RelSim: Relation Similarity Search in Schema-Rich Heterogeneous Information Networks

Chenguang Wang* Yizhou Sun† Yanglei Song‡ Jiawei Han‡ Yangqiu Song§
Lidan Wang¶ Ming Zhang*

Abstract

Recent studies have demonstrated the power of modeling real world data as *heterogeneous information networks (HINs)* consisting of multiple types of entities and relations. Unfortunately, most of such studies (e.g., similarity search) confine discussions on the networks with only a few entity and relationship types, such as DBLP. In the real world, however, the network schema can be rather complex, such as Freebase. In such *HINs with rich schema*, it is often too much burden to ask users to provide explicit guidance in selecting relations for similarity search. In this paper, we study the problem of relation similarity search in schema-rich HINs. Under our problem setting, users are only asked to provide some simple relation instance examples (e.g., ⟨Barack Obama, John Kerry⟩ and ⟨George W. Bush, Condoleezza Rice⟩) as a query, and we automatically detect the *latent semantic relation (LSR)* implied by the query (e.g., "president vs. secretary-of-state"). Such LSR will help to find other similar relation instances (e.g., ⟨Bill Clinton, Madeleine Albright⟩). In order to solve the problem, we first define a new meta-path-based relation similarity measure, *RelSim*, to measure the similarity between relation instances in schema-rich HINs. Then given a query, we propose an optimization model to efficiently learn LSR implied in the query through linear programming, and perform fast relation similarity search using RelSim based on the learned LSR. The experiments on real world datasets derived from Freebase demonstrate the effectiveness and efficiency of our approach.

1 Introduction

Heterogeneous information networks (HINs) have been used recently for modeling real world relationships in many applications [8, 15, 26]. Previous HIN studies [15, 16] are confined to networks with only a few entity and relation types, such as the DBLP network, with four entity types: *Paper*, *Venue*, *Author* and *Term*, and a few relation types connecting the entity types. However, in the real world, HINs can often be with more sophisticated network schemas, i.e., *schema-rich HINs*, containing many more entity types and relation types. For example, the Freebase network[1] contains 1,500+ types of entities, such as *Organization*, *Profession*, *Book*, *Musician*, *Film*, and *Location*, and 35,000+ types of relations among the entity types, such as "is president of" and "is secretary-of-state of" [5].

Many research problems arise with schema-rich HINs. Even the basic functions like similarity search, will need to be re-examined. In HINs with simple schema, explicit guidance for similarity search, can be easily provided by a user to represent her query intent or interest, e.g., *finding similar authors publishing papers at the same venue* can be specified as a composite relation *Author-Paper-Venue-Paper-Author*. However, in schema-rich HINs, it is unrealistic to ask users to provide relations explicitly since there are too many possible meaningful ones to be chosen from a complex network schema, especially when the relations needed are sophisticated.

In this paper, we consider the problem of relation similarity search in schema-rich HINs. In our problem setting, users are asked to just provide a set of simple examples, e.g., ⟨Barack Obama, John Kerry⟩ and ⟨George W. Bush, Condoleezza Rice⟩, as a query, and we automatically detect the *latent semantic relation (LSR)* in the query for the users. With such LSR, other similar relation instances satisfying the same LSR (e.g., "president vs. secretary-of-state," such as ⟨Bill Clinton, Madeleine Albright⟩) are found, and we use the new examples for learning a better LSR iteratively.

As shown in Fig. 1, our goal is to find similar relation instances based on the query $Q = \{\langle$Barack Obama, John Kerry$\rangle, \langle$George W. Bush, Condoleezza Rice$\rangle\}$. However, diverse LSRs are implied by the query. For example, except for LSR "president vs. secretary-of-state," ⟨Barack Obama,

*School of EECS, Peking University, Beijing, China. {wangchenguang, mzhang_cs}@pku.edu.cn

†College of Computer and Information Science, Northeastern University. yzsun@ccs.neu.edu

‡Department of Computer Science, University of Illinois at Urbana-Champaign. {ysong44, hanj}@illinois.edu

§Lane Department of Computer Science and Electrical Engineering, West Virginia University. yangqiu.song@mail.wvu.edu

¶IBM Research. lidan101@gmail.com

[1] http://www.freebase.com/

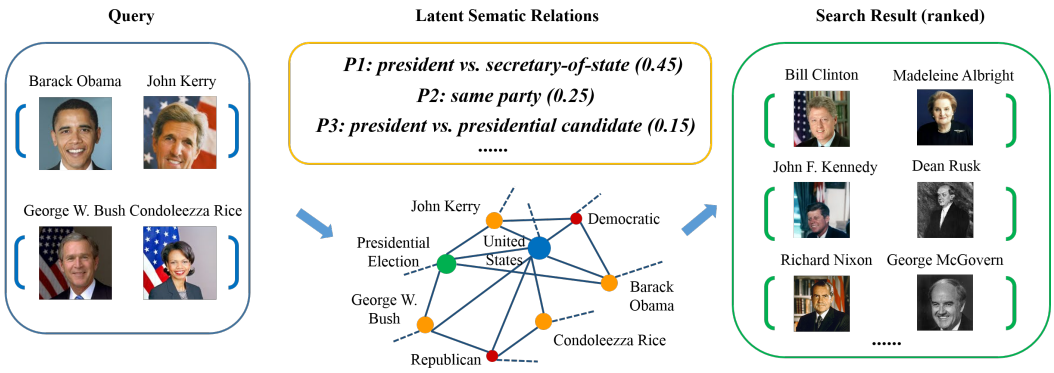

Figure 1: Relation similarity search in schema-rich HIN. Left: a user query; middle: different query-based meta-paths associated with corresponding weights ($\mathcal{P}_1$ = *President* $\xrightarrow{\text{is president of}}$ *Country* $\xleftarrow{\text{is secretary of state of}}$ *Secretary of State*, $\mathcal{P}_2$ = *Politician* $\xrightarrow{\text{is member of}}$ *Party* $\xleftarrow{\text{is member of}}$ *Politician*, $\mathcal{P}_3$ = *President* $\xrightarrow{\text{is president of}}$ *Country* $\xleftarrow{\text{is presidential candidate of}}$ *Presidential Candidate*); right: ranked similar relation instances.

John Kerry⟩ also satisfies LSR "president vs. presidential candidate." Only in the semantic relation of "president vs. secretary-of-state," ⟨Barack Obama, John Kerry⟩ and ⟨Bill Clinton, Madeleine Albright⟩ are similar. The interesting question is *how to measure the similarity between relation instances by distinguishing diverse LSRs?*

Relation similarity has been demonstrated its effectiveness for analogy detection, relation extraction, etc.. However, the existing relation similarity measures [2, 18] do not distinguish the diverse LSRs implied in a relation instance. Besides, there is no trivial way to apply entity similarity measures [12, 16] to measuring relation similarity. For example, "Barack Obama" is similar to "Bill Clinton" (president of United States), and "John Kerry" is similar to "John F. Kennedy" (Democratic). But ⟨Barack Obama, John Kerry⟩ is not similar to ⟨Bill Clinton, John F. Kennedy⟩ according to the LSR "president vs. secretary-of-state."

To tackle the problem, we first define a novel meta-path-based relation similarity measure, *RelSim*, to measure the similarity between two relation instances based on the LSR: two relation instances are more similar when sharing more important (heavily weighted) meta-paths. Then we provide an efficient solution to finding similar relation instances based on RelSim in schema-rich HINs for the user query. Given a query, before learning its LSR, we generate a query-based network schema, e.g., this schema can reduce the number of entity types from 1,500+ in Freebase to five types, which substantially facilitates the subsequent learning process. The most likely LSR is thus efficiently learned based on an optimization model through linear programming, which can best explain the semantic meaning in the query. The experimental results on datasets derived based on Freebase demonstrate the effectiveness and efficiency of our approach.

Our contributions can be highlighted as follows:

- We study *relation similarity search in schema-rich heterogeneous information networks*, a new but very important problem due to its broad applications (e.g., analogy detection).
- We define a novel relation similarity measure, *RelSim*, to compute the similarity between relation instances in HINs.
- We present a framework for relation similarity search in schema-rich HINs, mainly including *latent semantic relation* representation and learning, and an efficient search algorithm.

2 Schema-Rich HINs

In this section, we introduce the schema-rich HIN and some relevant concepts.

DEFINITION 1. *A **heterogeneous information network** (HIN) is a directed graph $G = (V, E)$ with an entity type mapping ϕ: $V \to \mathcal{A}$ and a relation type mapping ψ: $E \to \mathcal{R}$, where V denotes the entity set and E denotes the link set, $\mathcal{A}$ denotes the entity type set and $\mathcal{R}$ denotes the relation type set, and the number of entity types $|\mathcal{A}| > 1$ or the number of relation types $|\mathcal{R}| > 1$.*

The network schema provides a high-level description of a given heterogeneous information network.

DEFINITION 2. *Given a heterogeneous network $G = (V, E)$ with the entity type mapping ϕ: $V \to \mathcal{A}$ and the relation type mapping ψ: $E \to \mathcal{R}$, the **network schema** for network G, denoted as $T_G = (\mathcal{A}, \mathcal{R})$, is a directed graph with nodes as entity types from $\mathcal{A}$ and edges as relation types from $\mathcal{R}$.*

A **schema-rich HIN** is an HIN with the network schema that contains relatively larger number of types of entities and relations, compared to that of schema-simple HIN. Freebase and DBpedia are examples of schema-rich HINs, which contains at least thousands of types of entities and relations. In contrast, DBLP with simple network schema contains four types of entities and several types of relations between these entity types.

Another important concept in heterogeneous information network is *meta-path* [16], proposed to systematically define relations between entities at the schema level.

DEFINITION 3. *A **meta-path** $\mathcal{P}$ is a path defined on the graph of a network schema $T_G = (\mathcal{A}, \mathcal{R})$, and is denoted in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_L} A_{L+1}$, which defines a composite relation $R = R_1 \bullet R_2 \bullet \cdots \bullet R_L$ between types A_1 and A_{L+1}, where $\bullet$ denotes relation composition operator, and L is the length of $\mathcal{P}$.*

For simplicity, we also use type names connected by "$-$" to denote the meta-path when there exist no multiple relations between a pair of types: $\mathcal{P} = (A_1 - A_2 - \cdots - A_{L+1})$. For example, in the Freebase network, the composite relation *two Person co-founded an Organization* can be described as *Person* $\xrightarrow{\text{found}}$ *Organization* $\xrightarrow{\text{found}^{-1}}$ *Person*, or *Person-Organization-Person* for simplicity. We say a path $p = (v_1 - v_2 \dots v_{L+1})$ between v_1 and v_{L+1} in network G follows the meta-path $\mathcal{P}$, if $\forall l, \phi(v_l) = A_l$ and each edge $e_l = \langle v_l, v_{l+1} \rangle$ belongs to each relation type R_l in $\mathcal{P}$. We call these paths as *path instances* of $\mathcal{P}$, denoted as $p \in \mathcal{P}$. R_l^{-1} represents the reverse order of relation R_l.

3 Schema-Rich HIN-Based Relation Similarity Search

We study the *relation similarity search* problem, that is, finding similar relation instances for a user query in schema-rich HINs. Given a small set of relation instances as an example query (e.g., ⟨Larry Page, Sergey Brin⟩ and ⟨Jerry Yang, David Filo⟩), the system will first discover its *latent semantic relation (LSR)* (e.g., "co-founders") and then output similar relation instances (e.g., ⟨Bill Gates, Paul Allen⟩).

In a simple case, a query may imply a simple LSR that can be represented as a single meta-path, such as *Person* $\xrightarrow{\text{found}}$ *Organization* $\xrightarrow{\text{found}^{-1}}$ *Person*. In general, an LSR can be represented as a weighted combination of multiple meta-paths.

DEFINITION 4. *A **latent semantic relation** (LSR) is defined as a weighted combination of meta-paths, denoted as $\{w_m, \mathcal{P}_m\}_{m=1}^M$, where $\mathcal{P}_m$ is m^{th} meta-path and ω_m is the corresponding weight for $\mathcal{P}_m$.*

An advantage of modeling LSR as a weighted combination of meta-paths is augmenting the capability of representing different semantic meanings. For example, given a user query $Q = \{\langle$Larry Page, Sergey Brin$\rangle$, $\langle$Jerry Yang, David Filo$\rangle\}$, we show two meta-paths with weights between the two entities: $\mathcal{P}_1 =$ *Person* $\xrightarrow{\text{found}}$ *Organization* $\xrightarrow{\text{found}^{-1}}$ *Person*, $\mathcal{P}_2 =$ *Person* $\xrightarrow{\text{alma mater}}$ *Education* $\xrightarrow{\text{alma mater}^{-1}}$ *Person*. The corresponding weights are ω_1 and ω_2. If $\omega_1 > \omega_2$, there is a higher possibility that the LSR is "co-founders." If $\omega_1 = \omega_2$, the possibility of the LSR be "co-founders" is equal to be "schoolmates." If $\omega_1 < \omega_2$, there is a higher possibility that the LSR is "schoolmates." Different weighted combinations of meta-paths lead to different semantic meanings.

3.1 RelSim: A Novel Relation Similarity Measure

Although there are some existing relation similarity measures [2, 18], but they do not distinguish the diverse, subtle semantic meanings in the relation instance (i.e., they assume there is only one general relation held in one relation instance). For example, only with semantic meaning "co-founders," ⟨Larry Page, Sergey Brin⟩ and ⟨Bill Gates, Paul Allen⟩ are similar. When the meaning turns to "schoolmates", they are dissimilar. Here, we define a meta-path-based relation similarity measure, *RelSim*, to measure similarity between two relation instances based on the LSR with subtle semantic meaning. The intuition behind RelSim is that if two relation instances share more heavily weighted meta-paths, they tend to be more similar. We formally define RelSim below:

DEFINITION 5. ***RelSim: a meta-path-based relation similarity measure.** Given an LSR, denoted as $\{w_m, \mathcal{P}_m\}_{m=1}^M$, RelSim between two relation instances $r = \langle v^{(1)}, v^{(2)} \rangle$ and $r' = \langle v^{(1)'}, v^{(2)'} \rangle$ is defined as:*

$$RS(r, r') = \frac{2 \times \sum_m \omega_m \min(x_m, x'_m)}{\sum_m \omega_m x_m + \sum_m \omega_m x'_m} \tag{3.1}$$

where x_m is the number of path instances between $v^{(1)}$ and $v^{(2)}$ in relation r following meta-path $\mathcal{P}_m$, and x'_m is the number of path instances between $v^{(1)'}$ and $v^{(2)'}$ in relation r' following meta-path $\mathcal{P}_m$. We use a vector $\mathbf{x} = [x_1, \cdots, x_m, \cdots, x_M]$ to characterize a relation instance r, and a vector $\boldsymbol{\omega} = [\omega_1, \cdots, \omega_m, \cdots, \omega_M]$ to denote the corresponding weights. M is the number of meta-paths.

In schema-rich HINs, the number of path instances between two entities following a specific meta-path is often 1 or 0, denoting whether the two entities satisfy the meta-path-based relation. For example, "Larry Page" and "Sergey Brin" have co-founded one organization. By looking at RelSim defined in Def. 5, we can see that $RS(r, r')$ is defined in terms of two parts: (1) the *semantic overlap* in the numerator, which is the weighted number of overlapped meta-path-based relations of r and r'; and (2) the *semantic broadness* in the denominator, which is the weighted number of total meta-path-based relations satisfied by r and r'. Note that, if the number of path instances for a meta-path is larger than

1, i.e., $x_m > 1$, we treat the two entities have satisfied the relation x_m times. We can see that the larger number of overlapped meta-path-based relations shared by the r and r', the more similar the two relation instances are, which is further normalized by the semantic broadness of r and r'.

RelSim satisfies several nice properties as indicated from properties (1) to (3). The proof is similar to the proof of Theorem 1 in [16].

- (1) Range: $\forall r, r', 0 \leq sim(r, r') \leq 1$.
- (2) Symmetric: $sim(r, r') = sim(r', r)$.
- (3) Self-maximum: $sim(r, r) = 1$.

3.2 Problem Definition Given the above definitions, we now formally define the relation similarity search problem as follows.

Since we are aiming to find other relation instances similar to the ones stated in a query, we first define the RelSim between query and a relation instance as the average similarity between each relation instance in the query and the relation instance:

DEFINITION 6. *Given a user query $Q = \{r_k = \langle v_k^{(1)}, v_k^{(2)} \rangle\}, k = 1, \cdots, K$, and a relation instance r', the RelSim between Q and r' is calculated as $RS(Q, r') = \sum_k RS(r_k, r')/K$.*

Then our relation similarity search problem is to find all the top relation instances that are similar to query Q. In schema-rich HINs, it is not trivial to identify the LSR given a user query, as the possible semantic meaning implied in the query is diverse.

4 Latent Semantic Relation Learning

In this section, we introduce our LSR learning method.

4.1 Meta-Path Candidates Generation Before learning to find the most likely LSR implied in the query, based on the following observations:

1. It is commonsense that the real semantic meaning in a query is specific, i.e., the meaning should be represented with limited number of meta-paths that focus on relevant types of entities and relations;
2. It is time consuming and impractical to automatically generate meta-paths by enumerating all the possible meta-paths between entities in large-scale networks,

we need to find a small number of query-based meta-path candidates $\mathbf{P}$ that could express the real meaning. We therefore construct a query-based network schema based on the user query Q, through only keeping the types of entities and relations relevant to the query.

DEFINITION 7. *Query-based network schema. A query-based network schema is a sub-network schema of a schema-rich HIN. Given a schema-rich HIN $G = (V, E)$, a user query Q, and the radius of schema D, D is the maximum length of hops that an entity $v \in Q$ can arrive on the graph of schema, then the query-based network schema contains types of entities in the Q and within D-hop to the entities (denoted as V_u), and types of relations $\langle v^{(1)}, v^{(2)} \rangle$ ($v^{(1)}, v^{(2)} \in V_u$) (denoted as E_u). A query-based network schema is denoted as $QNS_G = (\mathcal{A}_u, \mathcal{R}_u)$, where $\mathcal{A}_u = \{\phi(v_u), v_u \in V_u\}$ and $\mathcal{R}_u = \{\psi(e_u), e_u \in E_u\}$.*

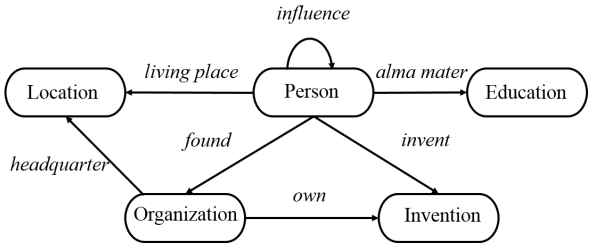

Figure 2: The query-based network schema for query $Q = \{\langle$Larry Page, Sergey Brin$\rangle, \langle$Jerry Yang, David Filo$\rangle\}$.

For example, given query Q, as illustrated in Fig. 2, the entity types, such as *Person, Organization, Education*, are relevant to the query. While types like *Film, Musician, Book* are not relevant, and thus ignored.

The generation procedure of a query-based network schema QNS_G is as below. Given a query Q and the radius D of the schema, first, for each example $r_k \in Q$, we enumerate all the neighbor entities within d-hop ($d \leq D/2$) relations for each entity ($v_k^{(1)}$ and $v_k^{(2)}$). Next, we look up the union of all entity and relation types to generate the $QNS_G = (\mathcal{A}_u, \mathcal{R}_u)$, where $\mathcal{A}_u$ is the union set of entity types, and $\mathcal{R}_u$ is the union set of relation types. For example, given the query Q in the previous example, QNS_G generated by the above process is shown in Fig. 2, where $\mathcal{A}_u = \{$*Person, Organization, Education,* etc.$\}$, and $\mathcal{R}_u = \{$*found, influence, alma mater,* etc.$\}$.

Most existing work assume that meta-paths are provided by users. This assumption can be true for schema-simple HIN (e.g., the DBLP network), it may be infeasible for schema-rich HIN such as the Freebase network. Besides, long meta-paths can be difficult to discover. A simple way can be proposed to automatically generate meta-paths: for a relation instance $\langle v^{(1)}, v^{(2)} \rangle$, one can generate all the possible meta-paths via enumerating all the relations, starting from $v^{(1)}$ and ending with $v^{(2)}$. However, it is time consuming and impractical. As pointed out in [12], the number of possible meta-paths grows sharply with the length of meta-paths. We therefore propose an efficient query-based meta-path generation algorithm (*QMPG*) to generate meta-paths for a relation instance $\langle v^{(1)}, v^{(2)} \rangle$ based on query-based network schema.

Motivated by binary search, given a relation instance, to generate the meta-paths within length-L for the relation instance, we first generate meta-paths that within $L/2$-hop to each entity of the relation instance, and then composite the meta-paths within length-$L/2$, to construct the meta-path candidate set $\mathbf{P}$. We build inverted indices on types of entities and relations to speed up the process.

4.2 Meta-Path Weights Optimization To express the user's need, it is easier for her to provide a query of several examples, and let model learn the weight of each meta-path automatically, rather than specifying the weights of them.

Given a query Q, and query-based meta-path candidate set $\mathbf{P}$, we propose an optimization model to learn the weight of each meta-path $\mathcal{P} \in \mathbf{P}$. We assume one or several meta-paths in $\mathbf{P}$ can capture the most likely LSR held in the query. For example, given a user query $Q = \{\langle$Larry Page, Sergey Brin$\rangle$, $\langle$Jerry Yang, David Filo$\rangle\}$, it's probable that the most likely LSR is a combination of two meta-paths:

$\mathcal{P}_1 =$ *Person* $\xrightarrow{\text{found}}$ *Organization* $\xrightarrow{\text{found}^{-1}}$ *Person*

and

$\mathcal{P}_2 =$ *Person* $\xrightarrow{\text{alma mater}}$ *Education* $\xrightarrow{\text{alma mater}^{-1}}$ *Person*,

indicating that *two Person co-founded an Organization, and both of them graduated from the same Education Institute*. Our task is to discover such important query-based meta-paths by optimizing the weights.

The difficulty of understanding the LSR is that there is a lot of background noise. For example, $\langle$Larry Page, Sergey Brin$\rangle$ and $\langle$Jerry Yang, David Filo$\rangle$ both have the meta-path $\mathcal{P}_1$. But at the same time, they also share meta-paths like $\mathcal{P}_4$, which is a less important meta-path. $\mathcal{P}_4$ can be considered as background noise, since randomly choosing a relation between *Person* and *Person* may have a higher possibility to satisfy $\mathcal{P}_4$. For example, "Larry Page" and "Paul Allen" do not share the important meta-paths, such as $\mathcal{P}_1$, with the examples in Q. We call such artificial pairs (e.g., $\langle$Larry Page, Paul Allen$\rangle$) as "negative examples."[2]

Formally, the negative examples are generated by randomly replacing the subject ($v_k^{(1)}$) (object ($v_k^{(2)}$)) entity of one relation instance by the subject (object) entity of another. A relation instance may have multiple negative examples. We hope to maximize the weights of query-based meta-paths that are mainly shared by positive examples (i.e., examples in Q), but never or rarely appear in negative examples.

Denote $K = |Q|$ as the number of examples in the user query, and $M = |\mathbf{P}|$ as the number of query-based meta-paths. Then, each relation instance would have a feature vector of length M, which is denoted as $\mathbf{x}_k$ ($k = 1, \cdots, K$). The m^{th} element of $\mathbf{x}_k$ is the number of path instances between $v_k^{(1)}$ and $v_k^{(2)}$ of $r_k \in Q$. We also denote $\tilde{\mathbf{x}}_k$ as a negative (or corrupted) example. We use L_2 norm to normalize all feature vectors.

We assign each meta-path a weight ω_m ($m = 1, \cdots, M$), $\omega_m \geq 0$ and regularize $\sum_{m=1}^{M} \omega_m = 1$. Then given the relation instances and the negative examples, we try to find a set of weights in which "important" meta-paths have higher weights, while "unimportant" ones near 0. Inspired by the *ranking loss* proposed as Eq. (17) in [4], we propose the following optimization model:

$$\begin{aligned} (4.2) \qquad \min_{\boldsymbol{\omega}} \quad & \sum_{k=1}^{K} \max\{0, c - \boldsymbol{\omega}^T \mathbf{x}_k + \boldsymbol{\omega}^T \tilde{\mathbf{x}}_k\} \\ \text{s.t.} \quad & \omega_m \geq 0 \qquad \forall m = 1, \cdots, M \\ & \sum_{m=1}^{M} \omega_m = 1 \end{aligned}$$

where $c \in (0, 1]$ is a tuning parameter. If $c = 1$, then we have $\max\{0, c - \boldsymbol{\omega}^T \mathbf{x}_k + \boldsymbol{\omega}^T \tilde{\mathbf{x}}_k\} = 1 - \boldsymbol{\omega}^T \mathbf{x}_k + \boldsymbol{\omega}^T \tilde{\mathbf{x}}_k$ (since $\mathbf{x}_k - \tilde{\mathbf{x}}_k$ is going to be a vector with each entry smaller than or equal to 1 after normalization, with the constraint $\sum_{m=1}^{M} \omega_m = 1$, we then have $\boldsymbol{\omega}^T(\mathbf{x}_k - \tilde{\mathbf{x}}_k) \leq 1$). As a result, this model will essentially maximize the weights of meta-paths that have the biggest difference between positive and negative examples. If $c < 1$, then the model will consider the accident that positive and negative examples share the important meta-paths, and that some of the important meta-paths are missing in some positive examples.

By introducing slack variables $\alpha_k = \max\{0, c - \boldsymbol{\omega}^T \mathbf{x}_k + \boldsymbol{\omega}^T \tilde{\mathbf{x}}_k\}$, the above optimization problem can be turned into linear programming with $(M + K)$ variables and $(M + 1 + 2K)$ constraints:

$$\begin{aligned} (4.3) \qquad \min_{\boldsymbol{\omega}, \alpha} \quad & \sum_{k=1}^{K} \alpha_k \\ \text{s.t.} \quad & \omega_m \geq 0 \qquad \forall m = 1, \cdots, M \qquad \sum_{m=1}^{M} \omega_m = 1 \\ & \alpha_k \geq 0 \qquad \alpha_k \geq c - \boldsymbol{\omega}^T \mathbf{x}_k + \boldsymbol{\omega}^T \tilde{\mathbf{x}}_k \qquad \forall k = 1, \cdots, K \end{aligned}$$

We use the interior point method (Chapter 11 in [3]) to solve the above linear programming problem. Now, we have a weighted query-based meta-path set $\mathbf{P}$, each $\mathcal{P}_m \in \mathbf{P}$ is associated with corresponding weight ω_m. We consider this weighted combination of query-based meta-paths as the LSR held in the query. Notice that rather than using positive-only learning methods [14] that have polynomial time complexity, our linear programming method better fits the online search.

Finally, we propose a fast RelSim-based relation similarity search algorithm by pruning the search space through only preserving the candidates that have at least one common meta-path with the LSR, and building inverted indices on the meta-paths to speed up the searching process.

[2] Sometimes, negative examples may accidentally share meta-paths with positive examples. But we have demonstrated the effectiveness by comparing that with the human provided negative examples in the experiment.

Table 1: **Rel-Full** dataset statistics. #Entities means the number of entities; #Relations means the number of relations.

Relation Categories	#Entities	#Relations	Examples
⟨*Organization, Founder*⟩	9,836,649	560,688,893	⟨Google, Larry Page⟩, ⟨Microsoft, Bill Gates⟩, ⟨Facebook, Mark Zuckerberg⟩
⟨*Book, Author*⟩	16,640,478	981,788,232	⟨Gone with the Wind, Margaret Mitchell⟩, ⟨The Kite Runner, Khaled Hosseini⟩
⟨*Actor, Film*⟩	4,340,986	182,121,412	⟨Leonardo DiCaprio, Inception⟩, ⟨Daniel Radcliffe, Harry Potter⟩, ⟨Jack Nicholson, Head⟩
⟨*Location, Contains*⟩	1,037,791	62,229,669	⟨United States of America, New York⟩, ⟨Victoria, Chillingollah⟩, ⟨New Mexico, Davis House⟩
⟨*Music, Track*⟩	1,653,931	86,658,343	⟨My Worlds, Baby⟩, ⟨21, Someone Like You⟩, ⟨Thriller, Beat It⟩
Total	26,841,657	1,483,834,223	⟨Google, Larry Page⟩, ⟨Leonardo DiCaprio, Inception⟩, ⟨Thriller, Beat It⟩

5 Experiments

In this section, we evaluate the effectiveness and efficiency of our proposed approach.

5.1 Datasets We construct a dataset called **Rel-Full** based on Freebase data as follows: We select five popular relation categories in Freebase, ⟨*Organization, Founder*⟩, ⟨*Book, Author*⟩, ⟨*Actor, Film*⟩, ⟨*Location, Contains*⟩, and ⟨*Music, Track*⟩. For each relation category, we randomly sample 5,000 entity pairs, then enumerate all the neighbor entities and relations within 2-hop of each entity. In Table 1, we show statistics of the five relation categories in **Rel-Full**, including the number of entities, relations, and some corresponding examples. We randomly generate 10 user queries from each relation category in **Rel-Full** by sampling 5 relation instances for each query. As a result, there are 50 queries in total.

5.2 Effectiveness Study We first study the effectiveness of similarity search results and query-based meta-path generation algorithm.

5.2.1 Analysis of Similarity Search Performance We test the performance of RelSim-based relation similarity search. NDCG@$\mathcal{K}$ are used as the evaluation measures. NDCG@$\mathcal{K}$ is the normalized discounted cumulative gain at the given value of $\mathcal{K}$ in the search result. NDCG@$\mathcal{K}$ assume value between 0 and 1, and a higher value indicates a better search result. We use three comparable methods as below. (1) Vector-Space-Model-based Similarity Search (*VSM-S*): based on the relation similarity function defined by vector space model (VSM) [19] for search; (2) Latent-Relational-Analysis-based Similarity Search (*LRA-S*): based on the relation similarity function defined by latent relational analysis (LRA) [18]; and (3) ImplicitWeb-based Similarity Search (*IW-S*): based on the relation similarity measure proposed in [2].

Table 2: Performance (NDCG@$\mathcal{K}$) of relation similarity search on **Rel-Full**.

	NDCG@5	NDCG@10	NDCG@20
VSM-S	0.5389	0.6296	0.7225
LRA-S	0.5880	0.6848	0.7814
IW-S	0.5210	0.6095	0.7010
RelSim-S	0.6395	0.7427	0.8432
RelSim-WS	**0.6651**	**0.7716**	**0.9559**

We re-implement all of the above methods, by replacing the lexical patterns with query-based meta-paths. Notice that, we apply the meta-path set $\mathbf{P}$ to *VSM-S*, *LRA-S* and *IW-S*. In *LRA-S*, we reduce the size of $\mathbf{P}$ to 100 following [18]. While in *IW-S*, we cluster meta-paths with the same parameter setting as in [2]. We denote *RelSim-WS* the framework with RelSim as the similarity measure, and the weight of each query-based meta-path in $\mathbf{P}$ is learned by the optimization model (Section 4.2). Further, *RelSim-S* is *RelSim-WS* without weights learning by setting meta-paths with same weight.

First, we manually label the top-20 results for the 50 queries, to test the quality of ranking lists given by the five methods. We label each candidate relation instance with three relevant levels: 0 (non-relevant), 1 (some-relevant), and 2 (very relevant). We report the NDCG@$\mathcal{K}$ for the 50 queries. Table 2 shows the quality of top-$\mathcal{K}$ ($\mathcal{K} = 5, 10, 20$) search result. From the result, we can see that RelSim-based methods (*RelSim-WS* and *RelSim-S*) outperform the other models. The reasons are as follows: (1) *RelSim-WS* can better use the semantics in schema-rich HINs because it automatically learns the weights of different meta-paths; (2) Both *RelSim-WS* and *RelSim-S* consider more subtle semantics by incorporating the number of shared meta-paths of two relation instances, rather than just normalizing the total number of meta-paths like most vector-based relation similarity measures do (e.g., *VSM-S*). Significance is measured using the t-test with p-value < 0.001.

Then, a case study on top-5 search result is shown in Table 3, based on the query $Q = \{$⟨Google, Larry Page⟩, ⟨Microsoft, Bill Gates⟩, ⟨Facebook, Mark Zuckerberg⟩, ⟨Yahoo!, Jerry Yang⟩, ⟨DreamWorks Animation, David Geffen⟩$\}$. Due to the space limitation, we just show the last names of entities. The most likely LSR held in Q is *(1) the Founder of Organization, (2) who also wins award in the same industry that the Organization runs business in*. The two most important query-based meta-paths below are used to represent the LSR, *Organization* $\xrightarrow{\text{is founded by}}$ *Founder* ($\omega = 0.384$), *Organization* $\xrightarrow{\text{run business in}}$ *Industry* $\xrightarrow{\text{win award in}^{-1}}$ *Founder* ($\omega = 0.274$). From the results, we can see that both *RelSim-WS* and *RelSim-S* get more reasonable results than the other methods. Although the results of the comparable methods contain the semantics (1) in Q, most of them do not imply the semantics (2). For example, in the search result generated by *IW-S*, "Walt Disney" is not

Table 3: Case study on top-5 relation similarity search results on **Rel-Full**.

Query: {⟨Google, Larry Page⟩, ⟨Microsoft, Bill Gates⟩, ⟨Facebook, Mark Zuckerberg⟩, ⟨Yahoo!, Jerry Yang⟩, ⟨DreamWorks Animation, David Geffen⟩}					
Rank	*VSM-S*	*LRA-S*	*IW-S*	*RelSim-S*	*RelSim-WS*
1	⟨Forbes, Forbes⟩	⟨Yelp, Inc., Simmons⟩	⟨Image Comics, Silvestri⟩	⟨DoubleClick, Merriman⟩	⟨Apple, Jobs⟩
2	⟨U-Haul, Shoen⟩	⟨Image Comics, Silvestri⟩	⟨Walt Disney, Disney⟩	⟨YouTube, Chen⟩	⟨IBM, Watson⟩
3	⟨HealthGrades, Hicks⟩	⟨U-Haul, Shoen⟩	⟨Forbes, Forbes⟩	⟨Apple, Wozniak⟩	⟨YouTube, Chen⟩
4	⟨Perot Systems, Perot⟩	⟨Forbes, Forbes⟩	⟨HealthGrades, Hicks⟩	⟨McDonald, McDonald⟩	⟨Linkedin, Hoffman⟩
5	⟨Image Comics, Silvestri⟩	⟨Perot Systems, Perot⟩	⟨New York Library, Dewey⟩	⟨Ford Motor, Ford⟩	⟨DoubleClick, Merriman⟩

Table 4: Example query-based meta-paths on **Rel-Full**. We show the most important four query-based meta-paths of different queries.

Query: {⟨Google, Larry Page⟩, ⟨Microsoft, Bill Gates⟩, etc.}	ω
$Organization \xrightarrow{\text{is founded by}} Founder$	0.384
$Organization \xrightarrow{\text{run business in}} Industry \xrightarrow{\text{win award in}^{-1}} Founder$	0.274
$Organization \xrightarrow{\text{is founded by}} Person \xrightarrow{\text{is influence peer}^{-1}} Founder$	0.174
$Organization \xrightarrow{'\text{s leadership}} Person \xrightarrow{\text{mailing address}} Location \xrightarrow{\text{mailing address}^{-1}} Founder$	0.115
Query: {⟨Google, Larry Page⟩, ⟨Yahoo!, Marissa Mayer⟩, etc.}	ω
$Organization \xrightarrow{\text{run by}} CEO \xrightarrow{\text{job title}^{-1}} Founder$	0.320
$Organization \xrightarrow{\text{founded date}} Date \xrightarrow{\text{graduation date}^{-1}} Founder$	0.229
$Organization \xrightarrow{\text{headquarter}} Location \xrightarrow{\text{education institute}^{-1}} Founder$	0.207
$Organization \xrightarrow{\text{run business in}} Industry \xrightarrow{\text{win award in}^{-1}} Founder$	0.113

an IT company, but at the second ranking position. The result shows that *RelSim-WS* gives the best ranking quality in terms of the human intuition, which is consistent with the previous quality result.

5.2.2 Case Study of Query-based Meta-Paths One of our major contributions is that by representing the LSRs with a set of weighted query-based meta-paths, we are able to distinguish the diverse semantics of LSRs held in a user query.

Table 4 shows the top-four (heavily weighted) meta-paths with the corresponding weights, for two different queries from ⟨Organization, Founder⟩ category. We can see that all the important meta-paths make sense. Besides length-1 meta-paths, we have multi-hop meta-paths that are unexpected yet quite important semantics held in the query. For example, given the query $\{\langle$Google, Larry Page$\rangle, etc.\}$, we can derive meta-paths like *Organization* $\xrightarrow{\text{run business in}}$ *Industry* $\xrightarrow{\text{win award in}^{-1}}$ *Founder* with length larger than one, and it is possible to find related relation instances w.r.t. the multi-hop meta-paths. Interestingly, for queries that have complex semantics, which can not be expressed with length-1 meta-paths, we could express the LSRs between them using multi-hop meta-paths, where originally there are no connections between the entities. For example, given the query $\{\langle$Lord Voldemort, J. K. Rowling$\rangle$, etc.$\}$, there is no length-1 meta-paths connecting them, but we are able to use *Character* $\xrightarrow{\text{appear in}}$ *Book* $\xrightarrow{\text{write}^{-1}}$ *Author* to explain the LSR *Character is in a Book, which is written by Author*.

Table 4 also shows a running example of the optimization model by providing different queries containing some common examples. In the query, ⟨Google, Larry Page⟩ implies different LSRs, such as "is founded by" and "runs by CEO," which are represented with different weighted combination of meta-paths. By providing different examples, such as ⟨Microsoft, Bill Gates⟩ (only satisfies "is founded by") and ⟨Yahoo!, Marissa Mayer⟩ (only satisfies "runs by CEO"), we can see that the meta-paths as well as the weight of the same meta-path change accordingly, which indicates the LSR changes from "is founded by" to "runs by CEO." The reason is that optimization model is able to distinguish the diverse LSRs.

5.3 Efficiency Study We compare *QMPG* with the meta-path generation method proposed in [12] (*PCRW-MPG*) (Fig. 3). We fix the radius of the query-based network schema $D = 4$, varying maximum length of meta-path L ($L = 1, 2, 3, 4$) for both methods, and test on **Rel-Full**. Each query is executed 5 times and the output time is the total average time of the 50 queries. The results show that *QMPG* can significantly improve the efficiency of query-based meta-path generation through applying binary search. For *QMPG*, we build the inverted indices on types of entities and relations in **Rel-Full** on a single machine with 128G memory, 24 CPU cores at 2.0GHZ.

5.4 Parameter Study We first test the impact of various radii of the query-based network schema on the search performance. In Fig. 4, *RelSim-WS* D ($D = 1, 2, 3, 4$) represents the *RelSim-WS* with different radii for the query-based network schema generation. We can see that, the larger radius D, the higher improvement the query-based network

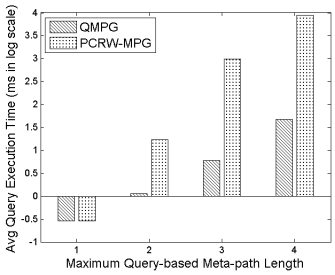

Figure 3: *QMPG* vs. *PCRW-MPG* with different maximum lengths on **Rel-Full**.

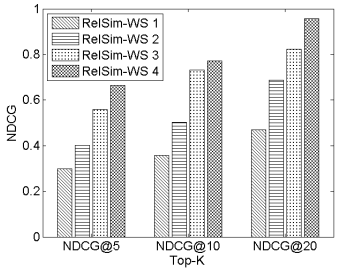

Figure 4: Parameter study of the query-based network schema with different radii.

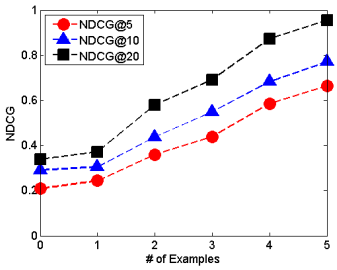

Figure 5: Parameter study of different # of examples (K) in query.

schema achieves. This indicates that more knowledge we have about the query, the better results we can expect, which follows the human intuition. In practice, we set $D = 4$, because if D gets larger, the number of meta-paths will grow prohibitively large.

We then investigate the impact of the number of examples (K) in query on the search results. Fig. 5 shows that when providing more similar examples in a query, the general end-to-end performance will be improved further. The reason is that when providing more examples, the semantic meaning implied in the query could be more specific, which follows the human intuition. In our experiment, we set $K = 5$, because it is difficult to ask a user to provide too many examples in real world.

6 Related Work

In this section, we review the related work on querying graphs or knowledge bases and similarity search.

6.1 Querying Graphs or Knowledge Bases There have been many works on subgraph querying [27] based on traditional subgraph isomorphism using identical label matching. However, we focus on the semantic similarity of graph structure, which does not require identical match. The subgraph querying is enriched with entity similarity and ontology in [25]. Our study yet provides a new perspective by using relation similarity instead of entity or ontology-based similarity. To retrieve data from databases or specially knowledge bases, the standard is often to use structured query languages such as SQL, SPARQL or a formal query model [11]. However, writing structured queries requires extensive experience in query language and data model, and good understanding of particular datasets [9]. We do not assume users have such domain knowledge. Instead, we only require users to provide examples of relation instances.

Query by example is well studied in relational databases. Typical work require structured queries, for example, query graphs or patterns [20, 28], meta-paths [16] or structured query languages, explicitly based on the known underlying schema. Recently, unstructured queries have been studied [10] without schema to query knowledge bases [21, 22, 23]. In contrast, our system allows unstructured queries as examples to query the network by incorporating the network schema.

6.2 Similarity Search Entity similarity search has been a hot research topic for years. Recent studies on entity similarity also find rules/meta-paths very useful. Path ranking algorithm [12], rule mining [7] and meta-path generation [13] have demonstrated the effectiveness of using the mined rules or meta-paths for link prediction-like tasks based on entity similarity, while our work is for relevant relation retrieval. There is no trivial way to apply entity similarity measures to computing relation similarity. There exist works on measuring relation similarity [18]. They usually generate a matrix with rows representing entity pairs and columns representing patterns [1], extracted from text data. Then certain similarity function, like cosine similarity [18], is applied to calculate the relation similarity by using the two corresponding rows in the matrix. We improve these studies in two aspects: First, our approach distinguishes the diverse latent semantic relations existing in a relation instance; second, we are able to utilize the rich structure information in HINs. Several studies have focused on understanding the relationship between entities, by ranking the relationships via pre-defined criteria, to help find similar entities [6] or subgraphs [17]. However, we are able to automatically find the latent relations with the optimization model.

7 Conclusion and Discussion

We have studied relation similarity search in schema-rich heterogeneous information networks. In order to solve the problem, we need to (1) correctly identify the most likely LSR implied by the input query, and (2) provide an efficient search algorithm that can answer the query in a real-time mode. We propose a framework to address the two

requirements. In the framework, we first represent an LSR as a weighted combination of query-based meta-paths that are generated based on the query-based network schema. Second, a novel meta-path-based relation similarity measure RelSim is introduced and used in an efficient similarity search algorithm. Our approach is important for many applications, such as relation based clustering, classification and recommendation. For example, RelSim is easy to be encoded in kernel-based clustering algorithms to canonicalize similar relations [24].

Acknowledgments

Chenguang Wang gratefully acknowledges the support by the National Natural Science Foundation of China (NSFC Grant No. 61472006 and 61272343), the National Basic Research Program (973 Program No. 2014CB340405), and Doctoral Fund of Ministry of Education of China (MOEC RFDP Grant No. 20130001110032). The research is partially supported by the U.S. Army Research Laboratory (ARL) under agreement W911NF-09-2-0053, and by DARPA under agreement No. FA8750-13-2-0008. Research is also partially sponsored by National Science Foundation IIS-1017362, IIS-1320617, IIS-1354329, HDTRA1-10-1-0120, CAREER No. 1453800 and Northeastern TIER 1, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

References

[1] D. T. Bollegala, M. Kusumoto, Y. Yoshida, and K.-I. Kawarabayashi. Mining for analogous tuples from an entity-relation graph. In *IJCAI*, pages 2064–2077, 2013.

[2] D. T. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring the similarity between implicit semantic relations from the web. In *WWW*, pages 651–660, 2009.

[3] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537, 2011.

[5] X. L. Dong, K. Murphy, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, pages 601–610, 2014.

[6] L. Fang, A. D. Sarma, C. Yu, and P. Bohannon. Rex: explaining relationships between entity pairs. *VLDB*, 5(3):241–252, 2011.

[7] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, pages 413–422, 2013.

[8] S. Gu, J. Yan, L. Ji, S. Yan, J. Huang, N. Liu, Y. Chen, and Z. Chen. Cross domain random walk for query intent pattern mining from search engine log. In *ICDM*, pages 221–230, 2011.

[9] H. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi, and C. Yu. Making database systems usable. In *SIGMOD*, pages 13–24, 2007.

[10] N. Jayaram, M. Gupta, A. Khan, C. Li, X. Yan, and R. Elmasri. Gqbe: Querying knowledge graphs by example entity tuples. In *ICDE*, pages 1250–1253, 2014.

[11] G. Kasneci, F. M. Suchanek, G. Ifrim, M. Ramanath, and G. Weikum. Naga: Searching and ranking knowledge. In *ICDE*, pages 953–962, 2008.

[12] N. Lao, T. Mitchell, and W. W. Cohen. Random walk inference and learning in a large scale knowledge base. In *EMNLP*, pages 529–539, 2011.

[13] C. Meng, R. Cheng, S. Maniu, P. Senellart, and W. Zhang. Discovering meta-paths in large heterogeneous information networks. In *WWW*, pages 754–764, 2015.

[14] S. Muggleton. Learning from positive data. In *Inductive logic programming*, pages 358–376. 1997.

[15] Y. Sun and J. Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[16] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB*, pages 992–1003, 2011.

[17] H. Tong and C. Faloutsos. Center-piece subgraphs: problem definition and fast solutions. In *KDD*, pages 404–413, 2006.

[18] P. Turney. Measuring semantic similarity by latent relational analysis. In *IJCAI*, pages 1136–1141, 2005.

[19] P. Turney, M. L. Littman, J. Bigham, and V. Shnayder. Combining independent modules to solve multiple-choice synonym and analogy problems. In *RANLP*, pages 482–486, 2003.

[20] C. Wang, N. Duan, M. Zhou, and M. Zhang. Paraphrasing adaptation for web search ranking. In *ACL*, pages 41–46, 2013.

[21] C. Wang, Y. Song, A. El-Kishky, D. Roth, M. Zhang, and J. Han. Incorporating world knowledge to document clustering via heterogeneous information networks. In *KDD*, pages 1215–1224, 2015.

[22] C. Wang, Y. Song, H. Li, M. Zhang, and J. Han. Knowsim: A document similarity measure on structured heterogeneous information networks. In *ICDM*, pages 506–513, 2015.

[23] C. Wang, Y. Song, H. Li, M. Zhang, and J. Han. Text classification with heterogeneous information network kernels. In *AAAI*, 2016.

[24] C. Wang, Y. Song, D. Roth, C. Wang, J. Han, H. Ji, and M. Zhang. Constrained information-theoretic tripartite graph clustering to identify semantically similar relations. In *IJCAI*, pages 3882–3889, 2015.

[25] Y. Wu, S. Yang, and X. Yan. Ontology-based subgraph querying. In *ICDE*, pages 697–708, 2013.

[26] C. Xiao, W. Wang, X. Lin, and H. Shang. Top-k set similarity joins. In *ICDE*, pages 916–927, 2009.

[27] X. Yan, P. S. Yu, and J. Han. Graph indexing: a frequent structure-based approach. In *SIGMOD*, pages 335–346, 2004.

[28] X. Yu, Y. Sun, P. Zhao, and J. Han. Query-driven discovery of semantically similar substructures in heterogeneous networks. In *KDD*, pages 1500–1503, 2012.