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

Chenguang Wang, Yizhou Sun, Yanglei Song, Jiawei Han, Yangqiu Song, Lidan Wang, Ming Zhang
Outline

Motivation
The issues of previous HIN studies

RelSim
Compute the similarity between relation instances

Experiments
Achieve the state-of-arts similarity search results on five datasets
Heterogeneous Information Networks

• HIN: Network with multiple object types and/or multiple link types, e.g., DBLP.
• Network schema: High-level description of a network.
• Meta-path: A **path/link** in the network schema.
Schema-Simple vs. Schema-Rich Heterogeneous Information Networks

• Previous studies: *Schema-simple HINs*
  • Similarity search in DBLP network: four entity types (Paper, Author, Venue, Term), and several relation types; easy to search: user provide relation(s)

Find similar authors publishing papers at the same venue

Author-Paper-Venue-Paper-Author
Schema-Simple vs. Schema-Rich Heterogeneous Information Networks

• In real world: *Schema-rich HINs*
  • Similarity search in Freebase network: 1,500+ entity types and 35,000+ relation types; hard to search: user CANNOT provide relation(s)

Find similar person serving the same party

User

Given COMPLEX network schema

CANNOT provides relation(s)

Search

Freebase network

Find similar person serving the same party

?
Schema-Simple vs. Schema-Rich Heterogeneous Information Networks

• In real world: *Schema-rich HINs*
  • Similarity search in Freebase network: 1,500+ entity types and 35,000+ relation types; hard to search: user CANNOT provide relation(s)
Relation Similarity Search Problem

1. Users are asked to just provide a set of simple examples
2. We automatically detect the latent semantic relation (LSR) in the query for the users
Relation Similarity Search Example

Query

Barack Obama  John Kerry
George W. Bush  Condoleezza Rice

Latent Semantic Relations

P1: president vs. secretary-of-state (0.45)
P2: same party (0.25)
P3: president vs. presidential candidate (0.15)

Search Result (ranked)

Bill Clinton  Madeleine Albright
John F. Kennedy  Dean Rusk
Richard Nixon  George McGovern

......
Challenges

Q. how to measure the similarity between relation instances by distinguishing diverse latent semantic relation(s)?

Q = \{<\text{Barack Obama, John Kerry}>, <\text{George W. Bush, Condoleezza Rice}>\}

- president vs. secretary-of-state (0.45)
  \textit{is president of} \quad \textit{is secretary of state of}
  \text{President} \quad \rightarrow \quad \text{Country} \quad \leftarrow \quad \text{Secretary of State}

- president vs. presidential candidate (0.15)
  \textit{is president of} \quad \textit{is presidential candidate of}
  \text{President} \quad \rightarrow \quad \text{Country} \quad \leftarrow \quad \text{Presidential Candidate}
RelSim: A Relation Similarity Measure

RelSim: a meta-path-based relation similarity measure.

Given an LSR \( \{w_m, P_m\}_{m=1}^M \), RelSim between \( r \) and \( r' \) is defined as

\[
RS(r, r') = \frac{2 \times \sum_m w_m \min(x_m, x'_m)}{\sum_m w_m x_m + \sum_m w_m x'_m}
\]

**Semantic overlap**: the weighted number of overlapped meta-path based relations between two instances

**Semantic overlap**: the weighted number of total meta-path-based relations satisfied by two instances

Intuition: two relation instances are more similar when sharing more important (heavily weighted) meta-paths

Properties: Range, Symmetric, Self-maximum
Latent Semantic Relation Learning

Number of meta-paths could be very large

\[ RS(r, r') = \frac{2 \times \sum m w_m \min(x_m, x'_m)}{\sum m w_m x_m + \sum m w_m x'_m} \]

The weight/importance of each meta-path is different when query is different

1. Meta-path candidates generation: enumerating all the possible meta-paths between entities in large-scale networks is impractical;
2. Meta-path weights optimization: the real semantic meaning in a query is specific.
Meta-Path Candidates Generation

Query based network schema: a sub-network schema of a schema-rich HIN that only contains the entity and relation types that relevant to the query.

Query based meta-path generation algorithm: using binary search based on the query based network schema.
Meta-Path Weights Optimization

**Intuition:** Discover important query-based meta-paths by optimizing the weights.

e.g. <Larry Page, Sergey Brin> and <Jerry Yang, David Filo> share,

```
PER → alma mater → EDU ← alma mater → PER
```

```
PER → invest → ORG ← employee → PER
```

the later is a less important one (satisfy with randomly choosing instances).

**Negative sample generation:** since there is a lot of background noise. Randomly replacing the subject(object) entity of one instance by the subject(object) entity of another. e.g. <Larry Page, Paul Allen>
Meta-Path Weights Optimization

Inspired by the ranking loss, we propose the optimization model:

\[
\min \sum_{k=1}^{K} \max \{0, c - \omega^T x_k + \omega^T \tilde{x}_k\} \\
\text{s.t. } \omega_m \geq 0 \quad \forall m = 1, \ldots, M \\
\sum_{m=1}^{M} \omega_m = 1 \\
\alpha_k \geq 0 \quad \forall k = 1, \ldots, K
\]

By introducing slack variables, the above optimization problem is turned into a linear programming with \((M + K)\) variables and \((M + 1 + 2K)\) constraints, solved by interior point method:

\[
\min \sum_{k=1}^{K} \alpha_k \\
\text{s.t. } \omega_m \geq 0 \quad \forall m = 1, \ldots, M \\
\sum_{m=1}^{M} \omega_m = 1 \\
\alpha_k \geq 0 \quad \forall k = 1, \ldots, K
\]
Experiments

- Datasets: **five real world datasets** are constructed based on Freebase
  - The largest one is **Rel-Full** dataset: five popular relation categories in Freebase are selected,
  - For each relation category, randomly sample 5,000 entity pairs, then enumerate all the neighbor entities and relations within 2-hop of each entity.

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

Performance (NDCG@K) of relation similarity search on Rel-Full.

<table>
<thead>
<tr>
<th></th>
<th>NDCG@5</th>
<th>NDCG@10</th>
<th>NDCG@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSM-S</td>
<td>0.5389</td>
<td>0.6296</td>
<td>0.7225</td>
</tr>
<tr>
<td>LRA-S</td>
<td>0.5880</td>
<td>0.6848</td>
<td>0.7814</td>
</tr>
<tr>
<td>IW-S</td>
<td>0.5210</td>
<td>0.6095</td>
<td>0.7010</td>
</tr>
<tr>
<td>RelSim-S</td>
<td>0.6395</td>
<td>0.7427</td>
<td>0.8432</td>
</tr>
<tr>
<td>RelSim-WS</td>
<td><strong>0.6651</strong></td>
<td><strong>0.7716</strong></td>
<td><strong>0.9559</strong></td>
</tr>
</tbody>
</table>

Finding #1: Our methods outperform the other methods in a significant way using t-test with p-value < 0.001;
Finding #2: RelSim-WS can better use the semantics in schema-rich HINs because it automatically learns the weights of different meta-paths;
Finding #3: Both RelSim-WS and RelSim-S consider more subtle semantics by incorporating the number of shared meta-paths of two relation instances.
Case Study of Meta-Paths

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

<table>
<thead>
<tr>
<th>Query: {(Google, Larry Page), (Microsoft, Bill Gates), etc.}</th>
<th>( \omega )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization ( \xrightarrow{\text{is founded by}} ) Founder</td>
<td>0.384</td>
</tr>
<tr>
<td>Organization ( \xrightarrow{\text{run business in}} ) Industry ( \xrightarrow{\text{win award in}^{-1}} ) Founder</td>
<td>0.274</td>
</tr>
<tr>
<td>Organization ( \xrightarrow{\text{is founded by}} ) Person ( \xrightarrow{\text{is influence peer}^{-1}} ) Founder</td>
<td>0.174</td>
</tr>
<tr>
<td>Organization ( \xrightarrow{\text{’s leadership}} ) Person ( \xrightarrow{\text{mailing address}^{-1}} ) Location ( \xrightarrow{\text{mailing address}^{-1}} ) Founder</td>
<td>0.115</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query: {(Google, Larry Page), (Yahoo!, Marissa Mayer), etc.}</th>
<th>( \omega )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization ( \xrightarrow{\text{run by}} ) CEO ( \xrightarrow{\text{job title}} ) Founder</td>
<td>0.32</td>
</tr>
<tr>
<td>Organization ( \xrightarrow{\text{founded date}} ) Date ( \xrightarrow{\text{graduation date}^{-1}} ) Founder</td>
<td>0.229</td>
</tr>
<tr>
<td>Organization ( \xrightarrow{\text{headquarter}} ) Location ( \xrightarrow{\text{education institute}} ) Founder</td>
<td>0.207</td>
</tr>
<tr>
<td>Organization ( \xrightarrow{\text{run business in}} ) Industry ( \xrightarrow{\text{win award in}^{-1}} ) Founder</td>
<td>0.113</td>
</tr>
</tbody>
</table>

Finding: Optimization model is able to distinguish the diverse LSRs.
## Conclusion

<table>
<thead>
<tr>
<th>Problem</th>
<th>Relation similarity search in schema-rich heterogeneous information networks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approach</td>
<td>RelSim, to compute the semantic similarity between relation instances.</td>
</tr>
<tr>
<td>Results</td>
<td>Our method performs the best on all the datasets.</td>
</tr>
</tbody>
</table>

Thank You! 😊