Semantic Parsing: transforming sentences to logical forms using machine learning models

Minh Le Nguyen
School of Information Science
Japan Advanced Institute of Science and Technology

Semantic Parsing

• Semantic parsing is the process of mapping a natural-language sentence to a complete, detailed semantic representation: logical form or meaning representation (MR).

• For many applications, the desired output is immediately executable by another program.

• Application domains:
  – CLang: RoboCup Coach Language
  – GeoQuery: A Database Query Application
  – Legal domain

Syntactic and Semantic Natural Language Learning

• Most computational research in natural-language learning has addressed “low-level” syntactic processing:
  – Morphology (e.g. past-tense generation)
  – Part-of-speech tagging
  – Chunking
  – Syntactic parsing

• Learning for semantic analysis has been restricted to relatively “shallow” meaning representations:
  – Word sense disambiguation (e.g. SENSEVAL)
  – Semantic role assignment (determining agent, patient, instrument, etc., e.g. FrameNet, PropBank)
  – Information extraction

CLang: RoboCup Coach Language

• In RoboCup Coach competition teams compete to coach simulated players
• The coaching instructions are given in a formal language called CLang

GeoQuery: A Database Query Application

• Query application for U.S. geography database containing about 800 facts [Zelle & Mooney, 1996]

Approach

• Applying ML to the transforming problem
• Motivations
  – Robustness, reduction of development rules
  – Treating ambiguity
  – Handling with the difficulty of consistent rules
• Current work
  – ML for query database language / robcup controlled language
  – ML for legal domain
**Machine Learning Framework**

**Learning phase**
- Structured ML Learner
- Training Examples
- Rules with weight
- NL sentence

**Transforming phase**
- Preprocessing
- Semantic Tagging
- Semantic parsing
- Generate Logical form

**Semantic Tagging**

Pos Tagging: Using our FlexCRF toolkit

- our
- player
- num
- owner

Semantic Tagging

- sentence
- Pos Tagging
- Decoding
- model

**Semantic parsing**

Semantics tagging

- CYK Parsing
- Semantic tree

**Example: Generate LF from a semantic tree**

**Current Work**

- **English Data (CLANG)**
  - Structured SVM (Robocup)
    - Precision: 85%
    - Recall: 74%
  - Maximum entropy model (DB query)
    - Precision 85%
    - Recall 51%

- **Japanese Data**
  - Splitting long sentences into a set of short sentences
  - Mapping NL Japanese sentence to logical form

**Legal domain**

Structured SVM

Online-SCFG

Dependency tree is suitable for the Japanese legal domain

Correspondences between DT's node and predicate in LF
Dependency Parsing

- Input: set of dependency trees and their logical forms
- Output: the correspondences of each node in DT with a predicate in LF
- Method: Using statistical machine translation to align a node in DT to a predicate in LF

Patterns learning

- Plan to participate CONLL-2007 task

Splitting clauses in legal domain

- Using machine learning for splitting clauses
  - NL to LF mapping
  - Combining

Online-SCFG Methods

- Preprocessing
  - Generate a sequence of word tokens
  - Transform a logical form representation into a sequence of atomic logical form

- Using GIZA++ to generate alignment between each word in NL to each token in LF

- Using synchronous grammar to estimate the model for generating logical form

- Using online structured prediction learning to estimate the SCFG grammar

- Some issues for Japanese data
  - Require a formal grammar representation for LF
  - In the case there is no formal grammar it becomes phrase based SMT models

Context-Free Semantic Grammar

- DEVELOPED BY AHO & ULLMAN (1972) AS A THEORY OF COMPILERS THAT COMBINES SYNTAX ANALYSIS AND CODE GENERATION IN A SINGLE PHASE

- GENERATES A PAIR OF STRINGS IN A SINGLE DERIVATION
Developed by Aho & Ullman (1972) as a theory of compilers that combines syntax analysis and code generation in a single phase. Generates a pair of strings in a single derivation.

**Pattern Template**

**QUERY** → **What is CITY / answer(CITY)**

**Synchronous Context-Free Grammars**


**QUERY** → **What is CITY / answer(CITY)**

**Probabilistic Parsing Model**

**d_1**

- CITY
- capital (CITY)
- loc_2 (STATE)
- stateid ('ohio')

**d_2**

- CITY
- capital (CITY)
- loc_2 (RIVER)
- riverid ('ohio')

- Pr(d_1(capital of Ohio) = \exp(1.3) / Z)

**Parsing Model**

- N (non-terminals) = {QUERY, CITY, STATE, ...}
- S (start symbol) = QUERY
- T_m (MRL terminals) = {answer, capital, loc_2, ( , ) , ...}
- T_n (NL words) = {What, is, the, capital, of, Ohio, ...}

**Online structured prediction learning**

- English data (CLANG)
  - It is applicable for Robocup language
  - Precision 89.5 (best precision)
  - Recall 61.2

- Japanese data (110 sentences)
  - Because of spare data problem so the alignment of each word in NL sentence and each token in LF is not good.
  - It is need to verify this problem in detail for improving the accuracy of our model
  - Enlarge the training data?

**Results**

- Precision 89.5 (best precision)
- Recall 61.2
Conclusions

• Learning is applicable for transforming NL to logical form
• The number of training data should be enlarged to make sure the accuracy of the models
• The splitting result for legal Japanese data is attractive

Future work

• Experiment on a larger corpus of Japanese data
• Integrate with the rule-based models
• Transforming a logical form representation to a NL sentence
• Semi-supervised learning models for semantic parsing