A Knowledge-Based Approach to Entity Resolution

Klaus-Dieter Schewe
Software Competence Center Hagenberg and Johannes-Kepler-University Linz
Austria
kd.schewe@scch.at, kd.schewe@faw.at

Qing Wang
Research School of Computer Science
The Australian National University
Australia
qing.wang@anu.edu.au
The problem of entity resolution

- The problem of **entity resolution** is to decide, whether two records/objects in a database correspond to the same real-world object.

- Consider the following records in the relation **PERSON**.

```markdown
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Department</th>
<th>University</th>
</tr>
</thead>
<tbody>
<tr>
<td>i₁</td>
<td>Sue Lee</td>
<td>Department of Philosophy</td>
<td>University of Otago</td>
</tr>
<tr>
<td>i₂</td>
<td>Sue Maneth</td>
<td>Bioethics Centre</td>
<td>University of Otago</td>
</tr>
<tr>
<td>i₃</td>
<td>Sue Lee</td>
<td>School of History and Philosophy</td>
<td>Massey University</td>
</tr>
<tr>
<td>i₄</td>
<td>Sue Lee</td>
<td>Bioethics Centre</td>
<td>University of Otago</td>
</tr>
</tbody>
</table>
```

- Questions:
  - Are Sue Lee (i₁) and Sue Lee (i₃) the same person?
  - Are Sue Lee (i₁) and Sue Lee (i₃) not the same person?
  - ...

- Without sufficient identity information, determining whether or not two personal records correspond to the same real-world person is difficult.
Identity Knowledge

- Assume that the following knowledge is acquired from some sources.

(1) Sue Lee at the Bioethics Centre of the University of Otago has studied at the Department of Philosophy of the University of Otago;
(2) Sue Lee at the Department of Philosophy of the University of Otago has not previously worked or studied at Massey University;
(3) Sue Lee at the Bioethics Centre of the University of Otago changed her surname to Maneth after marriage.

- How can we use such knowledge to determine the identity of people?

- We first need to represent knowledge properly (i.e., knowledge representation). Then,
  - How about the expressiveness of the knowledge representation formalism?
  - How about the complexity results of its related problems?
Problem Statement

- The **identification problem** pervasively exists in many research areas such as scientific communities, information integration, data cleaning, service-oriented architecture, etc.

- It is to determine which objects in a system correspond to the same object in the real-world.

- Restricted by the availability of identity information, determining whether two representations correspond to the same real-world object is not always possible.

- In particular, the problem occurs, if data is collected from different sources (as in stream data analysis / Big Data)

- A desired approach is to provide approximate identities of objects, meanwhile allowing us to improve the accuracy of identities over time.

- To achieve this, capturing the knowledge of why objects are identified to be the same or different real-world objects is equally important to unifying them into one identity.
Aims of the Research

• Many approaches in entity resolution are based on similarity of data values.
• However, if values are equal (or unequal), there is nothing to be gained by similarity measures.
• Therefore, investigate a simple yet expressive framework for representing identity knowledge by means of default rules, supporting examples and exceptions.

To study the decidability and complexity of the containment problem for knowledge patterns (will discuss this in detail later).
Formal Framework – Knowledge Pattern

Fix a family \( \{D_i\}_{i \in I} \) of domains and an identity domain \( D_O \in \{D_i\}_{i \in I} \).

\- A **knowledge pattern** \( P \) is a pair \( \langle \varphi, r \rangle \) where
  \- \( \varphi \) is a **pattern formula** in the form \( \text{IDEN}(x, y) \leftarrow \psi(x_1, \ldots, x_n, x, y) \), in which \( \psi \) is a conjunction of atoms, and \( x \) and \( y \) are variables over \( D_O \), and
  \- \( r \) is a **pattern relation** with \( n \) attributes \( A_1, \ldots, A_n \) that are in 1-1 correspondence to the variables \( x_1, \ldots, x_n \) in \( \varphi \), plus an attribute \( A^* \) with domain \( \{+, -\} \).

\- Each tuple \( t \) in \( r \) “generates” exactly one specific query from \( \psi(x_1, \ldots, x_n, x, y) \), called
  \- an **in-query** if \( t.A^* = + \);
  \- an **ex-query** otherwise.
First Example – Knowledge Pattern

• Suppose that $\text{PERSON} = \{\text{ID}, \text{Name}, \text{Department}, \text{University}\}$.

• The pattern $P_1 = \langle \varphi_1, r_1 \rangle$ describes that two persons are identical if they have the same name.

  $\varphi_1 : \text{IDEN}(x, y) \leftarrow \text{PERSON}(x, z_1, x_2, x_3) \land \text{PERSON}(y, z_1, y_2, y_3)$

<table>
<thead>
<tr>
<th>$A_{z_1}$</th>
<th>$A_{x_2}$</th>
<th>$A_{x_3}$</th>
<th>$A_{y_2}$</th>
<th>$A_{y_3}$</th>
<th>$A^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sue Lee</td>
<td>Bioethics Centre</td>
<td>University of Otago</td>
<td>Department of Philosophy</td>
<td>University of Otago</td>
<td>+</td>
</tr>
<tr>
<td>Sue Lee</td>
<td>$\lambda$</td>
<td>Massey University</td>
<td>Department of Philosophy</td>
<td>University of Otago</td>
<td>−</td>
</tr>
</tbody>
</table>

• It captures the knowledge statements (1) and (2) we mentioned before.

(1) Sue Lee at the Bioethics Centre of the University of Otago has studied at the Department of Philosophy of the University of Otago;
(2) Sue Lee at the Department of Philosophy of the University of Otago has not previously worked or studied at Massey University.
Second Example – Knowledge Pattern

• Suppose that \( \text{PERSON}=\{\text{ID, Name, Department, University}\} \).

• \( P_2 = \langle \varphi_2, r_2 \rangle \) describes that two persons are identical if they are affiliated with the same department and have certain name variations, as stipulated in \( r_2 \).

\[
\varphi_2 : \text{iden}(x, y) \leftarrow \text{PERSON}(x, x_1, z_2, z_3) \land \text{PERSON}(y, y_1, z_2, z_3)
\]

\[
r_2 : \begin{array}{cccccc}
A_{z_2} & A_{z_3} & A_{x_1} & A_{y_1} & A^* \\
\text{Bioethics Centre} & \text{University of Otago} & \text{Sue Lee} & \text{Sue Maneth} & +
\end{array}
\]

• It captures the knowledge statement (3) we mentioned before.

(3) Sue Lee at the Bioethics Centre of the University of Otago changed her surname to Maneth after marriage.
Third Example – Specific Queries

• Consider the pattern $P_1 = \langle \varphi_1, r_1 \rangle$ again.

\[
\varphi_1 : \text{IDEN}(x, y) \leftarrow \text{PERSON}(x, z_1, x_2, x_3) \land \text{PERSON}(y, z_1, y_2, y_3)
\]

\[
\begin{array}{ccccccc}
A_{z_1} & A_{x_2} & A_{x_3} & A_{y_2} & A_{y_3} & A^* \\
\hline
r_1 : & \text{Sue Lee} & \text{Bioethics Centre} & \text{University of Otago} & \text{Department of Philosophy} & \text{University of Otago} & + \\
& \text{Sue Lee} & \lambda & \text{Massey University} & \text{Department of Philosophy} & \text{University of Otago} & -
\end{array}
\]

• $P_1 = \langle \varphi_1, r_1 \rangle$ has an in-query $q_1^+(x, y)$ and an ex-query $q_1^-(x, y)$ generated by the first and second tuples of $r_1$, respectively:

\[-q_1^+(x, y) \equiv \text{PERSON}(x, \text{“Sue Lee”}, \text{“Bioethics Centre”}, \text{“University of Otago”}) \land \\
\quad \text{PERSON}(y, \text{“Sue Lee”}, \text{“Department of Philosophy”}, \text{“University of Otago”})
\]

\[-q_1^-(x, y) \equiv \exists x_2. \text{PERSON}(x, \text{“Sue Lee”}, x_2, \text{“Massey University”}) \land \\
\quad \text{PERSON}(y, \text{“Sue Lee”}, \text{“Department of Philosophy”}, \text{“University of Otago”})
\]
A **knowledge model** $M$ is associated with a finite, non-empty set of knowledge patterns.

The **program** of $M$ consists of knowledge rules that have one-to-one correspondence with knowledge patterns of $M$.

Let $\Sigma^+_P$ and $\Sigma^-_P$ be the set of in-queries and the set of ex-queries of the knowledge pattern $P$, respectively. Then a **(knowledge) rule** of $P$ has an expression of the form

$$\text{IDEN}(x, y) \leftarrow \bigvee_{q^+(x,y) \in \Sigma^+_P} q^+(x, y) \land \neg \bigvee_{q^-(x,y) \in \Sigma^-_P} q^-(x, y).$$

Semantically, the program of a knowledge model is interpreted in the same way as a program in Datalog with negation under the inflationary semantics.
Fourth Example – Knowledge Pattern

• Suppose that \textbf{PERSON} = \{ID, Name, Department, University\}, \textbf{PUBLICATION} = \{ID, Title\} and \textbf{AUTHOR} = \{PID, PubID, Order\}.

• \( P_3 = \langle \varphi_3, r_3 \rangle \) describes that two persons are identical if they have the same name and both co-authored with another author.

\[
\varphi_3 : \text{IDEN}(x, y) \leftarrow \text{PERSON}(x, z_1, x_2, x_3) \land \text{AUTHOR}(x, z_2, x_4) \land \\
\text{PERSON}(y, z_1, y_2, y_3) \land \text{AUTHOR}(y, z_3, y_4) \land \\
\text{AUTHOR}(z', z_2, x_5) \land \text{AUTHOR}(z, z_3, y_5) \land \text{IDEN}(z, z')
\]

\[
r_3 : \begin{array}{cccccccccccc}
A_{z_1} & A_{z_2} & A_{z_3} & A_z' & A_{x_2} & A_{x_3} & A_{x_4} & A_{x_5} & A_{y_2} & A_{y_3} & A_{y_4} & A_{y_5} & A^* \\
\lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & \lambda & +
\end{array}
\]

• \( P_3 \) has \text{IDEN} in the both sides of its knowledge rules, which allows us to incorporate recursion into the process of deducing identity knowledge.
The Containment Problem

- **Why it is important?** Because it can help minimise knowledge patterns of a knowledge model, for improving the efficiency of evaluating knowledge patterns, and consequently the overall performance of a knowledge model.

- Given two knowledge patterns $P_1$ and $P_2$ over the same schema $S$, $P_1$ is contained in $P_2$, denoted as $P_1 \subseteq P_2$, if, for each database instance $I$ of $S$, $R^{P_1}(I) \subseteq R^{P_2}(I)$ holds.

- $P_1$ and $P_2$ are equivalent, denoted as $P_1 \equiv P_2$, if $P_1 \subseteq P_2$ and $P_2 \subseteq P_1$ both hold.

- The containment problem for knowledge patterns is to determine whether or not $P_1 \subseteq P_2$ holds for all instances of $S$.

- Although knowledge patterns involve the union of conjunctive queries and also a restricted form of negation, the containment problem for knowledge patterns is not only decidable but also tractable.
Theorems

Theorem 1. (called Homomorphism Theorem; for details, see the book “Foundations of Databases” by Abiteboul, S., Hull, R. and Vianu, V.)

Let $\varphi$ and $\phi$ be conjunctive queries over the same schema. Then $\varphi \subseteq \phi$ iff there exists a homomorphism from $\phi$ to $\varphi$.

Theorem 2. (for the details, see our paper)

Let $P_1 = \langle \varphi, r_1 \rangle$ and $P_2 = \langle \phi, r_2 \rangle$ be two knowledge patterns over the same schema. Then $P_1 \subseteq P_2$ iff the following condition is satisfied:

$$\forall \varphi_1 \in \Sigma^+_P. (\exists \phi_1 \in \Sigma^+_P. \varphi_1 \subseteq \phi_1 \land$$

$$\forall \phi_2 \in \Sigma^-_{P_2}. (\exists \varphi_2 \in \Sigma^-_{P_1}. (\phi_2 \land \varphi_1 \subseteq \varphi_2 \land \varphi_1))).$$
Example - Containment between Patterns

- Consider the knowledge patterns $P_1 = \langle \varphi_1, r_1 \rangle$ and $P_2 = \langle \varphi_2, r_2 \rangle$.

\[
\begin{align*}
\varphi_1 &= \exists z_1, z_2, z_3. p_1(x, z_1) \land p_2(z_2, y, z_3) &
\varphi_2 &= \exists z_1, z_3. p_1(x, z_1) \land p_2(z_1, y, z_3)
\end{align*}
\]

\[
\begin{array}{c|ccc}
A_{z_1} & A_{z_2} & A_{z_3} & A^* \\
\hline
a & a & \lambda & + & t_1 \\
b & b & \lambda & + & t_2 \\
\lambda & \lambda & b & - & t_3 \\
\end{array}
\quad
\begin{array}{c|ccc}
A_{z_1} & A_{z_3} & A^* \\
\hline
\lambda & \lambda & + & t_1 \\
\lambda & b & - & t_2 \\
\end{array}
\]

- To check whether $P_1 \subseteq P_2$ holds, by Theorem 2, we just need to check whether the following containments (1)-(4) hold:

\[
\begin{align*}
(1). & \quad \varphi_{1}^{t_1} \subseteq \varphi_{2}^{t_1} \\
(2). & \quad \varphi_{1}^{t_3} \land \varphi_{1}^{t_1} \subseteq \varphi_{2}^{t_2} \land \varphi_{1}^{t_1} \\
(3). & \quad \varphi_{1}^{t_2} \subseteq \varphi_{2}^{t_1} \\
(4). & \quad \varphi_{1}^{t_3} \land \varphi_{1}^{t_2} \subseteq \varphi_{2}^{t_2} \land \varphi_{1}^{t_2}.
\end{align*}
\]

where:

\[
\begin{align*}
\varphi_{1}^{t_1} &= \exists z_3. p_1(x, a) \land p_2(a, y, z_3); \\
\varphi_{1}^{t_2} &= \exists z_3. p_1(x, b) \land p_2(b, y, z_3); \\
\varphi_{1}^{t_3} &= \exists z_1, z_2. p_1(x, z_1) \land p_2(z_2, y, b); \\
\varphi_{2}^{t_1} &= \exists z_1, z_3. p_1(x, z_1) \land p_2(z_1, y, z_3); \\
\varphi_{2}^{t_2} &= \exists z_1. p_1(x, z_1) \land p_2(z_1, y, b)
\end{align*}
\]
Example - Equivalence of Patterns

- The knowledge pattern $P_1 = \langle \varphi_1, r_1 \rangle$ can be transformed into $P''_1 = \langle \varphi_2, r'_1 \rangle$ such that $P_1 \equiv P''_1$.

\[
\begin{align*}
\varphi_1 &= \exists z_1, z_2, z_3. p_1(x, z_1) \land p_2(z_2, y, z_3) \\
r_1 &= \begin{array}{c|c|c|c|c}
A_{z_1} & A_{z_2} & A_{z_3} & A^* \\
an & a & \lambda & + & t_1 \\
b & b & \lambda & + & t_2 \\
\lambda & \lambda & b & - & t_3
\end{array}
\end{align*}
\]

\[
\begin{align*}
\varphi_2 &= \exists z_1, z_3. p_1(x, z_1) \land p_2(z_1, y, z_3) \\
r'_1 &= \begin{array}{c|c|c}
A_{z_1} & A_{z_3} & A^* \\
an & \lambda & + \\
b & \lambda & + \\
\lambda & b & -
\end{array}
\end{align*}
\]
Complexity Analysis

- Two complexity measures – data complexity and expression complexity – defined in a similar sense to the ones used for relational query languages.
  
  - **Expression complexity**: measures complexity as a function of the length of the representation of pattern formulae and relations.
  
  - **Data complexity**: measures complexity as a function of the size of the data instance (i.e., pattern formulae and relations are fixed).

- **Theorem 3.** *The containment problem for knowledge patterns is NP-complete with respect to expression complexity.*

- **Theorem 4.** *The containment problem for knowledge patterns is in PTIME with respect to data complexity.*
Conclusion

• The decidability and complexity of reasoning are important issues in the search for a suitable framework of representing identity knowledge.

• We have discussed a simple yet expressive framework for representing identity knowledge.

• One of our main results is that the complexity of determining containment of knowledge patterns is NP-complete in terms of expression complexity and is in PTIME in terms of data complexity.

• These results will lead us to develop a mechanism of finding an optimal representation of knowledge models, which can be used for improving identity knowledge management in the future.