Towards Ontology-Independent Dialogue State Tracking

Michael Heck

Dialog Systems and Machine Learning
INTRODUCTION
Introduction

- Task-oriented dialogue systems (DS) – virtual assistants – gained increased popularity and acceptance over the years
  - Accomplish tasks such as bookings, searches, management, ...

- DS need to support a wide variety of domains
  - Recent work focused on scalable multi-domain DS

- Data-driven deep learning based approaches improved system quality considerably
  - Shift from discrete to continuous representations of concepts
Statistical dialogue systems 101

User input

„I’m looking for a restaurant“

Semantic decoding

State tracker

Dialogue management

System response

„What kind of food do you have in mind?“

Inform(type=restaurant)

Request(food)

Ontology

<table>
<thead>
<tr>
<th>Domains</th>
<th>Slots</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>restaurant</td>
<td>food</td>
<td>Italian</td>
</tr>
<tr>
<td>restaurant</td>
<td>area</td>
<td>centre</td>
</tr>
<tr>
<td>restaurant</td>
<td>price</td>
<td>cheap</td>
</tr>
<tr>
<td>taxi</td>
<td>depart</td>
<td>station</td>
</tr>
<tr>
<td>taxi</td>
<td>arrive</td>
<td>hotel</td>
</tr>
</tbody>
</table>

... ... ...
Introduction

Dialogue state tracking

- Dialogue state: Summary of the conversation till current turn
  - Set of constraints, for example **slot-value pairs**
- Dialogue state tracking: Update dialogue state at each turn
  - Required to determine next system action

### Ontology

<table>
<thead>
<tr>
<th>Turn</th>
<th>Domain-slot pair</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>restaurant-pricerange</td>
<td>expensive</td>
<td>span</td>
</tr>
<tr>
<td>0</td>
<td>restaurant-area</td>
<td>center</td>
<td>span</td>
</tr>
<tr>
<td>1</td>
<td>restaurant-food</td>
<td>&lt;dontcare&gt;</td>
<td>(dontcare)</td>
</tr>
<tr>
<td>2</td>
<td>restaurant-name</td>
<td>fitzbillies</td>
<td>informed</td>
</tr>
<tr>
<td>3</td>
<td>restaurant-people</td>
<td>5</td>
<td>span</td>
</tr>
<tr>
<td>3</td>
<td>restaurant-book_time</td>
<td>11:30</td>
<td>span</td>
</tr>
<tr>
<td>3</td>
<td>restaurant-book_day</td>
<td>tuesday</td>
<td>span</td>
</tr>
<tr>
<td>4</td>
<td>restaurant-internet</td>
<td>&lt;true&gt;</td>
<td>(bool)</td>
</tr>
<tr>
<td>5</td>
<td>hotel-area</td>
<td>center</td>
<td>coreference</td>
</tr>
</tbody>
</table>
Introduction

Statistical dialogue systems 101

- Discrete representation of concepts limits capacities
Continuous representations in DS

Vector representations mitigate semantic decoding problem

- Similarity measures replace exact matching

Mrksic et al., 2017, Neural Belief Tracker - Data-Driven Dialogue State Tracking
Introduction

Ontology-independent DS

- "I’m looking for a restaurant"
- "Does the user look for a restaurant?"
- "Does the user look for a hotel?"

User input

System response

Answer generation

Dialogue management

Integrated decoder & State tracker

Semantic conditioning

- Conditioning with natural language replaces fixed ontology
  - Measure semantic similarity between input and concepts
DEEP LEARNING BASED DST
Deep learning based DST

- Achieves state-of-the-art performance in DST evaluations
  - Utilization of **semantic representations** is driving force
    - Leverages semantic similarity of concepts (slots, values, etc.)
    - Representation of previously unseen concepts is possible
    - Tighter integration of DS components

- Picklist based
  - DS as distribution over all possible slot-values
  - Individual scoring of all slot-value pairs

Henderson et al., 2014, Word-based dialog state tracking with recurrent neural networks
Wen et al., 2017, A network-based end-to-end trainable task-oriented dialogue system
Mrksic et al., 2017, Neural Belief Tracker - Data-Driven Dialogue State Tracking
Ramadan et al., 2018, Large-Scale Multi-Domain Belief Tracking with Knowledge Sharing
Deep learning based DST

Picklist-based DST

What food would you like?  I'd like Thai food

System Utterance  User Utterance  Ontology

Input encoders produce vector representations

- ✓ Robustness due to semantic representations
- ✓ Knowledge sharing across domains
- X Limited scalability

Ramadan et al., 2018, Large-Scale Multi-Domain Belief Tracking with Knowledge Sharing
Deep learning based DST

- Picklist based
  - DS as distribution over all possible slot-values
  - Individual scoring of all slot-value pairs

- Span based
  - Find values through span matching in dialogue context

Gao et al., 2019, Dialog state tracking: A neural reading comprehension approach
Chao and Lane, 2019, BERT-DST: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer
Kim et al., 2019, Efficient dialogue state tracking by selectively overwriting memory
Deep learning based DST

Span-based DST

- Transformer produces contextual representations of input
  - Sentence representation used to determine presence of value
  - Token representations used to determine value span
- \( \times \) Limited to extractive values

Chao and Lane, 2019, BERT-DST: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer
Deep learning based DST

- **Picklist based**
  - DS as distribution over all possible slot-values
  - Individual scoring of all slot-value pairs

- **Span based**
  - Find values through span matching in dialogue context

- **Hybrid**
  - Combine picklists with span prediction

Zhang et al., 2019, Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking
Deep learning based DST

Hybrid approaches

- Similarity matching with candidates in picklist, or span pred.
- Slot name (and domain name) as part of input

Zhang et al., 2019, Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking
SCHEMA-GUIDED PARADIGM
Schema-guided paradigm

Reality check

- Current evaluations don’t fully capture reality of scenarios
  - Few domains, one service per domain, static ontologies

- Many domains, many services (defined by APIs)
  - Mismatch of training and testing conditions

---

User

Find direct round trip flights from Baltimore to Seattle.

Sure, what dates are you looking for?

Flying out May 16 and returning May 20.

OK, I found a Delta flight for 302 dollars.

System

FindFlight:
- depart = Baltimore
- arrive = Seattle
- direct_only = True

Flight Service A

SearchFlight:
- origin = Baltimore
- destination = Seattle
- num_stops = 0

Intents:
- SearchFlight,
- ReserveFlight

Slots:
- origin, destination, num_stops, depart, return, ...

Flight Service B

SearchFlight:
- origin = Baltimore
- destination = Seattle
- num_stops = 0
- depart = May 16
- return = May 20

Intents:
- FindFlight,
- ReserveFlight

Slots:
- depart, arrive, depart_date, return_date, direct_only, ...
Shortcomings of recent systems

- Recent systems parse dialogues in terms of **fixed concepts**
  - Lack understanding of the **semantics** of concepts
- Example: “I want to buy tickets for a movie.”
  - Models predict “BuyMovieTickets” based on observed patterns
    - No association with real action of buying movie tickets
    - Similarity to action of buying concert tickets not captured
- Models not robust to changes
  - Need to be retrained as new slots or intents are added
  - Domain-specific parameters unsuitable for zero-shot application
Schema-guided paradigm

Challenges of building large-scale systems

- Support of heterogenous services/APIs
  - Might overlap in functionality

- Robustness towards changes in API
  - Robustness towards new slots and intents
  - Generalization to new slot values (with little or no retraining)

- Generalization to new APIs
  - Joint modelling across APIs
  - Zero-shot generalization
Schema-guided paradigm

Approaches to related problems

- Adaptation and transfer learning for Slot-filling for DST
- Parameter sharing for domain adaptation and joint training

Zhang et al., 2019, Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking
Approaches to related problems

- Zero-shot learning for Slot-filling for DST
  - Infusing semantic slot representations into unified model

Bidirectional LSTM

Book a table at Heinemann on Tuesday ...

Bapna et al., 2017, Towards Zero-Shot Frame Semantic Parsing for Domain Scaling
Shah et al., 2019, Robust Zero-Shot Cross-Domain Slot Filling with Example Values
Approaches to related problems

- Zero-shot learning for Slot-filling for DST
  - Infusing semantic slot representations into unified model

Bapna et al., 2017, Towards Zero-Shot Frame Semantic Parsing for Domain Scaling
Shah et al., 2019, Robust Zero-Shot Cross-Domain Slot Filling with Example Values
Schema-guided paradigm for dialogue modeling

- Each **service** provides a **schema**
  - Lists supported slots and intents
  - Provides natural language descriptions for schema elements

**Figure:** Example schema for a service called "payment".

Rastogi et al., 2020, Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset
Advocates building a single **unified** dialogue model for all services and APIs using **semantic conditioning**

- A model should not contain service specific components

**A service’s schema serves as input to the model**

- Uses descriptions to obtain **semantic representations** of schema elements
- Predictions are **conditioned** on semantics of schema
- Predictions over dynamic sets of intents and slots

- A model should generalize to unseen services, APIs, concepts
Schema-guided paradigm for dialogue modeling

- Zero-shot learning by using semantic modeling
  - Knowledge sharing by ...
    - ... relating semantically similar concepts
    - ... using single unified model
  - Handling of unseen services and API changes by using
    - natural language input
    - semantic representations to condition the model
SCHEMA-GUIDED DST
Schema-guided DST track at DSTC8

SGD Dataset

Benchmark highlighting challenges for large-scale systems

<table>
<thead>
<tr>
<th>Domains</th>
<th>Slots</th>
<th>Values</th>
<th>Dialogues</th>
<th>Avg. turns per dialogue</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSTC2</td>
<td>WOZ2.0</td>
<td>FRAMES</td>
<td>M2M</td>
<td>MultiWOZ</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>61</td>
<td>13</td>
<td>30</td>
</tr>
<tr>
<td>212</td>
<td>99</td>
<td>3,871</td>
<td>138</td>
<td>4,510</td>
</tr>
<tr>
<td>1,612</td>
<td>600</td>
<td>1,369</td>
<td>1,500</td>
<td>8,438</td>
</tr>
<tr>
<td>14.49</td>
<td>7.45</td>
<td>14.60</td>
<td>9.86</td>
<td>13.46</td>
</tr>
</tbody>
</table>

Table: Statistics of training portions of datasets

<table>
<thead>
<tr>
<th>Domain</th>
<th>Services</th>
<th>Domain</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm</td>
<td>1</td>
<td>Movies</td>
<td>3</td>
</tr>
<tr>
<td>Banks</td>
<td>2</td>
<td>Music</td>
<td>3</td>
</tr>
<tr>
<td>Buses</td>
<td>3</td>
<td>Payment</td>
<td>1</td>
</tr>
<tr>
<td>Calendar</td>
<td>1</td>
<td>RentalCars</td>
<td>3</td>
</tr>
<tr>
<td>Events</td>
<td>3</td>
<td>Restaurants</td>
<td>2</td>
</tr>
<tr>
<td>Flights</td>
<td>4</td>
<td>RideSharing</td>
<td>2</td>
</tr>
<tr>
<td>Homes</td>
<td>1</td>
<td>Services</td>
<td>4</td>
</tr>
<tr>
<td>Hotels</td>
<td>4</td>
<td>Train</td>
<td>1</td>
</tr>
<tr>
<td>Media</td>
<td>3</td>
<td>Travel</td>
<td>1</td>
</tr>
<tr>
<td>Messaging</td>
<td>1</td>
<td>Weather</td>
<td>1</td>
</tr>
</tbody>
</table>

Table: Domains and services in SGD dataset

Slot types:

- **Non-categorical**: set of possible values is unrestricted
  - Eval sets contains unseen values
- **Categorical**: possible values are pre-defined and fixed
Schema-guided DST

Baseline: Zero-shot dialogue state tracking

- Model is shared among all services and domains

- Uses 2 contextual encoders:
  - Finetuned BERT encodes context
  - Fixed pre-trained BERT encoding schema element descriptions
    - Intents, slots, categorical slot values

- Schema element-wise classification
  - Concat. context representation and schema element represent
  - Do for each turn and for each schema element

Rastogi et al., 2020, Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset
Schema-guided DST

Baseline: Zero-shot dialogue state tracking

<table>
<thead>
<tr>
<th>Dialogue Context Encoding Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
</tr>
<tr>
<td>System Utterance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Schema Encoding Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
</tr>
<tr>
<td>Intent</td>
</tr>
<tr>
<td>Slot</td>
</tr>
<tr>
<td>Value</td>
</tr>
</tbody>
</table>
Schema-guided DST

Evaluation metrics

- **Joint goal accuracy**
  - Average accuracy of predicting all slot assignments correctly

- **Average goal accuracy**
  - Average accuracy of predicting a slot value correctly

- **Active intent accuracy**
  - Fraction of user turns for which intent was predicted correctly

- **Requested slot F1**
  - Average F1 score for requested slots
## Schema-guided DST

### Evaluation results

<table>
<thead>
<tr>
<th>All services</th>
<th>Seen services</th>
<th>Unseen services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint GA</td>
<td>Avg GA</td>
<td>Intent Acc</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.25</td>
<td>0.56</td>
</tr>
</tbody>
</table>

- **Drawbacks**
  - No history (only single turn) in context
    - Many slot values appear multiple turns earlier
  - Separate models for context and schema
    - Interaction only after encoding
    - No finetuning of schema encoder
Schema-guided DST

Unified span detection framework for SG-DST

- **Single** BERT to encode context and schema elements
  - To facilitate more interaction and utilize attention mechanism
  - Multiple passes per turn and slot, one for each prediction task
    - Intent, categorical slot, non-categorical slot

- Adds (truncated) dialogue history to input

- Render all predictions a span prediction problem
  - To utilize same model architecture for multitask learning effect

Shi et al., 2020, A BERT-based Unified Span Detection Framework for Schema-Guided Dialogue State Tracking
Schema-guided DST

Unified span detection framework for SG-DST

<table>
<thead>
<tr>
<th>Intent</th>
<th>Sequence 1</th>
<th>Sequence 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Categorical Slot</td>
<td>( \text{turn}_{t-k} \ [\text{SEP1}] \ldots \text{turn}_t )</td>
<td>[\text{INTENT}] Service Name [\text{SEP2}] Intent Name Intent Description</td>
</tr>
<tr>
<td>Non Categorical Slot</td>
<td>( \text{turn}_{t-k} \ [\text{SEP1}] \ldots \text{turn}_t )</td>
<td>[\text{NC_SLOT}] Slot Name [\text{SEP2}] Slot Name Slot Description</td>
</tr>
</tbody>
</table>
Schema-guided DST

Evaluation results

<table>
<thead>
<tr>
<th></th>
<th>All services</th>
<th>Seen services</th>
<th>Unseen services</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Joint GA</td>
<td>Avg GA</td>
<td>Intent Acc</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.25</td>
<td>0.56</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.52</td>
<td>0.89</td>
</tr>
<tr>
<td>Shi</td>
<td>0.54</td>
<td>0.8</td>
<td>0.91</td>
</tr>
</tbody>
</table>

- **Important details**
  - Uses BERT-large instead of BERT-base
  - Post-submission tests showed advantage of even longer history

- **Observations**
  - ✓ Very good generalization to new services
    - Authors attribute this to joint encoding of context and schema
  - ❌ Req. slot F1 significantly lower, reason unclear (not discussed)
Schema-guided DST

Goal-oriented multi-task BERT-based DST

- **Single** BERT to encode context and schema elements
  - **Single pass** per turn and slot, all predictions are done at once
    - Intent + Slot (request, categorical, non-categorical)
    - Special classification heads work in parallel

- Adds (truncated) dialogue history to input

- Strict input format
  - Special tokens and padding for partitioning

Gulyaev et al., 2020, Goal-Oriented Multi-Task BERT-Based Dialogue State Tracker
Schema-guided DST

Goal-oriented multi-task BERT-based DST

<table>
<thead>
<tr>
<th>Question</th>
<th>Input sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>Slot and service description</td>
</tr>
<tr>
<td>Possible intents</td>
<td>Dialogue history</td>
</tr>
<tr>
<td>Possible values</td>
<td>Descriptions of intents supported by the service</td>
</tr>
<tr>
<td></td>
<td>Possible slot values (for categorical slots only)</td>
</tr>
</tbody>
</table>
Schema-guided DST

Evaluation results

<table>
<thead>
<tr>
<th></th>
<th>All services</th>
<th></th>
<th>Seen services</th>
<th></th>
<th>Unseen services</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Joint GA</td>
<td>Avg GA</td>
<td>Intent Acc</td>
<td>Req Slot F1</td>
<td>Joint GA</td>
</tr>
<tr>
<td>Baseline</td>
<td>0,25</td>
<td>0,56</td>
<td>0,91</td>
<td>0,97</td>
<td>0,41</td>
</tr>
<tr>
<td>Gulyaev</td>
<td>0,46</td>
<td>0,75</td>
<td>0,75</td>
<td>0,97</td>
<td>0,53</td>
</tr>
<tr>
<td>Shi</td>
<td>0,54</td>
<td>0,80</td>
<td>0,91</td>
<td>0,87</td>
<td>0,53</td>
</tr>
</tbody>
</table>

- Uses BERT-large (finetuned on Squad) instead of BERT-base

- Observations
  - ✅ Categorical slots as span prediction task boosts performance
  - Similarly, intent classification as span prediction boosts performance
  - ❌ Similar performance to (Shi), but lacks behind for intent acc.
  - Relies on token representations and span prediction
  - ❌ Struggles with domain switches, slot value transfers
Schema-guided DST

Fine-tuning BERT for schema-guided zero-shot DST

- 6 BERT fine-tuned models for prediction
  - Intent prediction
  - Slot prediction (Categorical, Free-form, Requested)
  - Slot transfer prediction (In-domain, Cross-domain)
  - Multiple passes: First Intent & Slot, then transfer prediction
- Adds (truncated) dialogue history to input
- Adds auxiliary context features to BERT input
  - Indicate if a value/intent was predicted in turn t-1
  - Indicate if a value was mentioned by the system

Ruan et al., 2020, Fine-Tuning BERT for Schema-Guided Zero-Shot Dialogue State Tracking
Schema-guided DST

Fine-tuning BERT for schema-guided zero-shot DST

Input for intent prediction model

Input for categorical slot prediction model

Input for free-form slot prediction model

Input for requested slot prediction model

Input for in-domain slot transfer model

Input for cross-domain slot transfer model
Evaluation results

<table>
<thead>
<tr>
<th></th>
<th>All services</th>
<th></th>
<th>Seen services</th>
<th></th>
<th>Unseen services</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Joint GA</td>
<td>Avg GA</td>
<td>Intent Acc</td>
<td>Req Slot F1</td>
<td>Joint GA</td>
<td>Avg GA</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.25</td>
<td>0.56</td>
<td>0.91</td>
<td>0.97</td>
<td>0.41</td>
<td>0.68</td>
</tr>
<tr>
<td>Gulyaev</td>
<td>0.46</td>
<td>0.75</td>
<td>0.75</td>
<td>0.97</td>
<td>0.53</td>
<td>0.74</td>
</tr>
<tr>
<td>Shi</td>
<td>0.54</td>
<td>0.8</td>
<td>0.91</td>
<td>0.87</td>
<td>0.53</td>
<td>0.75</td>
</tr>
<tr>
<td>Ruan</td>
<td>0.74</td>
<td>0.92</td>
<td>0.92</td>
<td>0.99</td>
<td>0.88</td>
<td>0.96</td>
</tr>
</tbody>
</table>

**Observations**

- ✓ Slot transfers significantly improve performance
  - In-domain transfers constitute value references across multiple turns
  - Cross-domain transfers rely on reference resolution mechanism
- ✗ Joint GA drops considerably for unseen services
- ✓/✗ Adding dev set data to training has some positive effect
Schema-guided DST

Reading comprehension and wide & deep DST

- Reading comprehension model for non-categorical slots
  - Unrestricted input size
  - Adds entire dialogue history to input
- Wide & deep model for categorical slots
  - Transformer model output + hand-crafted features
- Data augmentation to vary schema element descriptions
  - Automatic generation via back-translations
- Joint model for intent and requested slot prediction
  - Classify dialogue context + intent/slot description

Ma et al., 2020, An End-to-End Dialogue State Tracking System with Machine Reading Comprehension and Wide & Deep Classification
Schema-guided DST

Reading comprehension and wide & deep DST

a. MRC model for span-based slot and numerical slot

b. Wide & Deep model for boolean and text-based slot
## Evaluation results

<table>
<thead>
<tr>
<th></th>
<th>All services</th>
<th>Seen services</th>
<th>Unseen services</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Joint GA</td>
<td>Avg GA</td>
<td>Intent Acc</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.25</td>
<td>0.56</td>
<td>0.91</td>
</tr>
<tr>
<td>Gulyaev</td>
<td>0.46</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>Shi</td>
<td>0.54</td>
<td>0.8</td>
<td>0.91</td>
</tr>
<tr>
<td>Ruan</td>
<td>0.74</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>Ma</td>
<td>0.87</td>
<td>0.97</td>
<td>0.95</td>
</tr>
</tbody>
</table>

### Important details:
- Hand-crafted features are rule and heuristic based (+10% JGA)
- Data augmentation by back-translation from Chinese (+6% JGA)
- Numerical slots are rendered non-categorical
- Partial delexicalization (phone numbers)
- Dev set used as additional training data
Summary & Analysis

<table>
<thead>
<tr>
<th></th>
<th>All services</th>
<th>Seen services</th>
<th>Unseen services</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Joint GA</td>
<td>Avg GA</td>
<td>Intent Acc</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.25</td>
<td>0.56</td>
<td>0.91</td>
</tr>
<tr>
<td>Gulyaev</td>
<td>0.46</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>Shi</td>
<td>0.54</td>
<td>0.8</td>
<td>0.91</td>
</tr>
<tr>
<td>Ruan</td>
<td>0.74</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>Ma</td>
<td>0.87</td>
<td>0.97</td>
<td>0.95</td>
</tr>
</tbody>
</table>

What worked?

- Approach: Reading comprehension + classification
  - Few submissions use a Baseline-style approach using similarity scoring
- Most systems exploit synergy effects from multitasking
- Maximizing context
  - Slot value reference resolution necessary across multiple turns
- Using hand-crafted features and additional data
## Summary & Analysis

<table>
<thead>
<tr>
<th></th>
<th>All services</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Seen services</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Unseen services</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Joint GA</td>
<td>Avg GA</td>
<td>Intent Acc</td>
<td>Req Slot F1</td>
<td>Intent Acc</td>
<td>Req Slot F1</td>
<td>Joint GA</td>
<td>Avg GA</td>
<td>Intent Acc</td>
<td>Req Slot F1</td>
<td>Joint GA</td>
<td>Avg GA</td>
<td>Intent Acc</td>
<td>Req Slot F1</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.25</td>
<td>0.56</td>
<td>0.91</td>
<td>0.97</td>
<td></td>
<td>0.41</td>
<td>0.68</td>
<td>0.95</td>
<td>0.97</td>
<td></td>
<td>0.2</td>
<td>0.52</td>
<td>0.89</td>
<td>0.95</td>
</tr>
<tr>
<td>Gulyaev</td>
<td>0.46</td>
<td>0.75</td>
<td>0.75</td>
<td>0.97</td>
<td></td>
<td>0.53</td>
<td>0.74</td>
<td>0.87</td>
<td>0.97</td>
<td></td>
<td>0.44</td>
<td>0.75</td>
<td>0.71</td>
<td>0.97</td>
</tr>
<tr>
<td>Shi</td>
<td>0.54</td>
<td>0.8</td>
<td>0.91</td>
<td>0.87</td>
<td></td>
<td>0.53</td>
<td>0.75</td>
<td>0.96</td>
<td>0.85</td>
<td></td>
<td>0.55</td>
<td>0.82</td>
<td>0.9</td>
<td>0.88</td>
</tr>
<tr>
<td>Ruan</td>
<td>0.74</td>
<td>0.92</td>
<td>0.92</td>
<td>0.99</td>
<td></td>
<td>0.88</td>
<td>0.96</td>
<td>0.96</td>
<td>1</td>
<td></td>
<td>0.69</td>
<td>0.91</td>
<td>0.91</td>
<td>0.99</td>
</tr>
<tr>
<td>Ma</td>
<td>0.87</td>
<td>0.97</td>
<td>0.95</td>
<td>0.98</td>
<td></td>
<td>0.92</td>
<td>0.98</td>
<td>0.96</td>
<td>0.99</td>
<td></td>
<td>0.85</td>
<td>0.97</td>
<td>0.95</td>
<td>0.98</td>
</tr>
</tbody>
</table>

- **What worked maybe?**
  - Specialized tags, input formatting, input processing
    - Benefits not investigated enough
  - Compartmentalizing: Specialized models for sub-tasks
    - Best systems employ multiple specialized encoders
    - Unified models are among most robust
DISCUSSION & CONCLUSION
Discussion

Mission accomplished?

- Multiple specialized models vs. unified models
  - What is the best use of semantic encoding?
    - Specialized representations for subtasks vs. generalized representations
    - Impact on architectures’ generalization capacities? Trade-off observable

- Engineering, heuristics, augmentation
  - Potence of auxiliary features demonstrates insufficiencies in semantic encoding. How to overcome limitations of encoders?

- Role of similarity measures
  - No exploration of spaces of contextual representations
  - Post-encoding similarity scoring not sufficiently explored
Conclusion

- Semantic conditioning of complex models is promising
  - Huge performance gain within single challenge iteration: 25% Joint GA -> 87% Joint GA!
  - Seemingly a convergence towards a „universal“ approach

- What next?
  - Zero-shot performance still not satisfactory
    - Reliance on tweaks to minimize gap
  - What if information about active service is not provided?
  - What if user does out-of-service requests?
    - DSTC9: Incorporating external non-dialogue knowledge sources
Select references

- Mrksic et al., 2017, Neural Belief Tracker - Data-Driven Dialogue State Tracking
- Henderson et al., 2014, Word-based dialog state tracking with recurrent neural networks
- Wen et al., 2017, A network-based end-to-end trainable task-oriented dialogue system
- Ramadan et al., 2018, Large-Scale Multi-Domain Belief Tracking with Knowledge Sharing
- Gao et al., 2019, Dialog state tracking: A neural reading comprehension approach
- Chao and Lane, 2019, BERT-DST: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer
- Kim et al., 2019, Efficient dialogue state tracking by selectively overwriting memory
- Zhang et al., 2019, Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking
- Bapna et al., 2017, Towards Zero-Shot Frame Semantic Parsing for Domain Scaling
- Shah et al., 2019, Robust Zero-Shot Cross-Domain Slot Filling with Example Values
- Rastogi et al., 2017, Scalable Multi-Domain Dialogue State Tracking
- Rastogi et al., 2020, Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset
- Rastogi et al., 2020, Schema-Guided Dialogue State Tracking Task at DSTC8
- Shi et al., 2020, A BERT-based Unified Span Detection Framework for Schema-Guided Dialogue State Tracking
- Gulyaev et al., 2020, Goal-Oriented Multi-Task BERT-Based Dialogue State Tracker
- Ruan et al., 2020, Fine-Tuning BERT for Schema-Guided Zero-Shot Dialogue State Tracking
- Ma et al., 2020, An End-to-End Dialogue State Tracking System with Machine Reading Comprehension and Wide & Deep Classification