificamerican0501-34]

Berners-Lee, T., Hall, W., Hendler, J.A., O′Hara, K., Shadbolt, N., Weitzner, D.J., 2006. A framework for Web science. *Found. Trends Web Sci.*, **1**(1):1-130. [doi:10.1561/1800000001]

Berners-Lee, T., Hollenbach, J., Lu, K., Presbrey, J., Prud'ommeaux, E., Schraefel, M., 2008. Tabulator Redux: Browsing and Writing Linked Data. Proc. WWW Workshops: Linked Data on the Web.

Bizer, C., 2006. State of the LOD Cloud. Available from http://www4.wiwiss.fu-berlin.de/lodcloud/state/ [Accessed on Feb. 19, 2012].

Bizer, C., Heath, T., Berners-Lee, T., 2009. Linked data—the story so far. *Int. J. Semant. Web Inf. Syst.*, **5**(3):1-22. [doi:10.4018/jswis.2009081901]

Carroll, J.J., Bizer, C., Hayes, P., Stickler, P., 2005. Named Graphs, Provenance and Trust. Proc. 14th Int. Conf. on World Wide Web, p.613-622. [doi:10.1145/1060745.1060835]

Deerwester, S., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A., 1990. Indexing by latent semantic analysis. *J. Am. Soc. Inform. Sci.*, **41**(6):391-407. [doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9]

de Raedt, L., Kimmig, A., Toivonen, H., 2007. Problog: a Probabilistic Prolog and Its Application in Link Discovery. Proc. 20th Int. Joint Conf. on Artifical Intelligence, p.2468-2473.

Feigenbaum, L., Herman, I., Hongsermeier, T., Neuman, E., Stephens, S., 2007. The Semantic Web in action. *Sci. Am.*, **297**(6):90-97. [doi:10.1038/scientificamerican1207-90]

Heath, T., Bizer, C., 2011. Linked Data: Evolving the Web into a Global Data Space (1st Ed.). *In*: Jantsch, E., Waddington, C. (Eds.), Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool, California, p.1-136. [doi:10.2200/S00334ED1V01Y201102WBE001]

Hendler, J., 2001. Agents and the Semantic Web. *IEEE Intell. Syst.*, **16**(2):30-37. [doi:10.1109/5254.920597]

Hendler, J., 2007. Where are all the intelligent agents? *IEEE Intell. Syst.*, **22**(3):2-3. [doi:10.1109/MIS.2007.62]

Mika, P., 2005. Flink: Semantic Web technology for the extraction and analysis of social networks. *Web Semant.*, **3**(2-3):211-223. [doi:10.1016/j.websem.2005.05.006]

Mukherjea, S., 2005. Information retrieval and knowledge discovery utilising a biomedical Semantic Web. *Brief. Bioinform.*, **6**(3):252-262. [doi:10.1093/bib/6.3.252]

Mukherjea, S., Bamba, B., Kankar, P., 2005. Information retrieval and knowledge discovery utilizing a biomedical patent Semantic Web. *IEEE Trans. Knowl. Data Eng.*, **17**(8):1099-1110. [doi:10.1109/TKDE.2005.130]

Ruttenberg, A., Rees, J.A., Samwald, M., Marshall, M.S., 2009. Life sciences on the Semantic Web: the neurocommons and beyond. *Brief Bioinform.*, **10**(2):193-204. [doi:10.1093/bib/bbp004]

Sabou, M., d'Aquin, M., Motta, E., 2008. SCARLET: SemantiC relAtion discoveRy by harvesting onLinE onTologies. *LNCS*, **5021**:854-858. [doi:10.1007/978-3-540-68234-9_72]

Semantic Web Deployment Working Group, 2009. Simple Knowledge Organization System (SKOS). Available from http://www.w3.org/2001/sw/wiki/SKOS [Accessed on Feb. 19, 2012].

Stephens, S., Morales, A., Quinlan, M., 2006. Applying Semantic Web technologies to drug safety determination. *IEEE Intell. Syst.*, **21**(1):82-86. [doi:10.1109/MIS.2006.2]

Tarjan, R.E., 1981. Fast algorithms for solving path problems. *J. ACM*, **28**(3):594-614. [doi:10.1145/322261.322273]

Thomas, L.T., Valluri, S.R., Karlapalem, K., 2006. Margin: Maximal Frequent Subgraph Mining. Proc. 6th Int. Conf. on Data Mining, p.1097-1101.

Volz, J., Bizer, C., Gaedke, M., Kobilarov, G., 2009. Discovering and Maintaining Links on the Web of Data. Int. Semantic Web Conf., p.1-16.

W3C OWL Working Group, 2009. OWL 2 Web Ontology Language Overview. Available from http://www.w3.org/TR/owl2-overview/ [Accessed on Feb. 19, 2012].

W3C RDF Working Group, 2004. Resource Description Framework (RDF). Available from http://www.w3.org/2001/sw/wiki/RDF [Accessed on Feb. 19, 2012].

W3C SPARQL Working Group, 2008. SPARQL Query Language for RDF. Available from http://www.w3.org/2001/sw/wiki/SPARQL [Accessed on Feb. 19, 2012].

Recommended reading

Aleman-Meza, B., 2005. Ranking complex relationships on the Semantic Web. *IEEE Internet Comput.*, **9**(3):37-44. [doi:10.1109/MIC.2005.63]

Aleman-Meza, B., Nagarajan, M., Ramakrishnan, C., Ding, L., Kolari, P., Sheth, A.P., Arpinar, I.B., Joshi, A., Finin, T., 2006. Semantic Analytics on Social Networks: Experiences in Addressing the Problem of Conflict of Interest Detection. Proc. 15th Int. Conf. on World Wide Web, p.407-416. [doi:10.1145/1135777.1135838]

Anyanwu, K., Maduko, A., Sheth, A., 2007. SPARQ2L: Towards Support for Subgraph Extraction Queries in RDF Databases. Proc. 16th Int. Conf. on World Wide Web, p.797-806. [doi:10.1145/1242572.1242680]

Mukherjea, S., 2005. Information retrieval and knowledge discovery utilising a biomedical Semantic Web. *Brief. Bioinform.*, **6**(3):252-262. [doi:10.1093/bib/6.3.252]

Mukherjea, S., Bamba, B., Kankar, P., 2005. Information retrieval and knowledge discovery utilizing a biomedical patent Semantic Web. *IEEE Trans. Knowl. Data Eng.*, **17**(8):1099-1110. [doi:10.1109/TKDE.2005.130]