An Improved Semantic Search Model Based on Hybrid Fuzzy Description Logic

Ruixuan Li, Kunmei Wen, Zhengding Lu, Xiaolin Sun, Zhigang Wang

Huazhong University of Science and Technology

rxli@hust.edu.cn
Semantic search (R. Guha, WWW 2003)

Semantic Web + Search Engine
- finds out the internal knowledge in Knowledge Base (KB) to improve the search results gained by current search engines
- evolves to next generation of search engines built on Semantic Web

Two types of search
- searching the current Internet
- searching the Semantic Web portals
Related Work-Tap
**Related Work**

- **Swoogle** is a prototype system of information retrieval (IR). The search results are physical documents on Semantic Web (such as RDF and OWL files).
  - It has not used the semantic structure information in documents.

- Turing center in the University of Washington develops the system **KnowItAll** to extract the information on the Web.
  - Its long-term aim is to replace the search engine by information extraction.
Description Logic (DL): Tableaux algorithm & many optimized.

Classical DLs can only define the certain concepts and properties, which cannot solve the fuzzy problem of ontology system. Fuzzy DL are designed to expand the classic DLs to make it more applicable to ontology system.

Ranking the search results: the number of relationships between entities in a KB will be much larger than the number of entities themselves.
Our Focus

- Combine techniques of semantic reasoning and information retrieval
- Improve semantic search model by supporting imprecise and fuzzy search
Outline

- Semantic Search Model
- Architecture of Semantic Search Model
- RBAC Security Ontology
- Experiment and Evaluation
- Conclusion
Basic Model

- Idea: integrating search and inference
  - Formal DL reasoning: search the resources and relationships
  - Traditional IR: locate the exact resource

- Basic model:
  - Triple[D, Q, R(d_i, Q)]
    - D is the set of documents d_i
    - Q is a query
    - R(d_i, Q) is the similarity between d_i and Q, here d_i ∈ D
  - Here documents and queries are modeled as individuals and concepts respectively
Basic Model

- Basic model $\leftrightarrow$ formal DL (ALC constructors) + traditional IR

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top (Universe)</td>
<td>$\top$</td>
<td>$\Delta^I$</td>
</tr>
<tr>
<td>Bottom (Nothing)</td>
<td>$\bot$</td>
<td>$\phi$</td>
</tr>
<tr>
<td>Atomic Concept</td>
<td>$A$</td>
<td>$A^I \subseteq \Delta^I$</td>
</tr>
<tr>
<td>Atomic Role</td>
<td>$R$</td>
<td>$R^I \subseteq \Delta^I \times \Delta^I$</td>
</tr>
<tr>
<td>Conjunction</td>
<td>$C \cap D$</td>
<td>$C^I \cap D^I$</td>
</tr>
<tr>
<td>Disjunction</td>
<td>$C \cup D$</td>
<td>$C^I \cup D^I$</td>
</tr>
<tr>
<td>Negation</td>
<td>$\neg C$</td>
<td>$\Delta^I \setminus C^I$</td>
</tr>
<tr>
<td>Value restriction</td>
<td>$\forall R.C$</td>
<td>${ a \in \Delta^I \mid \forall b \in \Delta^I, (a,b) \in R^I \implies b \in C^I }$</td>
</tr>
<tr>
<td>Full existential quantification</td>
<td>$\exists R.C$</td>
<td>${ a \in \Delta^I \mid \exists b \in \Delta^I, (a,b) \in R^I \land b \in C^I }$</td>
</tr>
</tbody>
</table>
Different users have different privileges for different resources

A query is defined as the form

- \( Q_i = Q_{i1} \cap Q_{i2} \cap Q_{i3} \)
- \( Q_{i1} \) means a concept
- \( Q_{i2} \) is any formal query about resources or the relationships between them
- \( Q_{i3} \) is a keyword query
Query Form

Four typical queries as follows:

- $Q_{i1}$: **Concept Query**, form as $Q_{i2} = "C"$ where means a concept.

- $Q_{i2}$: **Relationship Query**, form as $Q_{i2} = "C1" \& "C2"$ where $C1$ and $C2$ are concepts.

- $Q_{i3}$: **Keyword Query**, form as $Q_{i3} = "D"$ where $D$ means a keyword which appears in the text. In fact, $Q_{i3}$ belongs to traditional query.

- $Q_{i1} \cap Q_{i3}$: **Conjunctive Query**, form as $Q_{i1} \cap Q_{i3} = ("A" \text{or} "B") \cap D$ where A means a user, B means a role and D means a keyword.
Concept Reasoning. Given $Q_i = Q_{i1}$ where $Q_{i1}$ means concept, we can get all the sub concepts and all the instances are returned.

Relationship Reasoning. Given $Q_i = Q_{i2}$ where $Q_{i2}$ includes two concepts, we can get the relationship between them or null if there is not any relationship.

 Conjunctive Query Reasoning. In fact it integrates inference with search by providing both formal query and keyword query. Given $Q_i = Q_{i1} \land Q_{i3}$ where $Q_{i1}$ means user or role and $Q_{i3}$ is a keyword query.
Result Ranking

- \( R_i = R_{i1} + R_{i2} + R_{i3} \) for the query \( Q_i = Q_{i1} \cap Q_{i2} \cap Q_{i3} \).

- Given \( Q_{i1} \), if \( d \) is the instance of the query \( Q_{i1} \) then the corresponding \( R_{i1} = 1 \), otherwise the value equals to 0.

- \( R_{i2} \) is determined by the important value of the relationship. Not deeply discussion here, need to be further explored.

- Use traditional tf-idf (term frequency • inverse document frequency) method to compute the value of \( R_{i3} \).
Extended Model

- Extended model ←→ formal DL + Type-1 FALC

- Type-1 FALC Syntax and semantics

\[
\begin{align*}
\top^I(d) &= 1 \\
\bot^I(d) &= 0 \\
(C \cap D)^I(d) &= \min\{C^I(d), D^I(d)\} \\
(C \cup D)^I(d) &= \max\{C^I(d), D^I(d)\} \\
\neg C^I(d) &= 1 - C^I(d) \\
(\forall R.C)^I(d) &= \inf_{d' \in \Delta} \max\{1 - R^I(d, d'), C^I(d)\} \\
(\exists R.C)^I(d) &= \sup_{d' \in \Delta} \min\{R^I(d, d'), C^I(d)\}
\end{align*}
\]
Extended Model

- Classic DL such as ALC cannot deal with the imprecise description

- Extended model: use type-1 FALC to replace IR model. The fuzzy degree is equal to the ranking value computed by traditional IR
Complete Model

- Exist **imprecise** terminological axiom (TBox) and **fuzzy** individual membership (ABox).

- It is necessary to maintain **trust degree** for both of TBox and ABox.

- Complete model $\leftarrow\rightarrow$ Type-1 FALC + Type-1 FALC
Propose type-2 FALC based on type-2 fuzzy sets, Type-2 FALC syntax and semantics

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top (Universe)</td>
<td>T</td>
<td>$\Delta^I$</td>
</tr>
<tr>
<td>Bottom (Nothing)</td>
<td>$\bot$</td>
<td>$\phi$</td>
</tr>
<tr>
<td>Atomic Concept</td>
<td>$A_{[a,b]}$</td>
<td>$A^I_{[a,b]} \subseteq \Delta^I$</td>
</tr>
<tr>
<td>Atomic Role</td>
<td>$R_{[a,b]}$</td>
<td>$R^I_{[a,b]} \subseteq \Delta^I \times \Delta^I$</td>
</tr>
<tr>
<td>Conjunction</td>
<td>$C_{[a,b]} \cap D_{[c,d]}$</td>
<td>$(C \cap D)^I_{[\min(a,c),\min(b,d)]}$</td>
</tr>
<tr>
<td>Disjunction</td>
<td>$C_{[a,b]} \cup D_{[c,d]}$</td>
<td>$(C \cup D)^I_{[\max(a,c),\max(b,d)]}$</td>
</tr>
<tr>
<td>Negation</td>
<td>$\neg C_{[a,b]}$</td>
<td>$C^I_{[1-b,1-a]}$</td>
</tr>
<tr>
<td>Value restriction</td>
<td>$\forall R_{[a,b]} \cdot C_{[c,d]}$</td>
<td>$\forall y. \max(R_{[1-b,1-a]}(x,y), C_{[c,d]}(y))$</td>
</tr>
<tr>
<td>Full existential quantification</td>
<td>$\exists R_{[a,b]} \cdot C_{[c,d]}$</td>
<td>$\exists y. \min(R_{[a,b]}(x,y), C_{[c,d]}(y))$</td>
</tr>
</tbody>
</table>
Complete Model

- The complete model uses type-2 fuzzy $ALC$ to replace the formal DL in the extended model. In real-world scenarios, there are imprecise terminological axiom (TBox) and fuzzy individual membership (ABox).

- We apply type-2 fuzzy $ALC$ to deal with the description in ontology for trust degree management (OntoTD).

- We design a simple ontology named OntoTD for trust management.
Outline

- Semantic Search Model
- Architecture of Semantic Search Model
- RBAC Security Ontology
- Experiment and Evaluation
- Conclusion
Architecture of Semantic Search Model

- Query Interface
- Query Processor
- Traditional Search Engine
- Initial Results
- User
- Inference Stop Controller
- Inference Engine
- Results Interface
- Result Ranking Engine
- Knowledge Base
- Other Ontology
- Ontology Translator
- Ontology Base
Outline

- Semantic Search Model
- Architecture of Semantic Search Model
- RBAC Security Ontology
- Experiment and Evaluation
- Conclusion
Great demands: Secure Search, such as Intranet search which must satisfy requirements of access control in the background of government or business.

Role-Based Access Control (RBAC)
- Various ways of specifying policy in the last decade
- Ontology-based approach
- A security ontology based on RBAC policy
We implement the security ontology using OWL DL as the ontology language

```xml
<owl:Class rdf:ID="Subject"/>
<owl:Class rdf:ID="Entity"/>
<owl:Class rdf:ID="Role"/>
<rdfs:subClassOf rdf:resource="#Subject"/>
<rdfs:subClassOf rdf:resource="#Privilege"/>
<owl:ObjectProperty rdf:ID="hasPriv"/>
```
Instance of Roles

Director

ProjectLeader1

ProductionEngineer1 QualityEngineer1

ProjectLeader2

ProductionEngineer2 QualityEngineer2
Instance of PolicyRules

Rule 2
- **grantee**: QualityEngineer1
- **hasPrivilege**: P_02
- **grantor**: Bob
  - **hasPrivilege**: P_02
    - **object**: webpage12
    - **operation**: browse

Rule 3
- **grantee**: ProductionEngineer2
- **hasPrivilege**: P_03
- **grantor**: Bob
  - **hasPrivilege**: P_03
    - **object**: webpage21
    - **operation**: browse

Rule 4
- **grantee**: QualityEngineer2
- **hasPrivilege**: P_04
- **grantor**: Bob
  - **hasPrivilege**: P_04
    - **object**: webpage22
    - **operation**: browse

Rule 1
- **grantee**: ProductionEngineer1
- **hasPrivilege**: P_01
- **grantor**: Bob
  - **hasPrivilege**: P_01
    - **object**: webpage11
    - **operation**: browse
Outline

- Semantic Search Model
- Architecture of Semantic Search Model
- RBAC Security Ontology
- Experiment and Evaluation
- Conclusion
Experiment Settings

- We implement Ontology Security Semantic Search Engine (Onto-SSSE) in Java
  - Use the Lucene as the traditional search engine based on keyword query
  - Use Jena as the reasoning tool based on RBAC security ontology
  - Only implemented basic model
Experiment Settings

- No commonly agreed evaluation methodology and benchmark for semantic search

- The OntoRBAC evaluation dataset
  - OntoRBAC security ontology (including 12 classes, 16 properties and 20 individuals)
  - The set of campus web pages (more than 200MB)
## Semantic Search Results

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Query Form</th>
<th>Query Item</th>
<th>Reasoning Type</th>
<th>Query Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q₁</td>
<td>Q₁₁</td>
<td>“Director”</td>
<td>Concept Reasoning</td>
<td>Sub-roles: ProjectLeader1, ProductionEngineer1, QualityEngineer1, ProjectLeader2, ProductionEngineer2, QualityEngineer2; Privileges:(browse, webpage11), (browse, webpage12), ……</td>
</tr>
<tr>
<td>Q₂</td>
<td>Q₁₂</td>
<td>“ProjectLeader1”&amp; “ProductionEngineer1”</td>
<td>Relationship Reasoning</td>
<td>seniorRoleOf</td>
</tr>
<tr>
<td>Q₃</td>
<td>Q₁₁∩Q₁₂</td>
<td>“Director &amp; computer”</td>
<td>Conjunctive Query Reasoning</td>
<td>webpage list: webpage11, webpage21…... Where include the text “computer” in these web pages</td>
</tr>
</tbody>
</table>
### Comparison with Traditional Search

<table>
<thead>
<tr>
<th>Query form</th>
<th>Reasoning Type</th>
<th>Traditional method</th>
<th>Semantical search</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_1$</td>
<td>Concept Reasoning</td>
<td>Not support</td>
<td>Support</td>
</tr>
<tr>
<td>$Q_2$</td>
<td>Relationship Reasoning</td>
<td>Not support</td>
<td>Support</td>
</tr>
<tr>
<td>$Q_3$</td>
<td>No Reasoning (keyword search)</td>
<td>Support</td>
<td>Support</td>
</tr>
<tr>
<td>$Q_1 \cap Q_3$</td>
<td>Conjunctive Query Reasoning</td>
<td>Not support</td>
<td>Support</td>
</tr>
</tbody>
</table>
Conclusion

- Three semantic search models including basic model, extended and complete one.
- Combine text IR with semantic inference in the model.
- RBAC security ontology - OntoRBAC.
- A semantic search system (Onto-SSSE) is implemented based on the basic model.
Future Work

We plan to achieve improvement in the following aspects:

- Perform search in a larger dataset.
- Implement the extended model and complete model.
- Improve the reasoning efficiency.
Thank you!