Unsupervised Learning and Modeling of Knowledge and Intent for Spoken Dialogue Systems

YUN-NUNG (VIVIAN) CHEN  HTTP://VIVIANCHEN.IDV.TW

APRIL 16TH, 2015 @ NEW YORK UNIVERSITY
Outline

Introduction

Ontology Induction [ASRU’13, SLT’14a]

Structure Learning [NAACL-HLT’15]

Semantic Decoding (submitted)

Conclusions
Outline

Introduction

- Ontology Induction [ASRU’13, SLT’14a]
- Structure Learning [NAACL-HLT’15]
- Semantic Decoding (submitted)

Conclusions
A Popular Robot - Baymax

Big Hero 6 -- Video content owned and licensed by Disney Entertainment, Marvel Entertainment, LLC, etc
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
- 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.
Knowledge Representation/Ontology

Traditional SDSs require **manual annotations** for **specific domains** to represent domain knowledge.

**Restaurant Domain**
- type
- price
- location
- located_in

**Movie Domain**
- genre
- director
- year
- released_in
- directed_by

Node: semantic concept/slot
Edge: relation between concepts
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**

`find a cheap taiwanese restaurant in oakland`

- target=“restaurant”, price=“cheap”, type=“taiwanese”, location=“oakland”

**Movie Domain**

`show me action movies directed by james cameron`

- 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.
Questions to Address

1) Given unlabelled raw audio recordings, how can a system automatically induce and organize domain-specific concepts?

2) With the automatically acquired knowledge, how can a system understand individual utterances?
Interaction Example

User

find a cheap restaurant for asian food

Cheap Asian restaurants include Kelly & Ping, Saigon Shack, etc.
What do you want to choose?

Q: How does a dialogue system process this request?
**SDS Process** – Available Domain Ontology

User

find a cheap restaurant for asian food

Intelligent Agent

Organized Domain Knowledge

- **price**
  - AMOD
- **food**
  - NN
- **seeking**
  - PREP_FOR

**target**
SDS Process – Available Domain Ontology

User: find a cheap restaurant for Asian food

Ontology Induction (semantic slot)

Intelligent Agent: Organized Domain Knowledge

price (AMOD), food (NN), seeking (PREP_FOR)
SDS Process – Available Domain Ontology

User

find a cheap restaurant for asian food

Ontology Induction

(semantic slot)

Intelligent Agent

price

AMOD

food

NN

target

PREP_FOR

seeking

Structure Learning

(inter-slot relation)

Organized Domain Knowledge
SDS Process – Spoken Language Understanding (SLU)

User

Intelligent Agent

find a cheap restaurant for asian food

seeking target food price

seeking="find" target="restaurant" price="cheap" food="asian food"
SDS Process – Spoken Language Understanding (SLU)

User

Intelligent Agent

Semantic Decoding

seeking=“find”
target=“restaurant”
price=“cheap”
food=“asian food”
SDS Process – Dialogue Management (DM)

User:

find a cheap restaurant for asian food

Intelligent Agent:

seeking

price AMOD

food NN

SELECT restaurant {
  restaurant.price="cheap"
  restaurant.food="asian food"
}

User target food price

A MOD

NN

PREP_FOR

target

seeking

price AMOD

food NN

SELECT restaurant {
  restaurant.price="cheap"
  restaurant.food="asian food"
}

Intelligent Agent

seeking

price AMOD

food NN

SELECT restaurant {
  restaurant.price="cheap"
  restaurant.food="asian food"
}
SDS Process – Dialogue Management (DM)

User

User: find a cheap restaurant for asian food

Intelligent Agent

SELECT restaurant {
    restaurant.price="cheap"
    restaurant.food="asian food"
}

Kelly & Ping
Saigon Shack

...
SDS Process – Natural Language Generation (NLG)

User

find a cheap restaurant for asian food

Intelligent Agent

Cheap Asian restaurants include Kelly & Ping, Saigon Shack, etc. What do you want to choose?
User

goals

find a cheap eating place for asian food

SELECT restaurant {
    restaurant.price=“cheap”
    restaurant.food=“asian food”
}
User: find a cheap restaurant for asian food

Goals

Ontology Induction (semantic slot)

SELECT restaurant {
  restaurant.price="cheap"
  restaurant.food="asian food"
}

Semantic Decoding

Structure Learning (inter-slot relation)
Goals

User

find a cheap restaurant for asian food

Ontology Induction

Structure Learning

Semantic Decoding
Knowledge Acquisition

1) Given unlabelled raw audio recordings, how can a system automatically induce and organize domain-specific concepts?

- Knowledge Acquisition
  - Ontology Induction
  - Structure Learning

Restaurant Asking Conversations

Unlabelled Collection

Knowledge Acquisition

Organized Domain Knowledge
SLU Modeling

2) With the automatically acquired knowledge, how can a system understand individual utterances?

SLU Modeling

Semantic Decoding

Organized Domain Knowledge

SLU Component

price=“cheap”

target=“restaurant”

“can i have a cheap restaurant”
Outline

Introduction

- Ontology Induction [ASRU’13, SLT’14a]
- Structure Learning [NAACL-HLT’15]
- Semantic Decoding (submitted)

Conclusions
Ontology Induction [ASRU’13, SLT’14a]

Input: Unlabelled user utterances

Output: Slots that are useful for a domain-specific SDS

Step 1: Frame-semantic parsing on all utterances for creating slot candidates

Step 2: Slot ranking model for differentiating domain-specific concepts from generic concepts

Step 3: Slot selection


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

Step 1: Frame-Semantic Parsing for Utterances

Task: adapting \textit{generic} frames to \textit{domain-specific} settings for SDSs
Step 2: Slot Ranking Model

Main Idea: rank *domain-specific* concepts higher than *generic* semantic concepts

*can I have a cheap restaurant*

Frame: capability
FT LU: can
FE LU: i

Frame: expensiveness
FT LU: cheap

Frame: locale by use
FT/FE LU: restaurant

slot candidate

slot filler
Step 2: Slot Ranking Model

Rank a slot candidate $s$ by integrating two scores

$$w(s) = (1 - \alpha) \log f(s) + \alpha \cdot \log h(s)$$

- slot frequency in the domain-specific conversation
  - slots with higher frequency $\rightarrow$ more important
- semantic coherence of slot fillers
  - domain-specific concepts $\rightarrow$ fewer topics
Step 2: Slot Ranking Model

\( h(s) \): Semantic coherence

- **slot name: quantity**
  - a
  - one
  - three
  - all
  - lower coherence in topic space

- **slot name: expensiveness**
  - cheap
  - inexpensive
  - expensive
  - higher coherence in topic space

\[
h(s) = \frac{1}{V(s)^2} \sum_{x_a, x_b \in V(s)} \text{Sim}(x_a, x_b)
\]

- the slot-filler set corresponding to \( s \)
- measured by cosine similarity between their word embeddings
Step 3: Slot Selection

Rank all slot candidates by their importance scores

\[ w(s) = (1 - \alpha) \log f(s) + \alpha \cdot \log h(s) \]

Output slot candidates with higher scores based on a threshold

<table>
<thead>
<tr>
<th>Slot</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>locale_by_use</td>
<td>0.89</td>
</tr>
<tr>
<td>capability</td>
<td>0.68</td>
</tr>
<tr>
<td>food</td>
<td>0.64</td>
</tr>
<tr>
<td>expensiveness</td>
<td>0.49</td>
</tr>
<tr>
<td>quantity</td>
<td>0.30</td>
</tr>
<tr>
<td>seeking</td>
<td>0.22</td>
</tr>
</tbody>
</table>
Experiments of Ontology Induction

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

Experiments of Ontology Induction

- **Slot Induction Evaluation**: Average Precision (AP) and Area Under the Precision-Recall Curve (AUC) of the slot ranking model to measure quality of induced slots via the mapping table.

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP (%)</td>
<td>AUC (%)</td>
</tr>
<tr>
<td>Baseline: MLE</td>
<td>56.7</td>
<td>54.7</td>
</tr>
<tr>
<td>MLE + Semantic Coherence</td>
<td><strong>71.7</strong> (+26.5%)</td>
<td><strong>70.4</strong> (+28.7%)</td>
</tr>
</tbody>
</table>

Semantic relations help decide domain-specific knowledge.

Induced slots have 70% of AP and align well with human-annotated slots for SDS.
Outline

Introduction

Ontology Induction [ASRU’13, SLT’14a]

Structure Learning [NAACL-HLT’15]

Semantic Decoding (submitted)

Conclusions
Structure Learning [NAACL-HLT’15]

Input: Unlabelled user utterances
Output: Slots with relations

Step 1: Construct a graph to represent slots, words, and relations
Step 2: Compute scores for edges (relations) and nodes (slots)
Step 3: Identify important relations connecting important slot pairs

Step 1: Knowledge Graph Construction

Syntactic dependency parsing on utterances

can  i  have  a  cheap  restaurant

Slot-based semantic knowledge graph

Word-based lexical knowledge graph

can  i  have  a  cheap  restaurant

locale_by_use  expensiveness  capability

restaurant  have  a  cheap
Step 1: Knowledge Graph Construction

The edge between a node pair is weighted as relation importance

How to decide the weights to represent relation importance?

Slot-based semantic knowledge graph

Word-based lexical knowledge graph
Step 2: Weight Measurement
Slot/Word Embeddings Training

Dependency-based word embeddings

ccomp

nsubj
dobj
det
amod

\[\text{can} \rightarrow \text{i} \rightarrow \text{have} \rightarrow \text{a} \rightarrow \text{cheap} \rightarrow \text{restaurant}\]

can = [0.8 \ldots 0.24]

have = [0.3 \ldots 0.21]

expensiveness = [0.12 \ldots 0.7]

capability = [0.3 \ldots 0.6]

Dependency-based slot embeddings

ccomp

nsubj
dobj
det
amod

\[\text{capability} \rightarrow \text{have} \rightarrow \text{a} \rightarrow \text{expensiveness} \rightarrow \text{locale}\_by\_use\]

Step 2: Weight Measurement

Compute edge weights to represent relation importance

- Slot-to-slot relation $L_{ss}$: similarity between slot embeddings
- Word-to-slot relation $L_{ws}$ or $L_{sw}$: frequency of the slot-word pair
- Word-to-word relation $L_{ww}$: similarity between word embeddings
Step 2: Slot Importance by Random Walk

Assumption: the slots with more dependencies to more important slots should be more important

The random walk algorithm computes importance for each slot

\[
\begin{align*}
\hat{r}_s(t+1) &= (1 - \alpha) \hat{r}_s(0) + \alpha L_{ss} L_{sw} \hat{r}_w(t) \\
\hat{r}_w(t+1) &= (1 - \alpha) \hat{r}_w(0) + \alpha L_{ww} L_{ws} \hat{r}_s(t)
\end{align*}
\]

scores propagated from word-layer then propagated within slot-layer

Converged scores can be obtained by a closed form solution.

https://github.com/yvchen/MRRW
Step 3: Identify Domain Slots w/ Relations

The converged slot importance suggests whether the slot is important 

Rank slot pairs by summing up their converged slot importance 

Select slot pairs with higher scores according to a threshold 

(Experiment 1)

(Experiment 2)
Experiments for Structure Learning
Experiment 1: Quality of Slot Importance

Dataset: Cambridge University SLU Corpus

<table>
<thead>
<tr>
<th>Approach</th>
<th>ASR</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP (%)</td>
<td>AUC (%)</td>
</tr>
<tr>
<td>Baseline: MLE</td>
<td>56.7</td>
<td>54.7</td>
</tr>
<tr>
<td>Random Walk:</td>
<td>69.0</td>
<td>68.5</td>
</tr>
<tr>
<td>MLE + Dependent Relations</td>
<td>(+21.8%)</td>
<td>(+24.8%)</td>
</tr>
</tbody>
</table>

Dependent relations help decide domain-specific knowledge.
Experiments for Structure Learning
Experiment 2: Relation Discovery Evaluation

Discover inter-slot relations connecting important slot pairs

<table>
<thead>
<tr>
<th>Rank</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(locale_by_use, NN, food)</td>
</tr>
<tr>
<td>2</td>
<td>(food, AMOD, expensiveness)</td>
</tr>
<tr>
<td>3</td>
<td>(locale_by_use, AMOD, expensiveness)</td>
</tr>
<tr>
<td>4</td>
<td>(seeking, PREP_FOR, food)</td>
</tr>
<tr>
<td>5</td>
<td>(food, AMOD, relational_quantity)</td>
</tr>
<tr>
<td>6</td>
<td>(desiring, DOBJ, food)</td>
</tr>
<tr>
<td>7</td>
<td>(seeking, PREP_FOR, locale_by_use)</td>
</tr>
<tr>
<td>8</td>
<td>(food, DET, quantity)</td>
</tr>
</tbody>
</table>

The reference ontology with the most frequent syntactic dependencies
Experiments for Structure Learning
Experiment 2: Relation Discovery Evaluation

Discover inter-slot relations connecting important slot pairs

<table>
<thead>
<tr>
<th>Rank</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>⟨locale_by_use, NN, food⟩</td>
</tr>
<tr>
<td>2</td>
<td>⟨food, AMOD, expensiveness⟩</td>
</tr>
<tr>
<td>3</td>
<td>⟨locale_by_use, AMOD, expensiveness⟩</td>
</tr>
<tr>
<td>4</td>
<td>⟨seeking, PREP_FOR, food⟩</td>
</tr>
<tr>
<td>5</td>
<td>⟨food, AMOD, relational_quantity⟩</td>
</tr>
<tr>
<td>6</td>
<td>⟨desiring, DOBJ, food⟩</td>
</tr>
<tr>
<td>7</td>
<td>⟨seeking, PREP_FOR, locale_by_use⟩</td>
</tr>
<tr>
<td>8</td>
<td>⟨food, DET, quantity⟩</td>
</tr>
</tbody>
</table>

The reference ontology with the most frequent syntactic dependencies aligns well with the reference one.
Outline

Introduction

Ontology Induction [ASRU’13, SLT’14a]

Structure Learning [NAACL-HLT’15]

Semantic Decoding (submitted)

Conclusions
Outline

Introduction

Ontology Induction [ASRU’13, SLT’14a]

Structure Learning [NAACL-HLT’15]

Semantic Decoding (submitted)

Conclusions
Semantic Decoding

Input: user utterances, automatically learned knowledge
Output: the semantic concepts included in each individual utterance
Matrix Factorization (MF) Feature Model

Ontology Induction

Structure Learning

$F_w$ $F_s$

SLU

Word Observation

cheap food restaurant expensiveness food locale_by_use

Slot Candidate

Train

Test Utterance

show me a list of cheap restaurants

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
Matrix Factorization (MF)
Knowledge Graph Propagation Model

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

\[
\begin{align*}
  f^+ &= \langle u, x^+ \rangle \\
  f^- &= \langle u, x^- \rangle \\
  p(f^+) &> p(f^-)
\end{align*}
\]

Objective:

\[
\sum_{f^+ \in \mathcal{O}} \sum_{f^- \not\in \mathcal{O}} \ln \sigma(\theta f^+ - \theta f^-)
\]

The objective is to learn a set of well-ranked semantic slots per utterance.
Experiments of Semantic Decoding

Experiment 1: Quality of Semantics Estimation

Dataset: Cambridge University SLU Corpus

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>Baseline: Logistic Regression</td>
<td>34.0</td>
<td>38.8</td>
</tr>
<tr>
<td>Random</td>
<td>22.5</td>
<td>25.1</td>
</tr>
<tr>
<td>Majority Class</td>
<td>32.9</td>
<td>38.4</td>
</tr>
<tr>
<td><strong>MF Approach</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature Model</td>
<td>37.6</td>
<td>45.3</td>
</tr>
<tr>
<td>Feature Model + Knowledge Graph Propagation</td>
<td>43.5</td>
<td>53.4</td>
</tr>
<tr>
<td></td>
<td>(+27.9%)</td>
<td>(+37.6%)</td>
</tr>
</tbody>
</table>

The MF approach effectively models hidden semantics to improve SLU.

Adding a knowledge graph propagation model further improves the results.
Experiments of Semantic Decoding
Experiment 2: Effectiveness of Relations

Dataset: Cambridge University SLU Corpus

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>Feature Model</td>
<td>37.6</td>
<td>45.3</td>
</tr>
<tr>
<td>Feature + Knowledge Graph Propagation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semantic Relation</td>
<td>41.4 (+10.1%)</td>
<td>51.6 (+13.9%)</td>
</tr>
<tr>
<td>Dependent Relation</td>
<td>41.6 (+10.6%)</td>
<td>49.0 (+8.2%)</td>
</tr>
<tr>
<td>Both</td>
<td>43.5 (+15.7%)</td>
<td>53.4 (+17.9%)</td>
</tr>
</tbody>
</table>

Both semantic and dependent relations are useful to infer hidden semantics.

Combining both types of relations further improves the performance.
Outline

Introduction

Ontology Induction [ASRU’13, SLT’14a]

Structure Learning [NAACL-HLT’15]

Semantic Decoding (submitted)

Conclusions

UNSUPERVISED LEARNING AND MODELING OF KNOWLEDGE AND INTENT FOR SPOKEN DIALOGUE SYSTEMS
Outline

Introduction

- Ontology Induction [ASRU’13, SLT’14a]
- Structure Learning [NAACL-HLT’15]
- Semantic Decoding (submitted)

Conclusions
Summary

Knowledge Acquisition

Ontology Induction ➔ Semantic relations are useful
Structure Learning ➔ Dependent relations are useful

SLU Modeling

Semantic Decoding ➔ The MF approach builds an SLU model to decode semantics
Conclusions

The knowledge acquisition procedure enables systems to automatically learn open domain knowledge and produce domain-specific ontologies.

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.

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

THANKS FOR YOUR ATTENTIONS!!
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:

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

Dependency-Based Embeddings

Word & Context Extraction

<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:

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

Evaluation Metrics

- **Slot Induction Evaluation**: Average Precision (AP) and Area Under the Precision-Recall Curve (AUC) of the slot ranking model to measure the quality of induced slots via the mapping table.

<table>
<thead>
<tr>
<th>Slot</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>locale_by_use</td>
<td>0.89</td>
</tr>
<tr>
<td>capability</td>
<td>0.68</td>
</tr>
<tr>
<td>food</td>
<td>0.64</td>
</tr>
<tr>
<td>expensiveness</td>
<td>0.49</td>
</tr>
<tr>
<td>quantity</td>
<td>0.30</td>
</tr>
</tbody>
</table>

Precision:
- 1
- 0
- 2/3
- 3/4
- 0

AP = 80.56%
Slot Induction on ASR & Manual Results

The slot importance:

\[ w(s) = (1 - \alpha) \log f(s) + \alpha \cdot \log h(s) \]

Users tend to speak important information more clearly, so misrecognition of less important slots may slightly benefit the slot induction performance.
Slot Mapping Table

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

<table>
<thead>
<tr>
<th>induced slots</th>
<th>reference slot</th>
</tr>
</thead>
<tbody>
<tr>
<td>origin</td>
<td>food</td>
</tr>
<tr>
<td>$u_1$</td>
<td>asian</td>
</tr>
<tr>
<td>$u_2$</td>
<td>:</td>
</tr>
<tr>
<td>$u_k$</td>
<td>japan</td>
</tr>
<tr>
<td>$u_n$</td>
<td>:</td>
</tr>
</tbody>
</table>

- locale by use building
- part orientational
- speak on topic
- seeking
- desire locating
- contacting
- sending
- price range
- area
- phone
- postcode

[back]
Random Walk Algorithm

The converged algorithm satisfies

$$\begin{cases} r_s^* = (1 - \alpha)r_s^{(0)} + \alpha L_{ss} L_{sw} r_w^* \\ r_w^* = (1 - \alpha)r_w^{(0)} + \alpha L_{ww} L_{ws} r_s^* \end{cases}$$

$$r_s^* = (1 - \alpha)r_s^{(0)} + \alpha L_{ss} L_{sw} \left( (1 - \alpha)r_w^{(0)} + \alpha L_{ww} L_{ws} r_s^* \right)$$

$$= (1 - \alpha)r_s^{(0)} + \alpha (1 - \alpha) L_{ss} L_{sw} r_w^{(0)} + \alpha^2 L_{ss} L_{sw} L_{ww} L_{ws} r_s^*$$

$$= \left( (1 - \alpha)r_s^{(0)} e^T + \alpha (1 - \alpha) L_{ss} L_{sw} r_w^{(0)} e^T + \alpha^2 L_{ss} L_{sw} L_{ww} L_{ws} \right) r_s^*$$

$$= M r_s^*$$

The derived closed form solution is the dominant eigenvector of $M$
The SEMAFOR evaluation

Table 5
Frame identification results on both the SemEval 2007 data set and the FrameNet 1.5 release. Precision, recall, and $F_1$ were evaluated under exact and partial frame matching; see Section 3.3. **Bold** indicates best results on the SemEval 2007 data, which are also statistically significant with respect to the baseline ($p < 0.05$).

<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><strong>69.75</strong></td>
<td><strong>54.91</strong></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></td>
<td></td>
</tr>
<tr>
<td>– unsupported features</td>
<td>82.97</td>
<td>82.97</td>
</tr>
<tr>
<td>&amp; – latent variable</td>
<td>80.30</td>
<td>80.30</td>
</tr>
<tr>
<td></td>
<td>75.54</td>
<td>75.54</td>
</tr>
</tbody>
</table>
Matrix Factorization

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

The product of two matrices fills the probability of hidden semantics.
<table>
<thead>
<tr>
<th>Message</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>hi i'd like a restaurant in the cheap price range in the centre part of town</td>
<td>type=restaurant, pricerange=cheap, area=centre</td>
</tr>
<tr>
<td>um i'd like chinese food please</td>
<td>food=chinese</td>
</tr>
<tr>
<td>how much is the main cost</td>
<td>pricerange</td>
</tr>
<tr>
<td>okay and uh what's the address</td>
<td>addr</td>
</tr>
<tr>
<td>great uh and if i wanted to uh go to an italian restaurant instead</td>
<td>food=italian, type=restaurant</td>
</tr>
<tr>
<td>italian please</td>
<td>food=italian</td>
</tr>
<tr>
<td>what's the address</td>
<td>addr</td>
</tr>
<tr>
<td>i would like a cheap chinese restaurant</td>
<td>pricerange=cheap, food=chinese, type=restaurant</td>
</tr>
<tr>
<td>something in the riverside</td>
<td>area=centre</td>
</tr>
</tbody>
</table>