UNSUPervised Learning AND Modeling of Knowledge AND Intent FOR Spoken Dialogue Systems

Yun-Nung (Vivian) Chen | http://vivianchen.idv.tw
OUTLINE

Introduction

Semantic Decoding [ACL-IJCNLP’15]
  • Ontology Induction
  • Knowledge Graph Propagation
  • Matrix Factorization
  • Experiments

Future Work

Conclusions
A POPULAR ROBOT - BAYMAX
A POPULAR ROBOT - BAYMAX

Baymax is capable of maintaining a good *spoken dialogue system* and *learning* new knowledge for better *understanding* and *interacting* with people.

The goal is to automate learning and understanding procedures in system development.
SPOKEN DIALOGUE SYSTEM (SDS)

Spoken dialogue systems are the intelligent agents that are able to help users finish tasks more efficiently via speech interactions.

Spoken dialogue systems are being incorporated into various devices (smart-phones, smart TVs, in-car navigating system, etc).

Apple’s Siri
Microsoft’s Cortana
Microsoft’s XBOX Kinect
Amazon’s Echo
Samsung’s SMART TV
Google Now

https://www.apple.com/ios/siri/
http://www.amazon.com/oc/echo/
https://www.google.com/landing/now/
LARGE SMART DEVICE POPULATION

The number of global smartphone users will surpass 2 billion in 2016.

As of 2012, there are 1.1 billion automobiles on the earth.

The more natural and convenient input of the devices evolves towards speech.
Traditional SDSs require **manual annotations** for **specific domains** to represent domain knowledge.

**Node:** semantic concept/slot

**Edge:** relation between concepts

- **Restaurant Domain:**
  - restaurant
  - price
  - type
  - location

- **Movie Domain:**
  - movie
  - genre
  - year
  - director
  - director
  - released_in
  - located_in
  - directed_by
A spoken language understanding (SLU) component requires the domain ontology to decode utterances into semantic forms, which contain **core content (a set of slots and slot-fillers)** of the utterance.

**Restaurant Domain**

- **target** = "restaurant"
- **price** = "cheap"
- **type** = "taiwanese"
- **location** = "seattle"

**Movie Domain**

- **target** = "movie"
- **genre** = "action"
- **director** = "james cameron"
CHALLENGES FOR SDS

An SDS in a new domain requires
1) A hand-crafted domain ontology
2) Utterances labelled with semantic representations
3) An SLU component for mapping utterances into semantic representations

With increasing spoken interactions, building domain ontologies and annotating utterances cost a lot so that the data does not scale up.

The goal is to enable an SDS to automatically learn this knowledge so that open domain requests can be handled.
User

find an inexpensive eating place for taiwanese food

Inexpensive Taiwanese eating places include Din Tai Fung, Boiling Point, etc. What do you want to choose? I can help you go there.

Q: How does a dialogue system process this request?
find an inexpensive eating place for taiwanese food
User

find an inexpensive eating place for taiwanese food

Ontology Induction *(semantic slot)*

price

AMOD

Food

NN

target

PREP_FOR

seeking

Organized Domain Knowledge

Intelligent Agent
User

find an inexpensive eating place for Taiwanese food

Ontology Induction \((semantic\ slot)\)

organized Domain Knowledge

Intelligent Agent

organized Domain Knowledge

Seeking

price

AMOD

Food

NN

Target

PREP_FOR

Structure Learning \((inter-slot \ relation)\)
User

find an inexpensive eating place for taiwanese food

Intelligent Agent

seeking

price

AMOD

food

NN

target

seeking=“find”
target=“eating place”
price=“inexpensive”
food=“taiwanese food”
User: find an inexpensive eating place for taiwanese food

Intelligent Agent: Semantic Decoding
- seeking="find"
- target="eating place"
- price="inexpensive"
- food="taiwanese food"
User: find an inexpensive eating place for Taiwanese food

Intelligent Agent:

```
SELECT restaurant {
  restaurant.price = "inexpensive"
  restaurant.food = "Taiwanese food"
}
```
User

find an inexpensive eating place for Taiwanese food

Intelligent Agent

SELECT restaurant {
  restaurant.price = "inexpensive"
  restaurant.food = "Taiwanese food"
}
User: find an inexpensive eating place for Taiwanese food

Intelligent Agent:

```python
SELECT restaurant {
    restaurant.price="inexpensive"
    restaurant.food="Taiwanese food"
}
```

Predicted behavior: navigation

- Din Tai Fung
- Boiling Point
  - : 
  - :
User: find an inexpensive eating place for Taiwanese food

SELECT restaurant {
    restaurant.price = "inexpensive"
    restaurant.food = "Taiwanese food"
}

Predicted behavior: navigation

Behavior Prediction:

Din Tai Fung
Boiling Point

User

find an inexpensive eating place for Taiwanese food

Inexpensive Taiwanese eating places include Din Tai Fung, Boiling Point, etc. What do you want to choose? I can help you go there. (navigation)
User

find an inexpensive eating place for taiwanese food

SELECT restaurant {
  restaurant.price="inexpensive"
  restaurant.food="taiwanese food"
}

Predicted behavior: navigation

Required Domain-Specific Information
FIVE GOALS

1. Ontology Induction (semantic slot)
   - price
   - food
   - target
   - seeking

2. Structure Learning (inter-slot relation)
   Required Domain-Specific Information

3. Surface Form Derivation (natural language)
   SELECT restaurant {
     restaurant.price="inexpensive"
     restaurant.food="taiwanese food"
   }

4. Semantic Decoding
   Predicted behavior: navigation

5. Behavior Prediction

User

find an inexpensive eating place for taiwanese food
FIVE GOALS

User

find an inexpensive eating place for Taiwanese food

1. Ontology Induction *(semantic slot)*

3. Surface Form Derivation *(natural language)*

2. Structure Learning *(inter-slot relation)*

4. Semantic Decoding

5. Behavior Prediction
FIVE GOALS

1. Ontology Induction
2. Structure Learning
3. Surface Form Derivation
4. Semantic Decoding
5. Behavior Prediction

Knowledge Acquisition

User

find an inexpensive eating place for Taiwanese food

SLU Modeling
OUTLINE

Introduction

Semantic Decoding [ACL-IJCNLP’15]
  • Ontology Induction
  • Knowledge Graph Propagation
  • Matrix Factorization
  • Experiments

Future Work

Conclusions
SEMANTIC DECODING

Input: user utterances

Output: the domain-specific semantic concepts included in each individual utterance

OUTLINE

Introduction

Semantic Decoding [ACL-IJCNLP’15]
- Ontology Induction
- Knowledge Graph Propagation
- Matrix Factorization
- Experiments

Future Work

Conclusions
PROBABILISTIC FRAME-SEMANTIC PARSING

FrameNet [Baker et al., 1998]
- a linguistically semantic resource, based on the frame-semantics theory
- “low fat milk” → “milk” evokes the “food” frame;
  “low fat” fills the descriptor frame element

SEMAFOR [Das et al., 2014]
- a state-of-the-art frame-semantics parser, trained on manually annotated FrameNet sentences

Frame-Semantic Parsing for Utterances

Can I have a cheap restaurant?

1st Issue: adapting generic frames to domain-specific settings for SDSs

FT: Frame Target; FE: Frame Element; LU: Lexical Unit
OUTLINE

Introduction

Semantic Decoding [ACL-IJCNLP’15]
  • Ontology Induction
  • Knowledge Graph Propagation (for 1st issue)
  • Matrix Factorization
  • Experiments

Future Work

Conclusions
SEMANTIC DECODING

Input: user utterances

Output: the domain-specific semantic concepts included in each individual utterance

1ST ISSUE: HOW TO ADAPT GENERIC SLOTS TO DOMAIN-SPECIFIC?

KNOWLEDGE GRAPH PROPAGATION MODEL

Assumption: The domain-specific words/slots have more dependency to each other.

The relation matrices allow each node propagate the scores to its neighbor in the knowledge graph, so that the domain-specific words/slots have higher scores during training.
KNOWLEDGE GRAPH CONSTRUCTION

Syntactic dependency parsing on utterances

```
can i have a cheap restaurant
capability expensiveness locale_by_use
```

Slot-based semantic knowledge graph

```
capability `s`
locale_by_use expensiveness
```

Word-based lexical knowledge graph

```
restaurant can w i cheap
have a w
```
The edge between a node pair is weighted as relation importance for build the matrix.

How to decide the weights to represent relation importance?

**KNOWLEDGE GRAPH CONSTRUCTION**

Slot-based semantic knowledge graph:
- capability
- locale_by_use
- expensiveness

Word-based lexical knowledge graph:
- restaurant
- can
- have
- cheap
- a
- w
- i
- s
WEIGHT MEASUREMENT BY EMBEDDINGS

Dependency-based word embeddings

```
<table>
<thead>
<tr>
<th>ccomp</th>
<th>nsubj</th>
<th>dobj</th>
<th>det</th>
<th>amod</th>
</tr>
</thead>
<tbody>
<tr>
<td>can i have a cheap restaurant</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
| can = [0.8 ... 0.24] |
| have = [0.3 ... 0.21] |
```

Dependency-based slot embeddings

```
<table>
<thead>
<tr>
<th>ccomp</th>
<th>nsubj</th>
<th>dobj</th>
<th>det</th>
<th>amod</th>
</tr>
</thead>
<tbody>
<tr>
<td>capability have a expensiveness locale_by_use</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
| expensiveness = [0.12 ... 0.7] |
| capability = [0.3 ... 0.6] |
```

Compute edge weights to represent relation importance

- Slot-to-slot semantic relation $R_s^S$: similarity between slot embeddings
- Slot-to-slot dependency relation $R_s^D$: dependency score between slot embeddings
- Word-to-word semantic relation $R_w^S$: similarity between word embeddings
- Word-to-word dependency relation $R_w^D$: dependency score between word embeddings

$$R_{s}^{SD} = R_{s}^{S} + R_{s}^{D}$$

$$R_{w}^{SD} = R_{w}^{S} + R_{w}^{D}$$
KNOWLEDGE GRAPH PROPAGATION MODEL

Word Observation
- cheap
- food
- restaurant

Slot Candidate
- expensiveness
- food
- locale_by_use

Word Relation Model

Slot Relation Model

Slot Induction

$R_w^{SD}$

$R_s^{SD}$

word relation matrix

slot relation matrix
OUTLINE

Introduction

Semantic Decoding [ACL-IJCNLP’15]
- Ontology Induction
- Knowledge Graph Propagation
- Matrix Factorization (for 2nd issue)
- Experiments

Future Work

Conclusions
**MATRIX FACTORIZATION (MF)**

**FEATURE MODEL**

**Utterance 1**

*i would like a cheap restaurant*

**Utterance 2**

*find a restaurant with chinese food*

**Test Utterance**

*show me a list of cheap restaurants*

---

2nd Issue: hidden semantics cannot be observed but may benefit the understanding performance.
2ND ISSUE: HOW TO LEARN THE IMPLICIT SEMANTICS?

MATRIX FACTORIZATION (MF)

The MF method completes a partially-missing matrix based on the latent semantics by decomposing it into product of two matrices.
MATRIX FACTORIZATION (MF)

The decomposed matrices represent latent semantics for utterances and words/slots respectively.

The product of two matrices fills the probability of hidden semantics.
Bayesian Personalized Ranking for MF

Model implicit feedback

- not treat unobserved facts as negative samples (true or false)
- give observed facts higher scores than unobserved facts

\[ f^+ = \langle u, x^+ \rangle \quad p(f^+) > p(f^-) \]
\[ f^- = \langle u, x^- \rangle \]

\[ p(M_{u,x} = 1 \mid \theta_{u,x}) = \sigma(\theta_{u,x}) = \frac{1}{1 + \exp(-\theta_{u,x})} \]

Objective:

\[ \sum_{f^+ \in \mathcal{O}} \sum_{f^- \notin \mathcal{O}} \ln \sigma(\theta_{f^+} - \theta_{f^-}) \]

The objective is to learn a set of well-ranked semantic slots per utterance.
**2ND ISSUE: HOW TO LEARN THE IMPLICIT SEMANTICS?**

**MATRIX FACTORIZATION (MF)**

The MF method completes a partially-missing matrix based on the latent semantics by decomposing it into product of two matrices.
OUTLINE

Introduction

Semantic Decoding [ACL-IJCNLP'15]
- Ontology Induction
- Knowledge Graph Propagation
- Matrix Factorization
- Experiments

Future Work

Conclusions
EXPERIMENTAL SETUP

Dataset

- **Cambridge University SLU corpus** [Henderson, 2012]
  - Restaurant recommendation in an in-car setting in Cambridge
  - WER = 37%
  - vocabulary size = 1868
  - 2,166 dialogues
  - 15,453 utterances
  - dialogue slot: addr, area, food, name, phone, postcode, price range, task, type

The mapping table between induced and reference slots

**EXPERIMENT 1: QUALITY OF SEMANTICS ESTIMATION**

Metric: Mean Average Precision (MAP) of all estimated slot probabilities for each utterance

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o</td>
<td>w/ Explicit</td>
</tr>
<tr>
<td><strong>Explicit</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>32.5</td>
<td>36.6</td>
</tr>
<tr>
<td>Multinomial Logistic Regression</td>
<td>34.0</td>
<td>38.8</td>
</tr>
</tbody>
</table>
## EXPERIMENT 1: QUALITY OF SEMANTICS ESTIMATION

Metric: Mean Average Precision (MAP) of all estimated slot probabilities for each utterance

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o</td>
<td>w/ Explicit</td>
</tr>
<tr>
<td>Explicit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>32.5</td>
<td>36.6</td>
</tr>
<tr>
<td>Multinomial Logistic Regression</td>
<td>34.0</td>
<td>38.8</td>
</tr>
<tr>
<td>Implicit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>Random</td>
<td></td>
</tr>
<tr>
<td>Majority</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF</td>
<td>Feature Model</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feature Model + Knowledge Graph Propagation</td>
<td></td>
</tr>
</tbody>
</table>
# EXPERIMENT 1: QUALITY OF SEMANTICS ESTIMATION

Metric: Mean Average Precision (MAP) of all estimated slot probabilities for each utterance

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o</td>
<td>w/ Explicit</td>
</tr>
<tr>
<td><strong>Explicit</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>32.5</td>
<td>+</td>
</tr>
<tr>
<td>Multinomial Logistic Regression</td>
<td>34.0</td>
<td>+</td>
</tr>
<tr>
<td><strong>Implicit</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>3.4</td>
<td></td>
</tr>
<tr>
<td>Majority</td>
<td>15.4</td>
<td></td>
</tr>
<tr>
<td><strong>MF</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature Model</td>
<td>24.2</td>
<td></td>
</tr>
<tr>
<td>Feature Model + Knowledge Graph Propagation</td>
<td><strong>40.5</strong></td>
<td>(+19.1%)</td>
</tr>
</tbody>
</table>
Metric: Mean Average Precision (MAP) of all estimated slot probabilities for each utterance

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o</td>
<td>w/ Explicit</td>
</tr>
<tr>
<td>Explicit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>32.5</td>
<td>36.6</td>
</tr>
<tr>
<td>Multinomial Logistic Regression</td>
<td>34.0</td>
<td>38.8</td>
</tr>
<tr>
<td>Implicit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>3.4</td>
<td>22.5</td>
</tr>
<tr>
<td>Majority</td>
<td>15.4</td>
<td>32.9</td>
</tr>
<tr>
<td>MF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature Model</td>
<td>24.2</td>
<td>37.6*</td>
</tr>
<tr>
<td>Feature Model + Knowledge Graph Propagation</td>
<td>40.5*</td>
<td>43.5*</td>
</tr>
</tbody>
</table>

The MF approach effectively models hidden semantics to improve SLU.

Adding a knowledge graph propagation model further improves the results.
EXPERIMENT 2: EFFECTIVENESS OF RELATIONS

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Model</td>
<td>37.6</td>
<td>45.3</td>
</tr>
<tr>
<td>Feature + Knowledge Graph</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic</td>
<td>$\begin{bmatrix} R_w^S &amp; 0 \ 0 &amp; R_s^S \end{bmatrix}$</td>
<td>41.4*</td>
</tr>
<tr>
<td>Dependency</td>
<td>$\begin{bmatrix} R_w^D &amp; 0 \ 0 &amp; R_s^D \end{bmatrix}$</td>
<td>41.6*</td>
</tr>
<tr>
<td>Word</td>
<td>$\begin{bmatrix} R_w^{SD} &amp; 0 \ 0 &amp; 0 \end{bmatrix}$</td>
<td>39.2*</td>
</tr>
<tr>
<td>Slot</td>
<td>$\begin{bmatrix} 0 &amp; 0 \ 0 &amp; R_s^{SD} \end{bmatrix}$</td>
<td>42.1*</td>
</tr>
<tr>
<td>Both</td>
<td>$\begin{bmatrix} R_w^{SD} &amp; 0 \ 0 &amp; R_s^{SD} \end{bmatrix}$</td>
<td></td>
</tr>
</tbody>
</table>

All types of relations are useful to infer hidden semantics.
# EXPERIMENT 2: EFFECTIVENESS OF RELATIONS

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Model</td>
<td>37.6</td>
<td>45.3</td>
</tr>
<tr>
<td>Semantic</td>
<td>41.4*</td>
<td>51.6*</td>
</tr>
<tr>
<td>Dependency</td>
<td>41.6*</td>
<td>49.0*</td>
</tr>
<tr>
<td>Word</td>
<td>39.2*</td>
<td>45.2</td>
</tr>
<tr>
<td>Slot</td>
<td>42.1*</td>
<td>49.9*</td>
</tr>
<tr>
<td>Both</td>
<td>43.5* (+15.7%)</td>
<td>53.4* (+17.9%)</td>
</tr>
</tbody>
</table>

All types of relations are useful to infer hidden semantics.

Combining different relations further improves the performance.
OUTLINE

Introduction

Semantic Decoding [ACL-IJCNLP’15]
  - Ontology Induction
  - Knowledge Graph Propagation
  - Bayesian Personalized Ranking for Matrix Factorization
  - Experiments

Future Work

Conclusions
LOW- AND HIGH-LEVEL UNDERSTANDING

Semantic concepts for individual utterances do not consider high-level semantics (user intents)
The follow-up behaviors are observable and usually correspond to user intents

"can i have a cheap restaurant"
SLU Component
price="cheap"
target="restaurant"
behavior=navigation

"i plan to dine in din tai fung tonight"
SLU Component
restaurant="din tai fung"
time="tonight"
behavior=reservation
BEHAVIOR PREDICTION

Utterance 1
play lady gaga's song bad romance

Utterance 2
i'd like to listen to lady gaga's bad romance

Predicting with Matrix Factorization
OUTLINE

Introduction

Semantic Decoding [ACL-IJCNLP’15]
  - Ontology Induction
  - Knowledge Graph Propagation
  - Bayesian Personalized Ranking for Matrix Factorization
  - Experiments

Future Work

Conclusions
CONCLUSIONS

The ontology induction and knowledge graph construction enable systems to automatically acquire open domain knowledge.

The MF technique for SLU modeling provides a principle model that is able to unify the automatically acquired knowledge, and then allows systems to consider implicit semantics for better understanding.

- Better semantic representations for individual utterances
- Better follow-up behavior prediction

The work shows the feasibility and the potential of improving generalization, maintenance, efficiency, and scalability of SDSs.
Q & A

Thanks for your attentions!!
hi i'd like a restaurant in the cheap price range in the centre part of town

um i'd like chinese food please

how much is the main cost

okay and uh what's the address

great uh and if i wanted to uh go to an italian restaurant instead

italian please

what's the address

i would like a cheap chinese restaurant

something in the riverside
WORD EMBEDDINGS

Training Process
- Each word $w$ is associated with a vector
- The contexts within the window size $c$ are considered as the training data $D$
- Objective function:

\[
\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq i \leq c, i \neq 0} \log p(w_t | w_{t+i})
\]

DEPENDENCY-BASED EMBEDDINGS

Word & Context Extraction

can i have a cheap restaurant

<table>
<thead>
<tr>
<th>Word</th>
<th>Contexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>can</td>
<td>have/ccomp</td>
</tr>
<tr>
<td>i</td>
<td>have/nsub⁻¹</td>
</tr>
<tr>
<td>have</td>
<td>can/ccomp⁻¹, i/nsubj, restaurant/dobj</td>
</tr>
<tr>
<td>a</td>
<td>restaurant/det⁻¹</td>
</tr>
<tr>
<td>cheap</td>
<td>restaurant/amod⁻¹</td>
</tr>
<tr>
<td>restaurant</td>
<td>have/dobj⁻¹, a/det, cheap/amod</td>
</tr>
</tbody>
</table>
DEPENDENCY-BASED EMBEDDINGS

Training Process

- Each word $w$ is associated with a vector $v_w$ and each context $c$ is represented as a vector $v_c$.
- Learn vector representations for both words and contexts such that the dot product $v_w \cdot v_c$ associated with good word-context pairs belonging to the training data $D$ is maximized.
- Objective function:

$$
\arg \max_{v_w, v_c} \sum_{(w,c) \in D} \log \frac{1}{1 + \exp(-v_c \cdot v_w)}
$$

# Slot Mapping Table

Create the mapping if slot fillers of the induced slot are included by the reference slot.

## Induced Slots vs Reference Slot

<table>
<thead>
<tr>
<th>Origin</th>
<th>Food</th>
<th>Food</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>asian</td>
<td>asian</td>
</tr>
<tr>
<td>$u_2$</td>
<td>beer</td>
<td>beer</td>
</tr>
<tr>
<td>$u_k$</td>
<td>japan</td>
<td>japan</td>
</tr>
<tr>
<td>$u_n$</td>
<td>noodle</td>
<td>noodle</td>
</tr>
</tbody>
</table>

Diagram illustrating the mapping between induced slots and reference slot with specific examples such as:
- Locale by use building
- Commerce scenario
- Expensiveness
- Price range
- Part orientational direction
- Locale
- Part inner outer
- Contacting
- Phone
- Sending
- Postcode
The SEMAFOR evaluation

<table>
<thead>
<tr>
<th>Frame Identification (§5.2)</th>
<th>exact matching</th>
<th>partial matching</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td><strong>SemEval 2007 Data</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gold targets</td>
<td>60.21</td>
<td>60.21</td>
</tr>
<tr>
<td>automatic targets (§4)</td>
<td>69.75</td>
<td>54.91</td>
</tr>
<tr>
<td>J&amp;N'07 targets</td>
<td>65.34</td>
<td>49.91</td>
</tr>
<tr>
<td>Baseline: J&amp;N'07</td>
<td>66.22</td>
<td>50.57</td>
</tr>
<tr>
<td><strong>FrameNet 1.5 Release</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gold targets</td>
<td>82.97</td>
<td>82.97</td>
</tr>
<tr>
<td>− unsupported features</td>
<td>80.30</td>
<td>80.30</td>
</tr>
<tr>
<td>&amp; – latent variable</td>
<td>75.54</td>
<td>75.54</td>
</tr>
</tbody>
</table>