Transformation-based Learning for Semantic Parsing

F. Jurcicek, M. Gasic, S. Keizer, F. Mairesse, B. Thomson, K. Yu, S. Young

Cambridge University Engineering Department | {fj228, mg436, sk561, f.mairesse, brmt2, ky219, sjy}@eng.cam.ac.uk

Goals

Develop a fast semantic decoder for dialogue systems

<u>what are the lowest airfare from Washington DC to Boston</u>

GOAL	=	airfare
airfare.type	=	lowest
from.city	=	Washington
from.state	=	DC
to.city	=	Boston

Capability to parse 10 – 100 ASR hypothesis in real time

Robust to speech recognition noise

Trainable on different domains:

- **Air travel information system (ATIS)**
- **Tourist information (TownInfo)**

Motivation

Semantic parsing maps natural language to formal language

Most of the mapping is fairly simple (cities, times, ...) in current dialogue systems.

Inference of compact set of transformation rules which transforms the initial naive semantic hypothesis

Transformations

Triggers	Transformations
„tickets“	replace the goal by „airfare“
„flights * from“	
& GOAL=airfare	replace the goal by flight
„Seattle“	add the slot „to.city=Seattle“
„connecting“	replace the slot „to.city=*“ by „stop.city=*“

Example of parsing

find all the flights between Toronto and San Diego that arrive on Saturday

1. **Parser assigns to initial semantics to input sentence.**

GOAL	=	flight

2. **Rules, whose triggers match the sentence a the hypothesised semantics, are sequentially applied.**

#	trigger	transformation
1	„between Toronto“	add the slot „from.city=Toronto“
2	„and San Diego“	add the slot „to.city=Sand Diego“
3	„Saturday“	add the slot „departure.day=Saturday“

GOAL	=	flight
from.city	=	Toronto
to.city	=	San Diego
departure.day	=	Saturday

3. **Additional rules can fix previous errors**

#	trigger	transformation
4	„arrive“	replace the slot „departure.day=*“ by „arrival.day=*“

GOAL	=	flight
from.city	=	Toronto
to.city	=	San Diego
arrive.day	=	Saturday

Efficiency

Inferred list of rules is very small for domains such as ATIS or TownInfo.

Dataset	# of inferred rules	concepts	# of rules/concepts
ATIS	372	83	4.5
TownInfo	195	28	6.9

Average number of semantic concepts in a sentence is 5.

Python implementation of the parser needs 6ms per sentence on Intel Pentium 2.8 GHz.

Evaluation

Compare TBL Semantic Parser with:

- **A handcrafted Phoenix grammar**
- **The Hidden Vector State model (He & Young, 2006)**
- **Probabilistic Combinatory Categorial Grammar Induction (Zettlemoyer & Collins, 2007)**
- **Markov Logic networks (Meza-Ruiz et al., 2008)**
- **Semantic Tuple Classifiers (Mairesse et al., 2008)**

Evaluation metrics is precision, recall and F-measure of dialogue act items (e.g. *food=Chinese*, *toloc.city=New York*)

Semantic decoder	Item precision	Item recall	Item F
ATIS dataset with transcribed utterances:			
Semantic Tuple Classifiers	96.7	92.4	94.5
Hidden Vector State			90.3
PCCG Induction	95.1	96.7	95.9
Markov Logic Networks	93.4	89.8	91.6
TBL Semantic Parser	96.4	95.1	95.7
TownInfo dataset with transcribed utterances:			
Semantic Tuple Classifiers	97.4	94.0	95.7
Phoenix grammar	96.3	94.2	95.3
TBL Semantic Parser	92.7	94.7	95.4
TownInfo dataset with ASR output:			
Semantic Tuple Classifiers	94.0	83.7	88.6
Phoenix grammar	90.3	79.5	85.5
TBL Semantic Parser	92.7	83.4	87.8

Conclusion: TBL Semantic Parser is robust to noise and competitive with the state of the art.

Open source

The code is available at http://code.google.com/p/tbed-parser

Acknowledgements

This research was partly funded by the UK EPSRC under grant agreement EP/F013930/1 and by the EU FP7 Programme under grant agreement 216594 (CLASSiC project: www.classic-project.org).

SEVENTH FRAMEWORK PROGRAMME

CLASSiC