Explicit State Tracking with Semi-supervision for Neural Dialogue Generation

Xisen Jin, Wenqiang lei, Zhaochun ren, Hongshen Chen, Shangsong Liang, Yihong Zhao, Dawei Yin

Outline

- Introduction & Motivation
- Method
  - Architecture: CopyFlow network
  - Training: Posterior regularization
- Experiments
- Conclusion

Dialogue systems

- Dialogue systems are receiving increasing attention in multiple applications.

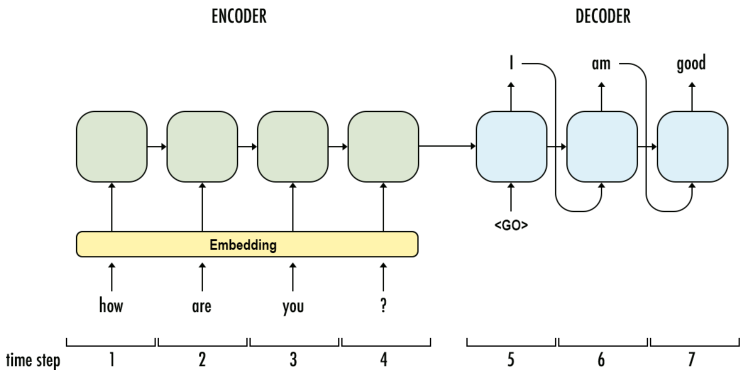
- Task oriented dialogue systems
  - Used for hotel booking, navigation, restaurant reservation and etc.
  - Retrieve a specific entity from a domain-specific KB and provide cohesive response
  - Interact with KB

- Non-task oriented dialogue systems
  - Forum question answering, chit chatting and etc.
  - Provide informative and cohesive response
  - All the domain knowledge is embedded in raw corpus
  - Trained on mass corpus(up to millions of dialogue turns)

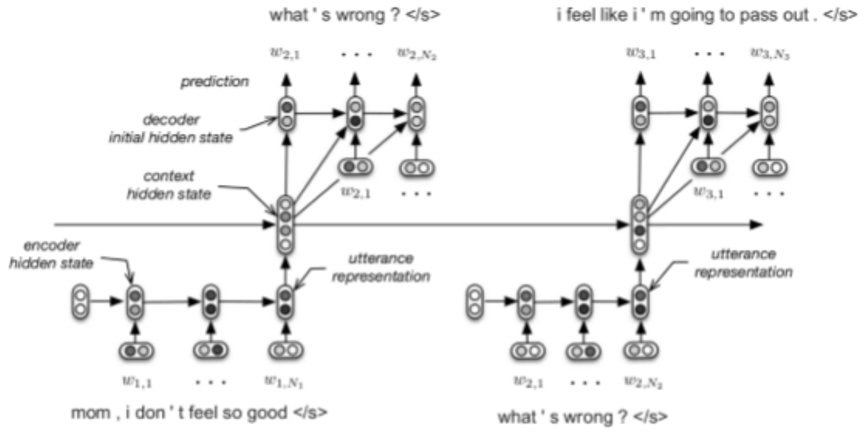
End-to-end training of dialogue systems

- End-to-end dialogue models can be implemented with RNNs: Seq2seq(Shang et al. 2015)

- Some variations of Seq2seq models
  - Hierarchical Encoder-Decoder(HRED)(Serban et al. 2015)
  - Sequicity Framework: End-to-end trainable modular dialogue system framework(Lei et al. 2018)



(a) Seq2Seq



(b) HRED

Dialogue state tracking

- A dialogue state refers to representation of user’ s intention up to current dialogue turn

- In task-oriented dialogue systems, dialogue state tracking is necessary since it is utilized for KB search
- Example training data for explicit dialogue states

User: Please find a moderately priced Italian restaurant

Slot	Value
Price	Moderate
Food type	Italian

- In non-task oriented dialogue systems, dialogue state tracking is helpful to generate context-aware and coherent responses
- Hardly have annotated data
- Usually implemented implicitly or with latent variables

Motivation - SEDST

- Current issue
  - Task oriented dialogue systems
    - Expensive state labeling
  - Non task oriented dialogue systems
    - Almost impossible state labeling
    - Implicit dialogue states are not capable for distinguishing similar concepts or entities(e.g. product names) in QA / transactional domain
    - Implicit dialogue states have poor interpretability

- **SEDST – semi-supervised explicit state tracking framework**
- Goal: Semi-supervised / Unsupervised explicit state tracking for task & non-task oriented dialogue systems

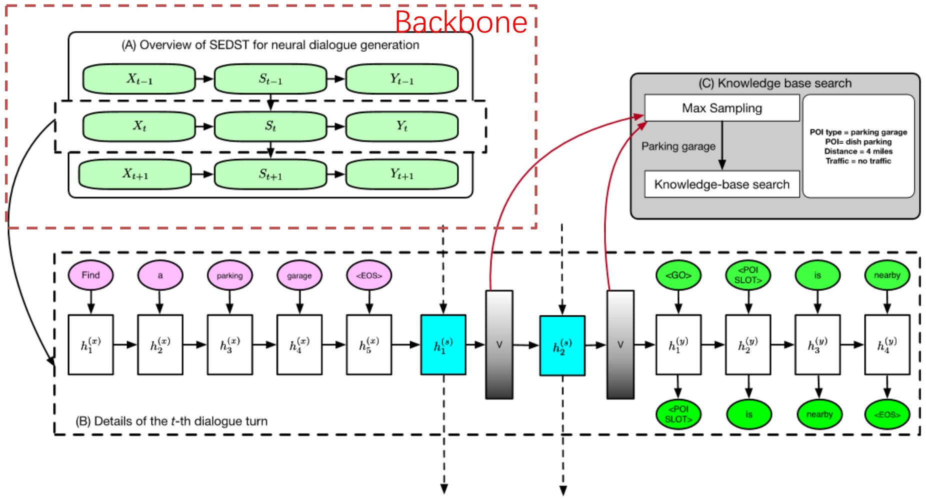
Outline

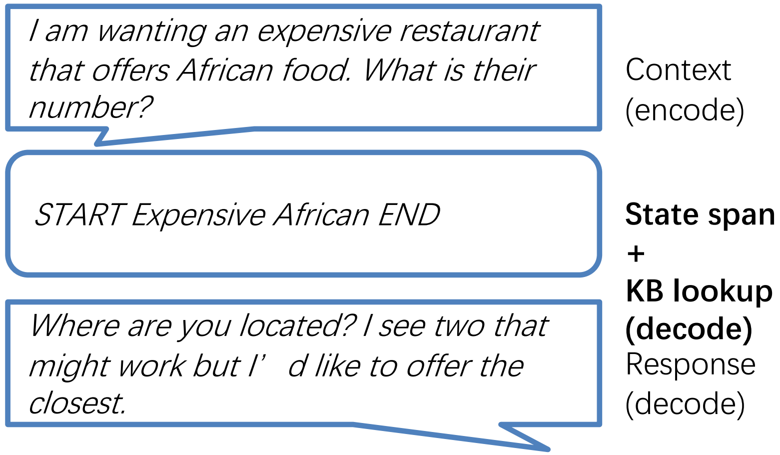
- Introduction & Motivation
- Method
  - Architecture: CopyFlow network
  - Training: Posterior regularization
- Experiments
- Conclusion

Backbone of SEDST

- **Copyflow network**

1. Input encoder
2. State span decoder
   - Decode dialogue states sequentially (Lei et al. 2018)
   - E.g: <inf> Italian <sep> moderate </inf>
3. Response decoder





Procedure of encoding and decoding in a dialogue turn

Model architecture – copyflow network

- Attention GRU encoder decoder
  - More details in the paper
- A "copyflow" from s to t:
  - Definition: Incorporating copying mechanism from *s* to *t*.
  - The probability of decoding a word is the sum of generation and copying probability
  - Generation probability $p_j^g = \frac{1}{Z} e^{w_3 h_j^{(y)}}$
  - Copying probability
    - "Hard" copy (Gu et al. 20 $p^{c(X)}(y_j) = \begin{cases} \frac{1}{Z} \sum_{i: w_{x_i} = y_j} e^{\psi(w_{x_i})}, y_j \in X \\ 0, otherwise \end{cases}$

      $w_{x_i} = y_j$: enable copying if the word exists in *s*
      $\psi(w_{x_i})$: score of copying the i-th positional word

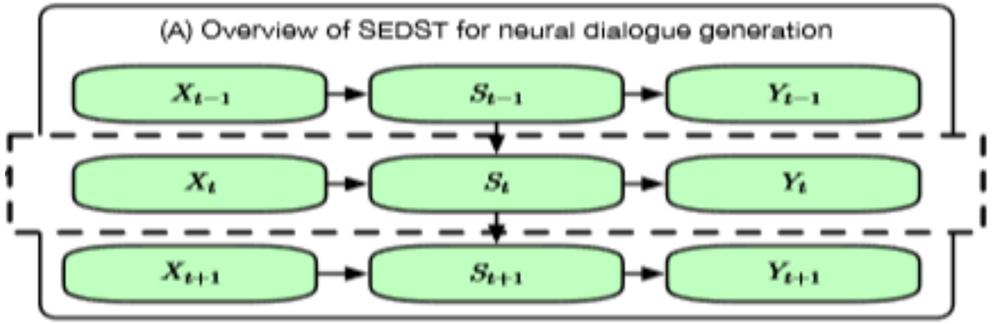
    - "Soft" copy (proposed) $p^{c(X)}(y_j) = \frac{1}{Z} \sum_{i=1}^{|X|} p_i(w_{x_i} = y_j) e^{\psi(w_{x_i})}$

      $p_u(w_{x_i} = y_j)$: the probability that the i-th word in source sequence is yj

Model architecture – copyflow network

- Inspired from the pattern that repeating of keywords could indicate dialogue states
- Co-occurrence may span over dialogue turns – keywords should be "copied" over dialogue turns
- From {previous responses, current inputs} to current state spans & From current state spans to current responses
  I. Enables the model to "cache" keywords in state spans
  II. Also possible to generate new words in state spans to further copy
- From previous state spans to current state spans
- The model learns to store information at state spans in the form of explicit word sequences

Role	Utterance
User	*I am wanting an expensive restaurant that offers African food. What is their number?*
Agent	*Where are you located? I see two that might work but I' d like to offer the closest.*
User	*I do not care about the area of town.*
Agent	*Bedouin is an expensive African restaurant in the city centre.*



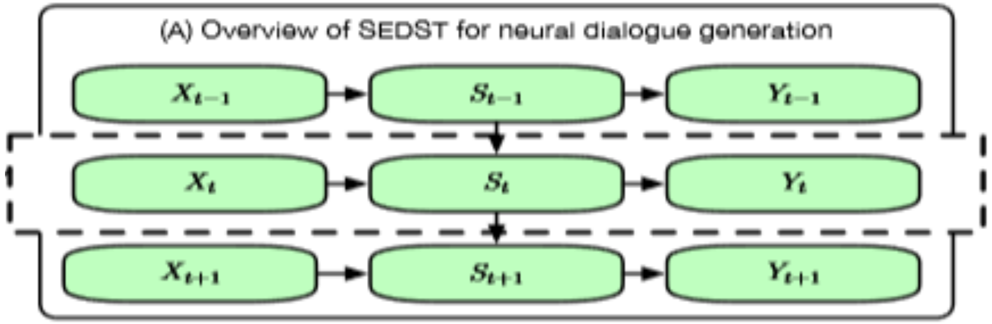
Training – Posterior regularization

- The training can be unstable without it
- Probabilistic interpretation of above-mentioned model
  - State span: $P_\Theta(S_t|R_{t-1}, S_{t-1}, U_t) = \Pi_i \mathbf{p}(s_t^{(i)}|s_t^{(<i)}, R_{t-1}, S_{t-1}, U_t)$.
  - Response generation: $P(R_t|R_{t-1}, U_t, S_t)$

- Additionally train a (helper) posterior network

$$Q_\Phi(S_t|\boxed{S_{t-1}, R_{t-1}, U_t, R_t}) = \Pi_i \mathbf{q}(s_t^{(i)}|s_t^{(<i)}, R_{t-1}, S_{t-1}, U_t, R_t)$$

Inputs for this model



Training – Posterior regularization

- Learning objective in semi-supervised scenarios for task-oriented datasets

$$\mathcal{L}_1 = -\sum^{\mathcal{A}\cup\mathcal{U}} log[P(R_t|R_{t-1}, U_t, S_t)]$$ Response generation loss

$$-\sum^{\mathcal{A}} log[P_\Theta(S_t|R_{t-1}, U_t, S_{t-1})Q_\Phi(S_t|R_{t-1}, U_t, S_{t-1}, R_t)]$$ Prior & Posterior state span generation loss

$$+\lambda \sum^{\mathcal{U}} \sum_{i=1}^{N} KL(\mathbf{q}_i||\mathbf{p}_i),$$ Regularization loss

- Interpretation:
  - Given limited data, the posterior network learns better than the prior network, since it is exposed to more inputs
  - The output of the prior network is forced to be close to that of the posterior network – "weak supervision"

Training – Posterior regularization

- Learning objective in unsupervised scenarios
- Have no annotated dialogue state data to train on for both prior and posterior network

- Method: Adjust the input and output of the posterior network as an auto-encoder
  - Learn to reconstruct the encoder input $R_{t-1}U_tR_t$ at its decoder
  - The model learns to cache keywords in $R_{t-1}U_tR_t$ into S_t

Training – Posterior regularization

- Learning objective in unsupervised scenarios

$$\mathcal{L}_2 = -\sum^{\mathcal{U}} log[P(R_t|R_{t-1}, U_t, S_t)]$$ Response generation loss

$$-\sum^{\mathcal{U}} log[Q_\Phi(R_{t-1}, U_t, R_t|\hat{S}_t)]$$ Reconstruction loss

$$+\lambda \sum^{\mathcal{U}} \sum_{i=1}^{N} KL(\mathbf{q}_i||\mathbf{p}_i) .$$ Regularization loss

- Interpretation:
  - The posterior network learns compacted representation of $R_{t-1}U_tR_t$ with a learning objective of autoencoder
  - Although the prior network can explore a generation strategy of the state span, it is regularized towards the posterior network.

Outline

- Introduction & Motivation
- Method
  - Architecture: CopyFlow network
  - Training: Posterior regularization
- Experiments
- Conclusion

Experiments

- Task-oriented Dataset
  - Cambridge Restaurant Reservation dataset(676 dialogues)
  - Stanford In-Car Assistant dataset(3029 dialogues)
- Non-task oriented Dataset
  - Ubuntu Dialogue corpus(487337 dialogues)
  - JD.com Customer Service corpus (425005 dialogues)
- Research Questions

  I. What is the overall performance of our model SEDST

  II. How much does unlabeled data help dialogue state tracking on task-oriented dialogues?

  III. Is our explicit state tracker helpful for non-task oriented response generation?

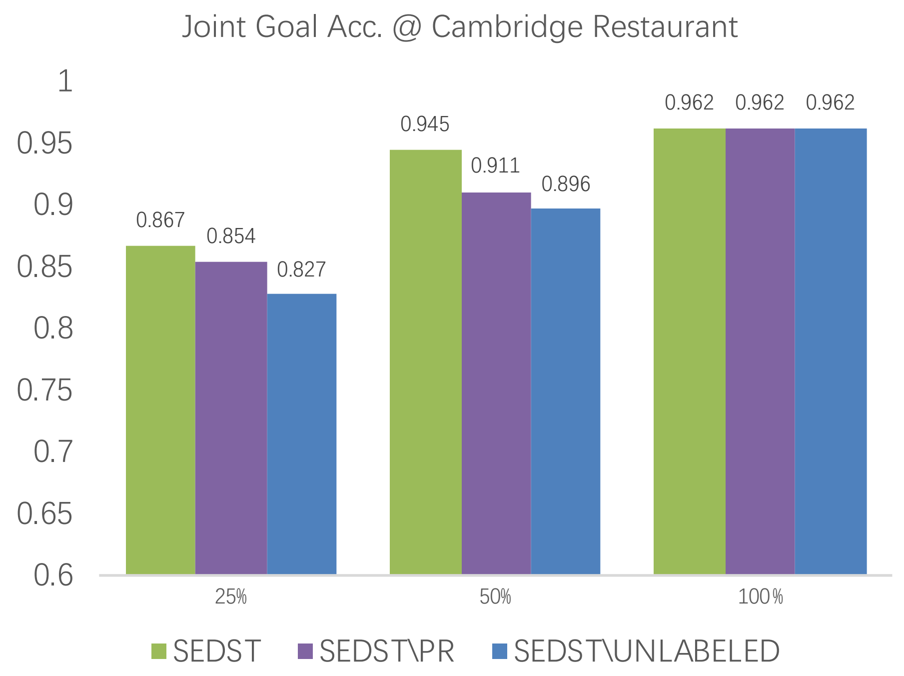
  IV. Does posterior regularization helps?

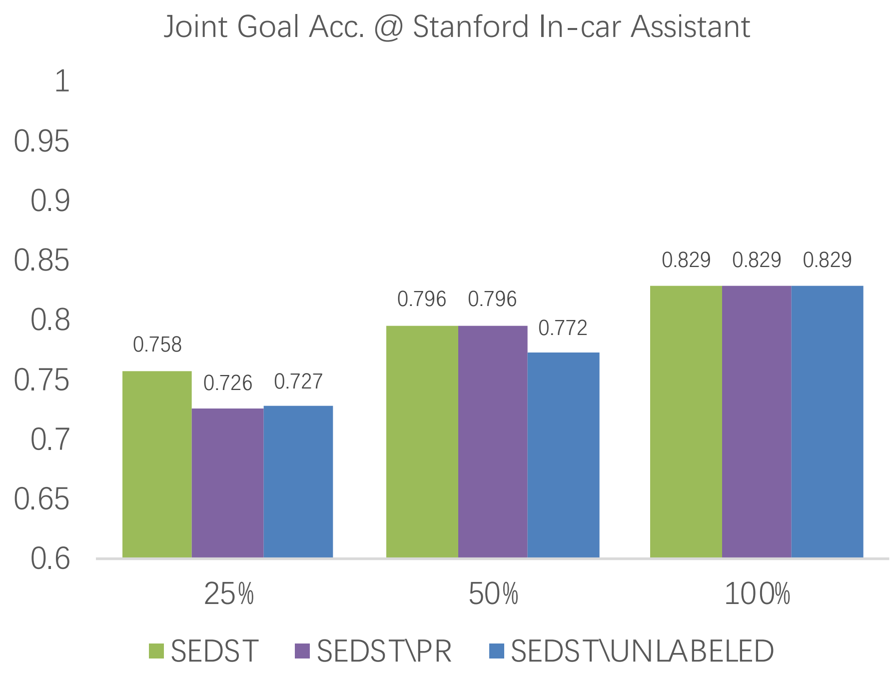
  V. Can SEDST generate interpretable state spans?

Results – task oriented

- Comparisons: SEDST, SEDST\PR(without posterior-regularization), SEDST\UNLABELED(only trained on labeled data)

- Metric 1: Joint Goal Accuracy(whether all the constraints are correct in a turn)

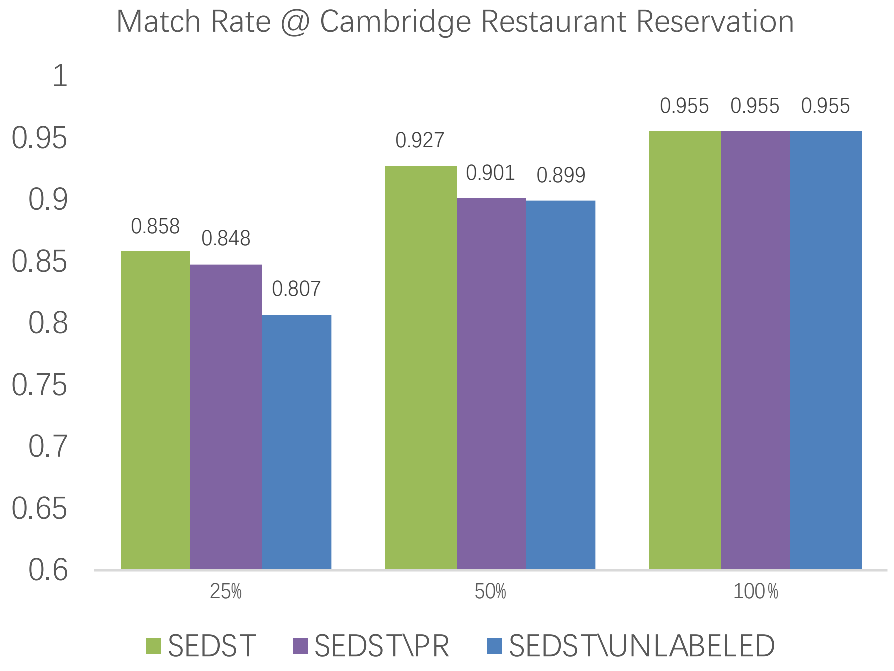


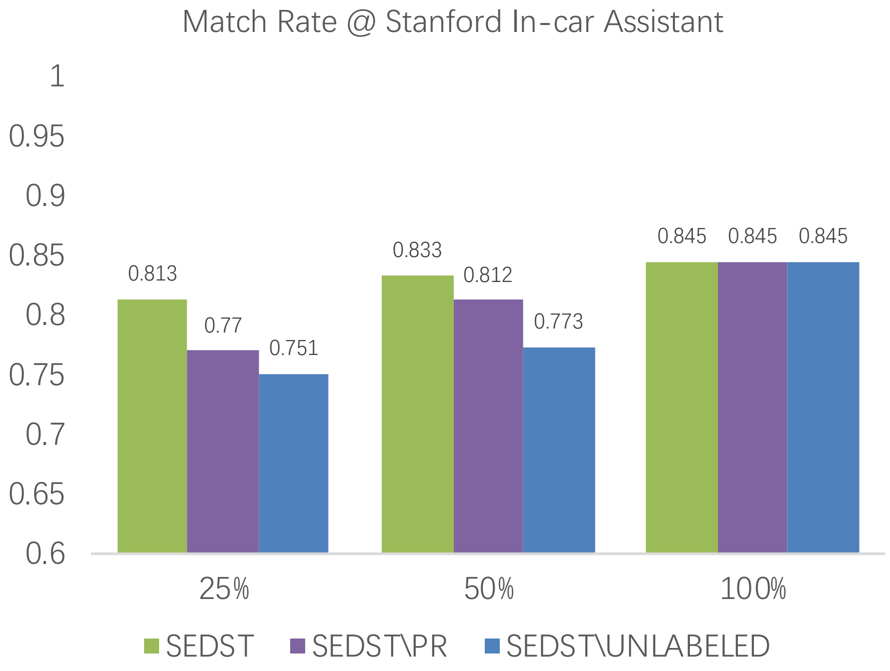


Results – task oriented

- Metric 2: Final Entity Match Rate(whether the state span in the final turn is correct)

- SEDST outperforms SEDST\UNLABELED: SEDST could utilize unlabeled data for learning
- SEDST outperforms SEDST\PR: Posterior regularization is effective





Results – Non task oriented

- Metric for response generation: Embedding Based Metrics
  - Embedding Average
  - Embedding Greedy
  - Embedding Extrema
- Specifically, our model outperforms VHRED and HVMN, which employs continuous latent variables for state tracking

Model	Emb. Average	Emb. Greedy	Emb. Extrema
SEQ2SEQ	0.216	0.169	0.126
HRED	0.542	0.412	0.319
VHRED	0.534	0.403	0.306
HVMN	0.558	0.423	0.322
DAWnet	0.530	0.390	0.333
SEDST\PR	0.586	0.438	0.330
SEDST	**0.609**	**0.451**	**0.337**

(a) Table: Embedding Metrics on Ubuntu Dialogue Corpus

Model	Emb. Average	Emb. Greedy	Emb. Extrema
SEQ2SEQ	0.425	0.479	0.264
HRED	0.549	0.587	**0.406**
VHRED	0.576	0.593	0.392
HVMN	0.564	0.596	0.405
DAWnet	0.579	0.574	0.375
SEDST\PR	0.575	0.602	0.373
SEDST	**0.585**	**0.607**	0.392

(b) Table: Embedding Metrics on JD.com Dialogue Corpus

Examples

Contexts	Generated	Ground Truth
I just realised nothing can access my mysql database anymore , it says cannot connect , why would this be ? I did set it to be remote	**State**: remote would anymore mysql access **Response**: you can use the mysql client to connect to the server	**State**: - **Response**: is it running ?
is it running ?	**State**: running I says mysql access **Response**: yes , I am running it now	**State**: - **Response**: How do I check ? I was able to type mysql commands
How do I check ? I was able to type mysql commands	**State**: commands mysql I running How **Response**: you can use sudo	**State**: - **Response**: thats not mean your server was running check with ps or top or others tool
thats not mean your server was running check with ps or top or others tool	**State**: mysql top tool server check **Response**: thanks	**State**: - **Response**: How do I check if it ’ s running ? Ah , k
How do I check if it ’ s running ? Ah , k	**State**:mysql check ps server thats **Response**: ps aux | grep mysql	**State**: - **Response**:ps aux | grep mysqld

Outline

- Introduction & Motivation
- Method
  - Architecture: CopyFlow network
  - Training: Posterior regularization
- Experiments
- Conclusion

Conclusion

- We have addressed the task of explicit dialogue state tracking for both task-oriented and non-task-oriented dialogue generation
  - For task-oriented dialogue systems, our model utilize unannotated data for learning.
  - For non-task-oriented dialogue systems, our model generate explicit dialogue states and improves response generation
- We verify the effectiveness of our model with intensive comparison and ablation study

- Future work: Transfer our model to other tasks (reading comprehension, summarization) / apply reinforcement learning to improve the state spans and response generation

Thanks!

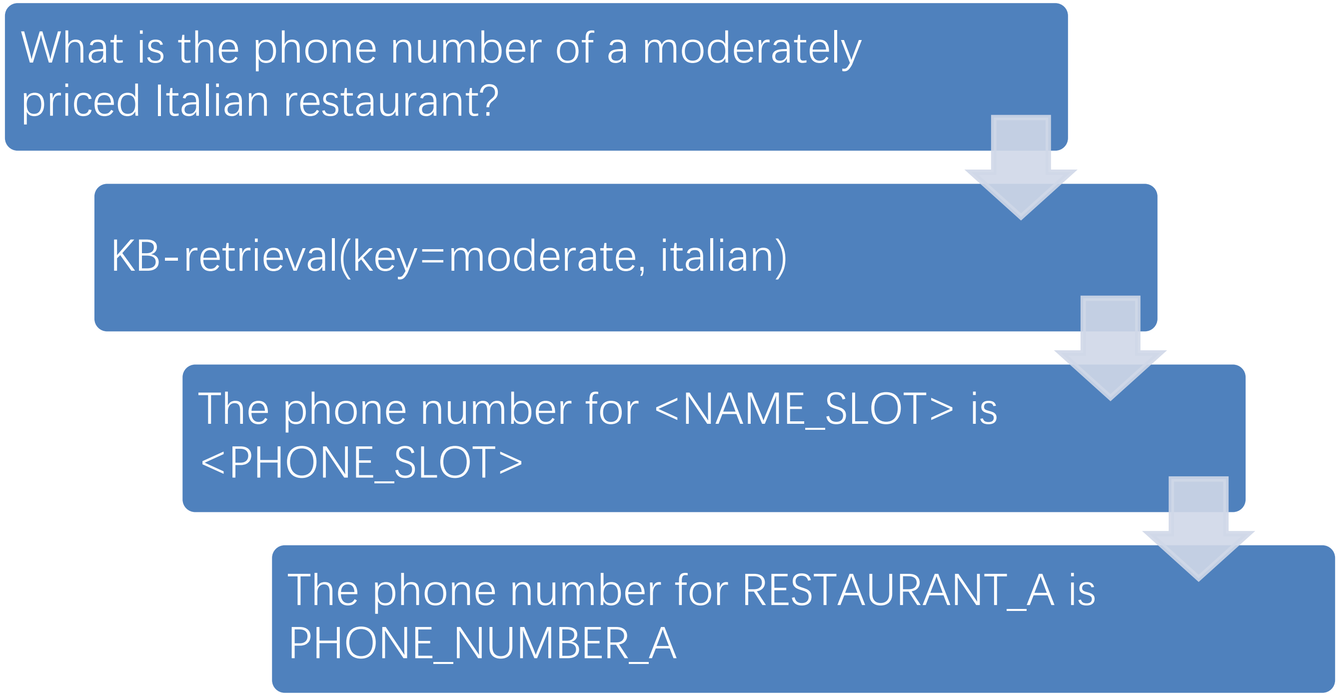
Thanks to SIGIR Student Travel Grant for travel reimbursement

Code available on:
https://github.com/AuCson/SEDST

Presenter Xisen Jin’ s homepage:
https://aucson.github.io

Details for KB search

- We follow the setting of Sequicity (Lei et al. 2018)
- The state spans are decoded without keyword and the queries are performed to all fields in a KB
- Entity type information can be obtained with a separate classifier or pre-defined table
- The retrieved results are utilized to fill in the placeholders in generated responses



Evaluation of state spans (non-task-oriented case)

- We compare with DAWNet(Wang et al. 2018), which first extract "predictive keywords" with unsupervised methods then train on them with supervised methods

- We calculate the proportion of the keywords that actually appear in the ground truth response

Model	Ubuntu Technical	Jd.com Customer Service
DAWnet	5.5%	32.6%
SEDST	14.7%	40.6%