Progress on the SPN Chatbot:
Handling compound and domain-specific queries

Zi Yin
Vin Sachidananda

(joint work with Prof. Prabhakar, Vig Sachidananda, Feiran Wang)
SPN Chatbot

Chatbot + UI provides an interactive and intelligent interface for:

- Troubleshooting
- Pattern Recognition
- Network Performance Analysis
- Monitoring
- Anomaly Detection
Voice Enabled Dashboard and Query Engine

User

Average uplink utilization in the last 5 mins for Top of Rack 6?

Dashboard

Conversational Agent

DB

Intent: “Link Statistics”
Entities: {'Average', 'Uplink', 'last', 300.0, ToR6}

SELECT AVG(utilization)
FROM Link_Statistics
WHERE time_adverb = 'last'
AND to_time = 300
AND direction = 'uplink'
AND link = ToR6
Compound Query (Language)

User

What about Top of Rack 5?

Dashboard

Conversational Agent

DB

Intent: “Link Statistics”
Entities: {'Average', 'Uplink', 'last', 300.0, ToR5}

```
SELECT AVG(utilization)
FROM Link_Statistics
WHERE time_adverb = 'last'
AND to_time = 300
AND direction = 'uplink'
AND link = ToR5
```
Compound Queries (Language + UX)

Intent: “Link Statistics”
Entities: {'Average', 'Uplink', 'last', 300.0, ToR5}
Compound Queries

- Compound Queries use information from preceding queries + UI interactions.

- For complex systems such as SPNs, users may take >15 queries to troubleshoot and monitor a network issue.

- Compound query understanding significantly reduces time/effort for user and enables longer (multi-turn) conversations.
Multi-Modal Dialogue State Tracking

Query: “Average uplink utilization in the last 5 mins for Top of Rack 6?”
Visual Context: None

\[ S_0 = \{ \text{time-adverb} = \text{last}, \] 
\[ \text{to-time} = 300, \]
\[ \text{direction} = \text{uplink}, \]
\[ \text{link} = \text{ToR6} \} \]

Extract a “state” for the conversation at the current time.
Multi-Modal Dialogue State Tracking

User: "What about Top of Rack 5?"

Query: "What about Top of Rack 5?"
Visual Context: None

Dashboard:

Conversational Agent: Intent Changed: False
$S_1' = \{\text{link = ToR5}\}$
$S_1 = S_1' \cup (S_0 \setminus S_1')$

Update the state based on new query information!
Multi-Modal Dialogue State Tracking

User: "This one?"

Visual Context: <'link': ToR5>

Query: “This one?”

Dashboard:

Intent Changed: False

\[ S'_1 = \{ \text{link} = \text{ToR5} \} \]

\[ S_1 = S'_1 \cup (S_0 \setminus S'_1) \]

Update the state based on new query + visual information!
Compound Query Understanding Demo

https://www.youtube.com/watch?v=gx53gWPqyBQ
## When is query understanding improved?

<table>
<thead>
<tr>
<th>Text Input</th>
<th>Visual Context</th>
<th>Predicted Intent</th>
<th>Predicted Intent w/ Vis. Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are some statistics?</td>
<td>Hovered on specific link</td>
<td>Large queues</td>
<td>Query Link Statistics</td>
</tr>
<tr>
<td>show me summary for this</td>
<td>Mouse previously on queue</td>
<td>General Summary</td>
<td>Queue Summary</td>
</tr>
<tr>
<td>How about this</td>
<td>Mouse on specific server</td>
<td>Help</td>
<td>Pick Server</td>
</tr>
<tr>
<td>Which of these has the most alerts</td>
<td>Hovered on a link</td>
<td>Most Alerts</td>
<td>Most Utilized Links</td>
</tr>
</tbody>
</table>
Domain Specific Word Embeddings

• Meanings of a word can be different in different domains
  • Also as know as the polysemy of a word

• Some examples in the networking domain

<table>
<thead>
<tr>
<th></th>
<th>Meaning in non-networking domains</th>
<th>Meaning in networking domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>line</td>
<td>A long, narrow mark or band</td>
<td>An interconnection cable between devices</td>
</tr>
<tr>
<td>queue</td>
<td>A sequence of people or vehicles awaiting their turns</td>
<td>A FIFO data structure</td>
</tr>
<tr>
<td>jitter</td>
<td>A sense of panic or extreme nervousness</td>
<td>An irregular pulse or signal</td>
</tr>
<tr>
<td>vertex</td>
<td>The highest point; the top or apex</td>
<td>A node in a graph</td>
</tr>
</tbody>
</table>
Domain-Specific Polysemy Issues

• Specific word usage cannot be captured by generic models
  • Buffer overflow, line rate, ...
  • Because these words are seldomly used in this way, a model built on a generic domain will think these usages are unlikely

• Generic speech recognition models will underperform
  • “What is the size of the largest cue?”
  • ...

Anchor Method for Detecting Domain Adaptation

- Triangulate a word against a bunch of “anchor” words
- If the relative position of a word has shifted across the two corpora, it is likely that the meanings of the word are different in the two domains
The Algorithm

- A vocabulary of $n$ words, common between domain $E$ and domain $F$
- The embedding for the $i$th word is $e_i$ in domain $E$, and $f_i$ in domain $F$
- Calculate the difference between anchor vectors,
  \[ d_i = \|e_iE^T - f_iF^T\| \]
- Sort the differences and pick the top $k$ candidates
Anchor Methods for Networking Domain

• We applied the anchor method to find the words with networking-specific meanings

• Some top candidates picked by the anchor methods:
  • Lawrence (person name vs national lab)
  • bug (insect vs error in program)
  • vertices (highest point vs node in a graph)
  • northbound (traffic towards north vs traffic outside data center)
  • jitter (panic vs irregular pulse)
  • ms (name suffix vs abbreviation for Microsoft)
Implications of Domain Adaptation

- Because word embeddings are used in many NLP models (i.e. chatbot), it’s important to have the correct word meanings!
Future Work