Resolving Over-constrained Conditional Temporal Problems Using Semantically Similar Alternatives

Peng Yu¹, Jiaying Shen², Peter Z. Yeh², Brian Williams¹
¹Massachusetts Institute of Technology
²Nuance Communications, Inc.

July 14th, 2016
Over-constrained situations

• Commonly encountered in temporal reasoning:
  – As humans we always ask for more than what we can do.

• Existing approaches of autonomous systems have limited supports for them:
  – Either weakening and/or suspending temporal constraints\[1\], or removing domain requirements completely\[2\].

• Human experts can often resolve such situations through weakening temporal or domain requirements.
  – Leave your work 20 minutes early.
  – Go to a Korean restaurant instead of a Chinese one.

Objective

We want a system that works with the users to resolve over-constrained planning problems through making trade-offs between domain and temporal requirements.
Key Contributions

• We developed the Conflict-Directed Semantic Relaxation algorithm, to compute relaxations for conflicting requirements through weakening domain requirements descriptions, in addition to temporal constraints.

  “Delay your arrival by 5 minutes.”

  “How about a Chinese restaurant instead of a Korean restaurant?”

  “If Korean restaurant does not work, how about Chinese? (instead of BurgerKing)”

– Explore alternative destinations that were not encoded in the original problem;

– Prioritize relaxations that are likely to be preferred by users.
Key Questions

• Computing domain relaxations to resolve conflicts between requirements.

• Prioritizing domain relaxations and enumerating them in best-first order.
Prior Work on Temporal Relaxation

• When a conflict is discovered between constraints, previous relaxation algorithm will try to resolve it through alternative variable assignments, or continuously weakening the temporal constraints.

\[ V_{\text{rest}} = \text{Danji}; \]
\[ \text{Arrive Penn Station} \leq 21:15. \]

\[ \text{Resolve Conflict} \]

\[ V_{\text{rest}} = \text{BarKogi} \]
\[ \text{Arrival} \leq 21:20. \]

\[ V_{\text{rest}} = \text{Googan} \]
\[ \text{Arrival} \leq 21:25. \]

\[ V_{\text{rest}} = \text{Bann} \]
\[ \text{Arrival} \leq 21:28. \]

\[ \cdots \cdots \cdots \]

\[ \text{Arrive Penn Station} \leq 21:18. \]
Conflict Resolution using Domain Relaxations

- In addition, we also weaken the domain descriptions, allowing more options to be considered in order to resolve the conflicts.

\[ V_{\text{rest}} = \text{Danji}; \quad \text{Arrive Penn Station} \leq 21:15. \]

\[ V_{\text{rest}} = \text{BarKogi} \quad \text{Arrival} \leq 21:20. \]

\[ V_{\text{rest}} = \text{Googan} \quad \text{Arrival} \leq 21:25. \]

\[ \text{Cuisine}(V_{\text{rest}}): \text{Korean} \rightarrow \text{Chinese} \]

\[ V_{\text{rest}} = \text{Feng Shui} \]

\[ V_{\text{rest}} = \text{Tang Pavilion} \]

\[ V_{\text{rest}} = \text{Joe’s Shanghai} \]

... ... ...
The domain of variables are specified by a set of semantic constraints, encoded as logical queries.

Given a domain relaxation, we will query the knowledge base for additional domain candidates, using the weakened semantic constraints.
Similarity Measurement

• We need a measurement of the similarity between semantic constraints:
  - Supports a total ordering between alternatives.
  - Works across multiple domains.
  - Distinguishes between concepts represented by the same word: Chinese (cuisine) restaurant and Chinese (genre) movie.

• Currently, the weakening of semantic constraints are guided by a phrase similarity model\[1\], generated by the Word2Vec\[2\] package over Freebase concepts.

<table>
<thead>
<tr>
<th>Cuisine KOREAN</th>
<th>CHINESE</th>
<th>THAI</th>
<th>MEXICAN</th>
<th>AMERICAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>m.048vr</td>
<td>m.01xw9</td>
<td>m.07hxn</td>
<td>m.051zk</td>
<td>m.01zlzf2</td>
</tr>
<tr>
<td>0.7140</td>
<td>0.6945</td>
<td>0.5169</td>
<td>0.3183</td>
<td></td>
</tr>
</tbody>
</table>

Empirical Evaluation

• We invited 9 participants to evaluate the usefulness of CDSR, by using the travel advisor to manage their day-to-day tasks:

  – CDSR found solutions for the participants in 52 out of 54 sessions.
  
  – Temporal relaxation approach provided solutions in only 43 sessions.
  
  – The quality scores indicate that users are in general satisfied with the solution provided by CDSR.
Acknowledgements

• This project is partly supported by the Boeing Company under contract MIT-BA-GTA-1, and the Nuance NL/AI Lab.

• The authors want to thank Szymon Sidor, Jonathan Raiman, Deepak Ramachandran and Daniel Walker for their help and valuable inputs on this project.

• The integrate trip planner tool can be accessed using this URL:
  https://uhura.csail.mit.edu

• The Amazon Echo custom skill can be downloaded using this URL:
  https://github.com/yu-peng/uhura-echo-interface
Visit Us At the Poster Session!

• Details on CDSR’s best-first enumeration procedure with domain relaxations.

• Experiment and user study results, implementation issues and limitations.

• A deeper look into the integrated travel advisory system built on top of CDSR.
Questions
We developed the Conflict-Directed Semantic Relaxation (CDSR) algorithm for enumerating relaxations to over-constrained temporal problems in best-first order.

CDSR resolves conflicts by continuously relaxing temporal constraints, as well as adding additional values to the variable domains through weakening their semantic constraints.
Domain Relaxations

- The domain of some variables are specified by a set of semantic constraints, encoded as SparQL queries for querying the knowledge base.
- CDSR relaxes the semantic constraint and then queries the knowledge base for additional candidates.

\[ \text{Rating}(x) > 4 \land \text{Cuisine}(x) = \text{Chinese} \land \text{Location}(x) = \text{Cambridge} \]

Knowledge Base:
- \{Panda Express, Bamboo Garden\}
- \{Jang Su Jang, Korean Garden\}
- \{TC Garden\}
Relaxing Multiple Domain Constraints

\[ Rating(x) > 4 \land Cuisine(x) = Chinese \land Location(x) = Cambridge \]

\[ Rating(x) > 3 \]

\[ Cuisine(x) = Korean \]

\[ Location(x) = Boston \]

\[ Rating(x) > 2 \]

\[ Cuisine(x) = Korean \]

\[ Location(x) = Boston \]

\[ Cuisine(x) = Thai \]

\[ Location(x) = Boston \]

\[ Location(x) = Brookline \]
Reaching Candidates for Domain Relaxations

- The domain of some variables are specified by a set of semantic constraints, encoded as SparQL queries.
- Given a domain relaxation, we will query the knowledge base for additional domain candidates, using the weakened semantic constraints.

```
KOREAN RESTAURANT
?r ns:type.object.type ns:dining.restaurant.
?c ns:type.object.type ns:dining.cuisine.
FILTER (?c = KOREAN).
```

```
CHINESE RESTAURANT
FILTER (?c = CHINESE).
```
Empirical Evaluation

• We invited 9 participants to evaluate the usefulness of CDSR, by using the travel advisor to manage their day-to-day tasks:
  – CDSR found solutions for the participants in 52 out of 54 sessions.
  – Temporal relaxation approach provided solutions in only 43 sessions.
  – The quality scores indicate that CDSR's solutions are acceptable in most scenarios, users are in general satisfied with the solution provided by the system.

<table>
<thead>
<tr>
<th>Session</th>
<th>Quality Score</th>
<th>Temporal Relaxation</th>
<th>Domain Relaxation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.3 (1.4)</td>
<td>2.0 (2.6)</td>
<td>2.1 (2.7)</td>
</tr>
<tr>
<td>2</td>
<td>2.4 (1.5)</td>
<td>1.3 (2.9)</td>
<td>3.0 (3.3)</td>
</tr>
<tr>
<td>3</td>
<td>2.7 (1.5)</td>
<td>2.9 (3.0)</td>
<td>3.1 (2.8)</td>
</tr>
<tr>
<td>4</td>
<td>3.7 (1.6)</td>
<td>0.3 (0.7)</td>
<td>1.7 (3.4)</td>
</tr>
<tr>
<td>5</td>
<td>3.2 (1.4)</td>
<td>1.9 (2.6)</td>
<td>1.7 (3.0)</td>
</tr>
<tr>
<td>6</td>
<td>3.3 (1.5)</td>
<td>0.6 (1.1)</td>
<td>0.0 (0.0)</td>
</tr>
</tbody>
</table>