# Prompt-based methods for Dialog State Tracking

## Thesis Presentation

Mandava, Sai Pavan

Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart

15.02.2023

# Outline

# Outline

# Introduction

- Task-oriented dialog systems
  - perform a wide range of tasks across multiple domains
  - *E.g. ticket booking, restaurant booking, etc.*
- Modular-based dialog systems
  - NLU, DST, PL, NLG

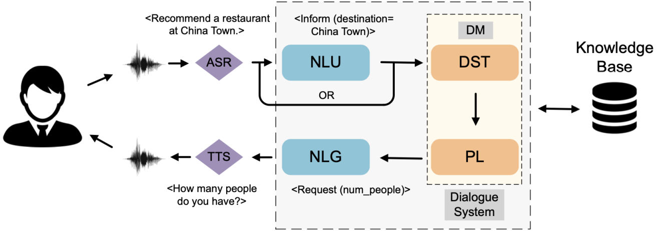



Fig: Modular-based task-oriented dialog system

# Dialog State Tracking (DST)

- Essential module for the dialog system to understand user's requests
- Tracks the user goals in the form of dialog states (or "belief states")
- Dialog states contains a set of *(slot, value)* pairs
  - Updated at each turn of the conversation

**DST Example**

**USER:** Plan a train trip to Berlin this Friday for two people.
**Belief states:** $\{(destination,\ Berlin),\ (day,\ Friday),\ (people,\ 2)\}$

- Ontology of domains
  - Represents knowledge & information required for specific tasks
  - Contains pre-defined set of slots and all possible values for each slot
  - Some Neural-based models solve the DST as classification task
- Problems with depending on ontology
  - Ontology is hard to obtain for new domains
  - Costly and time-consuming

# PLMs & Prompt Learning

- Pre-trained Language Models (PLMs)
  - Trained on large amounts of textual data
  - Encode linguistic knowledge into the huge amount of parameters
  - Can be efficiently used to solve NLP tasks
  - BERT(Devlin et al. 2019), GPT-2(Radford et al. 2019), GPT-3(Brown et al. 2020)
- Prompt Learning
  - New way of efficiently using the generation capabilities of PLMs to solve different language tasks
  - Downstream task is converted to a textual prompt and given as input, the PLM directly generates the outputs from prompts
  - Prompting methods can be effectively used under *zero-shot* and *few-shot* settings when there's not enough training data
  - GPT-3 (Brown et al. 2020), Few-shot Bot (Madotto et al. 2021), PET (Schick and Schütze 2021) explored prompt-based methods for several tasks

# Prompt Learning (contd.)

| Name | Notation | Example |
|---|---|---|
| *Input* | $x$ | I missed the bus today. |
| *Output* | $y$ | sad |
| *Prompt Function* | $f_{prompt}(x)$ | [$X$] I felt so [$Z$] |
| *Prompt* | $x'$ | I missed the bus today. I felt so [$Z$] |
| *Answered Prompt* | $f_{fill}(x', z)$ | I missed the bus today. I felt so sad |
| *Answer* | $z$ | *happy*, *sad*, *scared* |

Fig: Terminology and notations in prompt learning

- Prompt Types: *prefix* & *cloze* prompts
- Prompt selection: manual, discrete, & continuous prompts
- Training strategy: Fixed-prompt LM Fine Tuning
  - fixed prompts are applied to training data and fine-tune the LM
  - under low-resource few-shot settings

# Motivation & Objectives

- Previous work & their limitations
  - TOD-BERT (C.-S. Wu et al. 2020)
    - Pre-trained BERT on 9 different task-oriented datasets
    - Fine-tuned for DST task as multi-class classification
    - Depends on the ontology of domains for predicting slot-values
  - SOLOIST (Peng et al. 2021)
    - Pre-trained GPT-2 for two dialogue datasets
    - Fine-tuned to generate belief states as sequence of words
    - Performs poorly under low-resource settings
- Research Objectives
  - Can the dialog states be extracted from the PLM using prompts?
  - Can the prompt-based methods learn the DST task under low-resource settings without depending on the ontology of domains?
  - Compare prompt-based approach with the baseline model
  - Identify the drawbacks & limitations of prompt-based approach
  - Can different multi-prompt techniques help improve the performance of DST task?

# Outline

# Dataset - MultiWOZ (Budzianowski et al. 2018)

- MultiWOZ 2.1 (Eric et al. 2019) is used to benchmark the DST
- Contains huge number of dialogues across multiple domains
- Each Dialog $\rightarrow$ multiple turns $\rightarrow$ multiple *(slot,value)* pairs
- Five domains are picked for few-shot experiments
  - *Restaurant, Hotel, Attraction, Taxi, Train*
- Six data splits are created to perform few-shot experiments
  - Different proportions of dialogues in each split
  - All the five domains are evenly distributed in each split

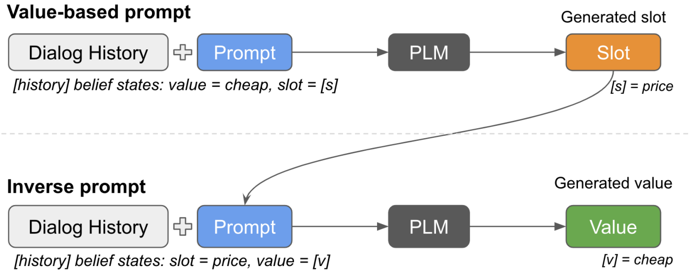
| Data Splits | # Dialogues | # Total Turns | # (slot, value) |
|---|---|---|---|
| *5-dpd* | 25 | 100 | 294 |
| *10-dpd* | 50 | 234 | 758 |
| *50-dpd* | 250 | 1114 | 3535 |
| *100-dpd* | 500 | 2292 | 7408 |
| *125-dpd* | 625 | 2831 | 9053 |
| *250-dpd* | 1125 | 5187 | 17214 |
| *valid* | 190 | 900 | 3106 |
| *test* | 193 | 894 | 3411 |

# Baseline (Soloist)

- Soloist (Peng et al. 2021) is the baseline model
- Initialized with 12-layer GPT-2 language model
- Pre-training step
  - Pre-trained on two task-oriented dialogue datasets
  - Pre-trained model is publicly available
- Fine-tuning step
  - Fine-tuned on all MultiWOZ 2.1 data splits to perform the belief predictions task
  - Takes dialog history as input and generates belief states as sequence of words
  - *belief:* $slot_1 = value_1; slot_2 = value_2, \ldots$

# Prompt-based methods

- Yang et al. 2022 proposed prompt learning framework for DST
- This approach doesn't depend on the ontology of domains
- Two components: *value-based prompt* and *inverse prompt*
- Value-based prompt uses belief state values in prompts and generates the slots from PLM
- Inverse prompt is an auxiliary task that uses the slot generated from value-based prompt and attempts to generate back the value.



# Prompt-based methods - Training

| Type | Prompt templates |
|---|---|
| value-based prompt | belief states: value = [v], slot = [s] |
| inverse prompt | belief states: slot = [s], value = [v] |

- The pre-trained Soloist is used to fine-tune the prompting methods
- All MultiWOZ data splits are used in the fine-tuning phase
- Loss function for value-based prompt

$$\mathcal{L} = -\sum_{t}^{|D|} \log P\left(s_t \mid c_t, f\left(v_t\right)\right)$$

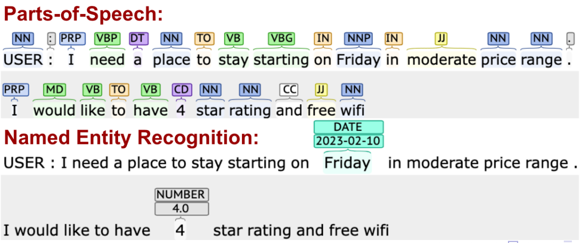
- Loss function for inverse prompt

$$\tilde{\mathcal{L}} = -\sum_{t}^{|D|} \log P\left(v_t' \mid c_t, I\left(s_t\right)\right)$$

- Total Loss: $\mathcal{L}^* = \mathcal{L} + w * \tilde{\mathcal{L}}$
  - Experiments are performed on different inverse prompt weights $w$

# Prompt-based methods - Testing

- Testing slot generation
  - During inference, only value-based prompts are used
  - Prompts are filled with values and given as input to PLM
  - Next word with the highest probability is the generated slot
  - Rule-based approach for extracting value candidates
- Rule-based Value Extraction:
  - Stanford CoreNLP Stanza is used to first extract POS tags
  - Adjectives (JJ) and Adverbs (RB) are considered as possible values
  - Consider the previous negator 'not'
  - Consider all named entities (name of place, time, day, numbers)
  - Custom Regex NER rules, filtered stop words and repeated values



# Multi-prompt method (Prompt Ensemble)

- Only a single value-based prompt is used in the previous experiments
- Multiple prompts can be used together to improve the performance
- Prompt Ensembling uses multiple value-based prompts during training and inference to take advantage of different prompts
- Four hand-crafted prompt templates for value-based prompt

| | Prompt ensemble templates |
|---|---|
| $f_1$ | belief states: [v] = [s] |
| $f_2$ | [v] is the value of [s] |
| $f_3$ | [v] is of slot type [s] |
| $f_4$ | belief states: value = [v], slot = [s] |

- A single model is trained with multiple prompts
- The probability of generated slot over multiple prompt functions:

$$P\left(s_t \mid c_t\right)=\sum_{k}^{|K|} \alpha_k * P\left(s_t \mid c_t, f_k\left(v_t\right)\right)$$

# Multi-prompt method (Prompt Augmentation)

- Provides a few additional answered prompts that can demonstrate to the PLM how the actual task can be performed
- Sample selection is manually hand-picked from training data
- Experiments are performed on two sets of demonstration samples
  - Sample set 1: 8 demonstrations
  - Sample set 2: 5 demonstrations
- Demonstrations are concatenated to the input during inference
- Number of demonstration examples that can be used is bounded by the GPT-2 max input length of 1024

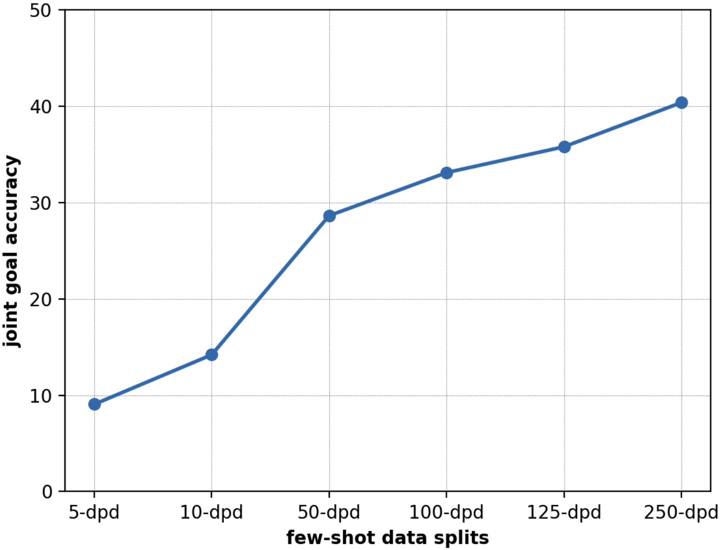
| **Demonstration learning** | |
|---|---|
| Book a cheap flight to Frankfurt. | *Frankfurt* is of slot *destination* |
| Plan a train trip to Berlin. | *Berlin* is of slot *destination* |
| Book a taxi to the University. | *University* is of slot *destination* |
| Book a train to Stuttgart. | *Stuttgart* is of slot [s] |

# Evaluation Metrics

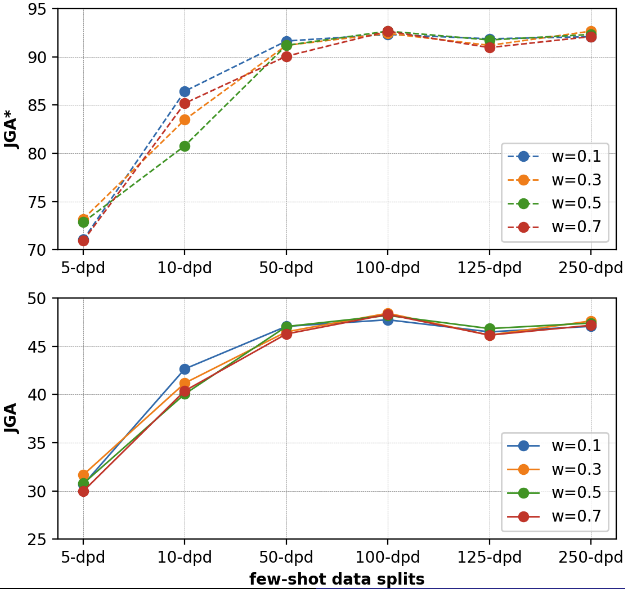
- Joint Goal Accuracy (JGA)
    - Standard evaluation metric for DST
    - Correct if all the predicted belief states match with the ground-truth
    - All the slots and values must exactly match
- Rule-based value extraction methods may extract irrelevant values
- JGA* (Yang et al. 2022)
    - To exclude the influence of wrongly extracted values, JGA* is used
    - JGA* - Joint Goal Accuracy is computed only for the belief states where the values are extracted correctly

# Outline

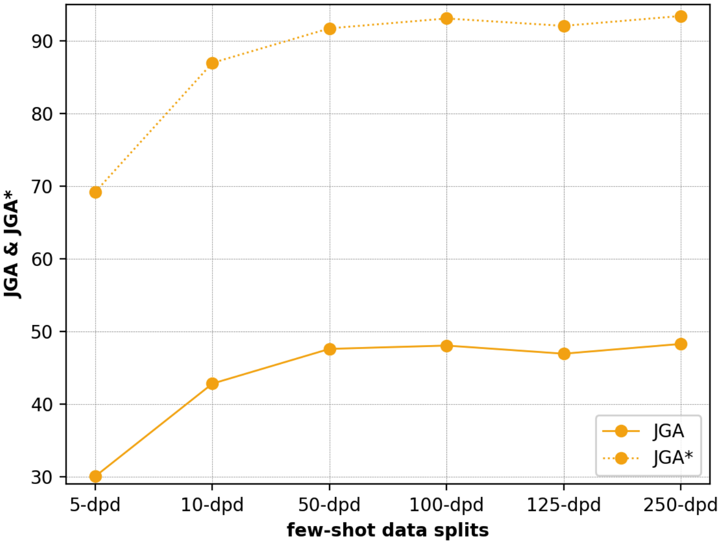
# Baseline (Soloist) results



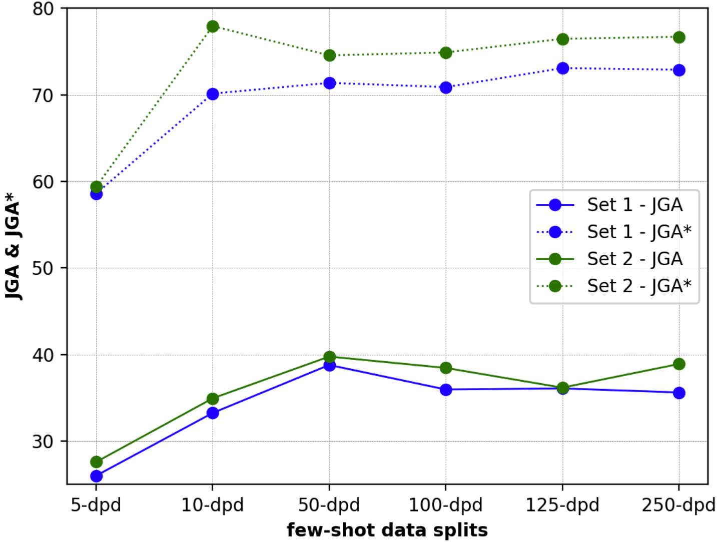
# Prompt-based methods



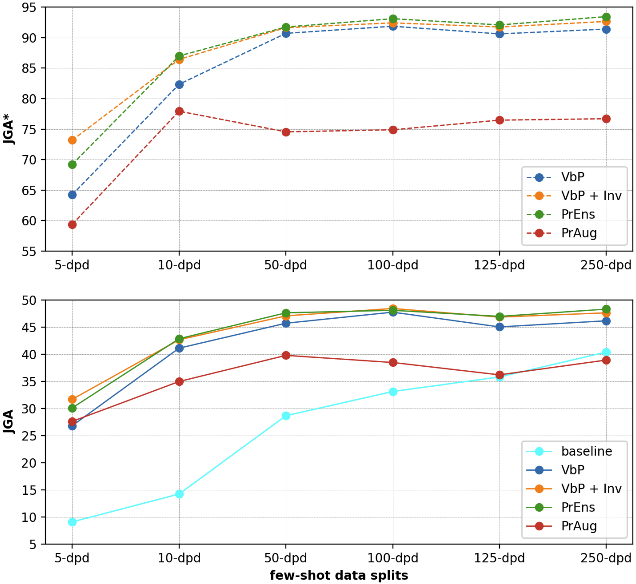
# Prompt Ensemble results



# Prompt Augmentation results



# Comparison of results



# Outline

# Analysis of SOLOIST model

## Example of wrong belief state prediction

USER: I need an expensive place to eat in the west.
SYSTEM: Is there a specific type of food you would like?
USER: yes, i would like eat indian food.
**True states:** (area, west),(food, indian),(pricerange, expensive)
**Generated:** *(area, west),(food, indian),(pricerange, cheap),(area, east)*

- Open-ended generation
- Susceptible to generating random slot-value pairs
- Repeated slot-value generations
- From the above example:
  - slot *area* is repeated with a different value
  - value for slot *pricerange* is incorrect

# Analysis of prompt-based methods

**Incorrect slot generations by value-based prompt**

USER: I need to be picked up from pizza hut city centre after 04:30
**True states:** (departure, pizza hut city centre), (leave, 04:30)
**Generated:** *(destination, pizza hut city centre), (arrive, 04:30)*

- Incorrect slots generated under low-resource splits (i.e., *5-dpd,10-dpd*)
- Model struggled to distinguish between slots:
  - *departure vs destination*
  - *leave vs arrive*
- Possibly due to limited training data

# Limitations of Value-based prompt

## Repeated Values in Belief States

USER: hi, can you help me find a 3 star place to stay?
SYSTEM: Is there a particular area or price range you prefer?
USER: how about a place in centre of town that is of type hotel.
SYSTEM: how long would you like to stay, and how many people?
USER: I'll arrive on saturday and stay for 3 nights with 3 people.
**True states:** (area, centre), (stars, 3), (type, hotel), (day, saturday), (stay, 3), (people, 3)

- User requirements may have repeated values in belief states
- Value for *stars*, *stay*, and *people* is the same
- Value-based prompt can only generate one slot for all the repeated values

# Error Analysis of Value Extraction

## Problems with Value Extraction

USER: I want a place to stay that has free wifi and free parking.
SYSTEM: do you have a preference for area or price range?
USER: I don't have a preference. I want a hotel not guesthouse.
**True states:** (area, <u>dont care</u>), (internet, <u>yes</u>), (parking, <u>yes</u>), (price, <u>dont care</u>), (type, hotel)
**Extracted Values:** *free*, *hotel*

---

USER: I kind of need help finding a nice hotel in the north part of town.
**True states:** (area, north), (price, expensive), (type, hotel)
**Extracted Values:** *kind*, *nice*, *hotel*, *north*

- Value Extraction on test split
  - Accuracy of *79%* on all the values
  - Turn-level accuracy of *49%*
- Drawbacks of extracting values from POS tags

# Outline

# Conclusion

- Prompt-based methods learned the DST task efficiently under low-resource few-shot settings without relying on the ontology.
- Prompt-based methods significantly outperformed the baseline SOLOIST model under low-resource settings.
- Some limitations in the prompt-based approach
- Prompt Ensemble model only achieved minor improvements over single value-based prompt
- Performance of Prompt Augmentation is limited due to insufficient demonstration examples

# Future work

- Explore automated prompt search methods for choosing the right prompts instead of manually creating the templates
- Improve the value extraction methods
  - Combination of text summarization and semantic tagging
- Can bigger language models perform better in prompting the DST task?

# References I


Brown, Tom et al. (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Budzianowski, Paweł et al. (Oct. 2018). "MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 5016–5026. DOI: 10.18653/v1/D18-1547. URL: https://aclanthology.org/D18-1547.

# References II


Devlin, Jacob et al. (June 2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Eric, Mihail et al. (2019). “MultiWOZ 2.1: Multi-Domain Dialogue State Corrections and State Tracking Baselines”. In: *CoRR* abs/1907.01669. arXiv: 1907.01669. URL: http://arxiv.org/abs/1907.01669.

Madotto, Andrea et al. (2021). “Few-Shot Bot: Prompt-Based Learning for Dialogue Systems”. In: *CoRR* abs/2110.08118. arXiv: 2110.08118. URL: https://arxiv.org/abs/2110.08118.

# References III

Peng, Baolin et al. (2021). "SOLOIST: Building Task Bots at Scale with Transfer Learning and Machine Teaching". In: *Transactions of the Association for Computational Linguistics* 9, pp. 807–824. DOI: 10.1162/tacl_a_00399. URL: https://aclanthology.org/2021.tacl-1.49.

Radford, Alec et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Schick, Timo and Hinrich Schütze (Nov. 2021). "Few-Shot Text Generation with Natural Language Instructions". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 390–402. DOI: 10.18653/v1/2021.emnlp-main.32. URL: https://aclanthology.org/2021.emnlp-main.32.

Wu, Chien-Sheng et al. (Nov. 2020). "TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 917–929. DOI: 10.18653/v1/2020.emnlp-main.66. URL: https://aclanthology.org/2020.emnlp-main.66.

Yang, Yuting et al. (2022). "Prompt Learning for Few-Shot Dialogue State Tracking". In: *CoRR* abs/2201.05780. arXiv: 2201.05780. URL: https://arxiv.org/abs/2201.05780.

*Thanks for your time!*