Open Domain Statistical Spoken Dialogue Systems

Steve Young
Contents

• Building an End-to-End Statistical Dialogue System for a Single Domain
  • Spoken Language Understanding
  • Belief Tracking
  • Policies and Dialogue Management
  • Natural Language Generation

• Towards Open-Domain Dialogue Systems
  • A multi-domain architecture
  • Distributed Dialogue Management
  • Incremental Domain Learning
  • On-line Adaptation

• Conclusions
Statistical Spoken Dialogue

To enable fully automatic on-line learning, all components must be trainable from data.

“Deploy, Collect Data, Improve”
I’d like a cheap Italian on the east side of town

Inform:

\[
\text{inform(} \\text{price=cheap, food=Italian, area=east) [0.7]}\]

You’d like a cheap restaurant on the east side of town?
What kind of food would you like?

Confirm request:

\[
\text{confirm-request(} \\text{price=cheap, area=east, food=?)}\]

You’d like a cheap restaurant on the east side of town?
What kind of food would you like?

Confirm request:

\[
\text{confirm-request(food)}\]
Spoken Language Understanding (SLU)

Various decoding strategies

a) Semantic parsing

Grammar Rules

I’d like a cheap Italian on the east side of town

Phoenix Parser

Frame: inform
Type: restaurant
Food: italian
Price: cheap
Area: east

b) Semantic tagging

\[ \hat{Y} = \arg \max_Y P(Y \mid X) \]

eg. HMM, CRF

X = I’d like a cheap Italian on the east side of town
Y = B-inform I-inform o B-price B-food o o B-area I-area I-area I-area

inform price=cheap food=italian area=east

c) Semantic tuple classifier

N-gram Features

SVM-area

SVM-food

SVM-price

etc

I’d like a <p-value> <f-value> on the <a-value> side of town

area=east [p=0.7]
food=italian [p=0.8]
price=cheap [p=0.5]

etc
## SLU Performance

### Cambridge Restaurant System:
- Noisy in-car data, various conditions, 37% average word error rate (WER)
- 10571 training utterances, 4882 test utterances

<table>
<thead>
<tr>
<th>Features</th>
<th>Trained On</th>
<th>F-Score</th>
<th>Item Cross Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phoenix</td>
<td>—</td>
<td>0.69</td>
<td>2.78</td>
</tr>
<tr>
<td>CRF</td>
<td>ASR 1-best</td>
<td>0.67</td>
<td>2.75</td>
</tr>
<tr>
<td>N-grams</td>
<td>ASR 1-best</td>
<td>0.69</td>
<td>1.79</td>
</tr>
<tr>
<td>N-grams</td>
<td>ASR 2-best</td>
<td>0.70</td>
<td>1.72</td>
</tr>
<tr>
<td>Weighted N-grams</td>
<td>ASR 10-best</td>
<td>0.71</td>
<td>1.76</td>
</tr>
<tr>
<td>Weighted N-grams</td>
<td>Confusion Network</td>
<td>0.73</td>
<td>1.68</td>
</tr>
<tr>
<td>Weighted N-grams + Context</td>
<td>Confusion Network</td>
<td>0.77</td>
<td>1.43</td>
</tr>
</tbody>
</table>

Belief Tracking

Aim: to maintain a distribution over all dialogue state variables using SLU output at each turn as evidence

3 principal approaches:
• rule-based
• dynamic Bayesian network
• discriminative model (eg RNN)
Dynamic Bayesian Networks (DBNs)

Goal

User Behaviour

User Act

Memory

Recognition Errors

History

Observation at time t

I'm looking for an Indian restaurant…

Ontology

type = bar, restaurant, hotel

food = french, chinese, italian, …

All nodes conditioned by previous action

… and previous time-slice

Recurrent Neural Net Belief Tracking

Word-Based Dialog State Tracking with Recurrent Neural Networks
M. Henderson, B. Thomson and S. Young, SigDial 2014, Philadelphia, PA
Belief Tracking Performance

Cambridge Restaurant System (Dialog State Tracking Challenge 2):
- Telephone data, various conditions, 20% to 40% average word error rate (WER)
- 1612 training dialogs, 1117 test dialogs
- Joint Slot Accuracy (fraction of turns in which all goal labels are correct)
- Joint L2 (L2 norm between tracker output distribution and reference)

<table>
<thead>
<tr>
<th>System</th>
<th>Features</th>
<th>Accuracy</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>SLU</td>
<td>61.6%</td>
<td>0.74</td>
</tr>
<tr>
<td>Bayes Net</td>
<td>SLU</td>
<td>67.5%</td>
<td>0.55</td>
</tr>
<tr>
<td>Delex RNN</td>
<td>SLU</td>
<td>73.7%</td>
<td>0.41</td>
</tr>
<tr>
<td>Full RNN</td>
<td>SLU</td>
<td>74.2%</td>
<td>0.39</td>
</tr>
<tr>
<td>Delex RNN</td>
<td>ASR</td>
<td>74.6%</td>
<td>0.38</td>
</tr>
<tr>
<td>Full RNN</td>
<td>ASR</td>
<td>76.8%</td>
<td>0.35</td>
</tr>
</tbody>
</table>

The Second Dialog State Tracking Challenge
M. Henderson, B. Thomson and J. Williams, SigDial 2014, Philadelphia, PA
Dialog Management

Partially Observable Markov Decision Process
- action at each turn is function of belief state $b$
- policy optimised by maximising expected cumulative reward $R$
- trained on corpora, user simulator or on-line

Exact solutions intractable, but wide range of approximations:
- gradient ascent directly on policy $\pi$ (NAC)
- maximise GP approximation of Q-function (GP-SARSA)

Natural Actor-Critic

$$\pi(a \mid b, \theta) = \frac{e^{\theta \cdot \phi_a(b)}}{\sum_{a'} e^{\theta \cdot \phi_{a'}(b)}}$$

Action specific features $\phi_a(b)$ defined on $b$

Policy defined directly on softmax $\theta \cdot \phi_a(b)$

$$J(\theta) = \mathbb{E} \left[ \frac{1}{T} \sum_t r(b_t, a_t) \mid \pi_\theta \right]$$

Cost function is sum over observed per turn rewards

Optimise using natural gradient ascent

$$\tilde{\nabla}J(\theta) = F_{\theta}^{-1} \nabla J(\theta)$$

Gradient is estimated by sampling dialogues so Fisher Information Matrix does not need to be explicitly computed.

F. Jurcicek, B. Thomson and S. Young (2011). "Natural Actor and Belief Critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as POMDPs." ACM Transactions on Speech and Language Processing, 7(3)
GP-SARSA

\[ Q_0^\pi(b, a) \sim GP(0, k((b, a), (b, a))) \]

\[ Q^\pi(b, a) = E_\pi(R) \] is expected total reward \( R \) following policy \( \pi \) from point \( (b, a) \)

Given trajectory \( B_t = (b_1, a_1), \ldots, (b_t, a_t) \) and rewards \( r_t = r_1, \ldots, r_t \)

posterior is

\[ Q_t^\pi(b, a) \mid r_t, B_t \sim N(Q(b, a), \text{cov}((b, a), (b, a))) \]

GP-SARSA
Reinforcement Learning

Choose: \( a_{t+1} \sim Q_t^\pi(b_t, a_t) \)
Update: \( b_t \rightarrow b_{t+1} \)
Observe: \( r_{t+1} \)
Update: \( Q_t^\pi \rightarrow Q_{t+1}^\pi \)

Dialog Manager Performance

Cambridge Restaurant System:
- Reward = +20 for success -1 per turn
- User simulator-based training, 100k dialogs
- Telephone-based on-line training, 1200 dialogs
- Telephone-based real-user testing, 500 dialogs
- Telephone speech recognition, 20% average word error rate (WER)

<table>
<thead>
<tr>
<th>Method</th>
<th>Training</th>
<th>Reward</th>
<th>Success Rate</th>
<th>#Turns</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAC</td>
<td>Simulator</td>
<td>11.9</td>
<td>91.8%</td>
<td>6.5</td>
</tr>
<tr>
<td>GP-Sarsa</td>
<td>Simulator</td>
<td>11.6</td>
<td>91.2%</td>
<td>6.6</td>
</tr>
<tr>
<td>GP-Sarsa</td>
<td>On-line</td>
<td>13.4</td>
<td>96.8%</td>
<td>6.0</td>
</tr>
</tbody>
</table>

You’d like a cheap restaurant on the east side of town? What kind of food would you like?

3 principal approaches:
- hand-crafting with parameterised templates
- generative linguistic rules
- data driven using “over-generate and filter” approach
Constrained RNN Generation

Inform(name=Seven_Days, food=Chinese)

(dialog act 1-hot representation)

RNN trained on data pairs consisting of
a) 1-hot representation of system dialog act
b) corresponding delexicalised output utterance

Generation Performance

Open-Domain Statistical Dialogue
Objectives

To develop spoken dialogue systems which:

1. allow users to reference multiple domains within a single conversation

2. supports natural conversation even in rarely visited domains

3. can learn automatically on-line through interaction with user

“Deploy, Collect Data, Improve”

Note that user in the loop enables on-line reinforcement learning
Hello, how can I help you?

*What appointments do I have tomorrow?*

*You have a meeting at 10am with John and a teleconf at noon with Bill.*

*I need to go to London first thing, can you reschedule the meeting with John?*

*John is free tomorrow at 3pm, is that ok?*

Yes, that's fine. I also need a taxi to the station.

*Meeting with John at 15.00 is confirmed. What time do you need the taxi?*

*When does the train depart to London?*

*The 9.15am gets in at 10.06.*

*When is the one before that?*

*The train before that leaves at 8.45am and arrives at 9.40.*

*Ok I will take that, book the taxi for 8.15am from my house.*

*Ok, I will book the taxi for 8.15am, is that correct?*

Yes that's right.

Ok. Do you need anything else?

*Not for now thanks.*
Run-time Architecture

Speech Input

Topic DM

Cal DM

Taxi DM

Train DM

Committee Manager

NLG

Speech Output

Domain Factory

Ontology

Belief State Manager

"What appointments ..."

"... go to London...

"... need a taxi ...

"Q_i(b,a)"
Distributed Dialog Management

• Each DM operates independently, receives speech, tracks its own beliefs and proposes system actions

• DM’s operate as a Bayesian Committee Machine, each machine’s Q-value has a confidence attached to it:

\[
\tilde{Q}(b,a) = \sum Q(b,a) \sum_{i=1}^{M} \sum_{i}^{Q}(b,a)^{-1} \tilde{Q}_i(b,a)
\]

\[
\sum Q(b,a)^{-1} = -(M-1) \ast k((b,a),(b,a))^{-1} + \sum_{i=1}^{M} \sum_{i}^{Q}(b,a)^{-1}
\]

• Reinforcement learning operates on the group, distributing rewards at each turn according to previous action selection.

Modular, flexible, incremental, trainable on-line, …

Initially pool all available data and learn generic models.
Incremental Domain Learning

Refine with more data using generic models as priors

## Performance of Generic Policies

<table>
<thead>
<tr>
<th>Strategy</th>
<th>#Dialogs</th>
<th>Restaurant</th>
<th>Hotel</th>
</tr>
</thead>
<tbody>
<tr>
<td>in-domain</td>
<td>250</td>
<td>62.5%</td>
<td>64.3%</td>
</tr>
<tr>
<td>in-domain</td>
<td>500</td>
<td>67.5%</td>
<td>70.1%</td>
</tr>
<tr>
<td><strong>generic</strong></td>
<td>500</td>
<td><strong>73.0%</strong></td>
<td><strong>76.2%</strong></td>
</tr>
<tr>
<td>in-domain</td>
<td>2500</td>
<td>83.9%</td>
<td>85.9%</td>
</tr>
<tr>
<td>in-domain</td>
<td>5000</td>
<td>86.4%</td>
<td>86.9%</td>
</tr>
<tr>
<td><strong>generic</strong></td>
<td>5000</td>
<td><strong>86.5%</strong></td>
<td><strong>87.1%</strong></td>
</tr>
</tbody>
</table>

Success rates averaged over 10 policies and 1000 dialogues per condition

---

Distributed Dialogue Policies for Multi-Domain Statistical Dialogue Management
On-line Adaptation with Real Users

Performance is acceptable after only 50 dialogues in the new domain.

San Francisco Restaurant Domain

a) with generic prior
b) no prior
Conclusions

• End-to-end statistical dialogue is feasible, and can match or exceed hand-crafted systems in limited domains

• User-in-loop makes on-line learning feasible, even for previously unseen domains

• Distributed hierarchical models, with generic parameters and “committees of experts” enable systems to learn to expand coverage whilst avoiding unacceptable user experience.

• Focus today has been on expanding dialogue management. Current work suggests that similar ideas extend to SLU and NLG.
### CUED Dialogue Systems Group

<table>
<thead>
<tr>
<th>Current</th>
<th>Past</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steve Young</td>
<td>Blaise Thomson, Apple</td>
</tr>
<tr>
<td>Milica Gasic</td>
<td>Dongho Kim, Apple</td>
</tr>
<tr>
<td>David Vandyke</td>
<td>Matt Henderson, Google</td>
</tr>
<tr>
<td>Lina Rojas-Barahona</td>
<td>Prof Kai Yu, SJTU</td>
</tr>
<tr>
<td>Nikola Mrksic</td>
<td>Jason Williams, Microsoft</td>
</tr>
<tr>
<td>Eddy Su</td>
<td>Pirros Tsiakoulis, Innoetics Ltd</td>
</tr>
<tr>
<td>Shawn Wen</td>
<td>Francois Mairesse, Amazon</td>
</tr>
<tr>
<td><strong>Stefan Ultes</strong></td>
<td>Prof Filip Jurcicek, Charles U.</td>
</tr>
</tbody>
</table>

*starting Jan 2016*
Key strengths:
• automatic feature extraction
• ability to compactly encode sequence information

But hard to build a practical system without pulling out and explicit action set and without individually trainable modules.