Query Sentences as Semantic (Sub) Networks

Joel Booth

IGERT Fellow
Computer Science
Outline

1. Motivation & Context
2. Data and Models
3. Algorithm
4. Preliminary Results and Evaluation
5. Future Work
Motivation

Natural Language Query

Database
Research Context

- Transportation Information Systems
- Integration
  - Natural language processing
  - Ontologies
  - Transportation Query Language
  - HCI concerns
Key Point

- The is an intermediate representation
  - Not a complete QA or DB system
Motivating Example

*Is there a cheap place on my way home?*

- What counts as cheap?
- Where is home?
- How do I get there?
- What is “on the way”? 
Objectives

- Represent natural language queries as semantic networks
- Extract implicit semantics from the NL query
- Minimal reliance on external resources and domain specific knowledge
- Shallow features only
Novelty

- Extensive integration
- Spatial and temporal concepts
  - Paired with explicit linguistic realizations
- Previously unused representation model
The Ontology

- A superset of the database model
  - Transportation
    - Space, time, spatio-temporal, graphs
  - User
    - Preferences, etc.
  - Linguistics
    - Deixis
  - Annotated with WordNet entries
Algorithm Overview

1. Parse the input sentence
2. Generate candidate set
3. Candidate selection
4. Identify the target of the query
5. Build the resultant network
User Input

- A natural language query

*Is there a cheap place to eat on my way home?*
Syntactic Parse

• Structural and Dependency
• Stanford Parser
Candidate Set Generation

- Pairwise comparison between parse nodes & ontological concepts
  - Match metrics
    - WordNet
    - String
  - Match = candidate
  - Each node may have multiple candidates
    - Candidates are not always unique
Candidate Selection

- Each match type is weighted
  - Edit distance
  - Distance in WordNet
  - ...

- Generate a cumulative score for each candidate concept

- Select the highest ranked concept
  - Minimum threshold required
Query Target Identification

- Query target = what the user is looking for
- Based on the syntactic dependency parse
  - 5 patterns
  - Just follow edges in the graph
Query Target Example

- Where is/are ... ?
- Is/are there ... ?

“To be”

[Diagram showing the structure of a question with the copula and noun highlighted]
Graph Generation

- Start with a “bag of concepts”
- Perform an all pairs shortest path to connect within the ontology
Is there a cheap place to eat on my way home?
Tuning

- Weights and thresholds
- Done by hand
- Training set of 12 sentences
Matching Evaluation

- Preliminary testing
- 12 test sentences
- Promising results
  - 43 relevant nodes in the parse graph
    - 39 correctly resolved
    - 3 semi-correctly resolved
    - 1 correctly unresolved
    - No explicit errors
Conclusions

- A semantic representation of queries
- Algorithm based on shallow features
- Promising early results
Future Work

- Extensive testing
- Expanded ontology
- Improved matching metrics
- Complete the “big picture”
Questions?