Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5b
70569 Stuttgart

Master Thesis Proposal

Prompt-based methods for Dialog State Tracking

27 July, 2022

Mandava, Sai Pavan
M.Sc. Computational Linguistics
Mat.Nr.: 3461015
st169661@stud.uni-stuttgart.de

Supervisor
Prof. Dr. Thang Vu
thang.vu@ims.uni-stuttgart.de

1 Motivation

Dialog State Tracking (DST) is an essential module in dialog systems, it tracks the user goals in the form of dialog states given the entire dialog history. In dialog systems, "dialog states" - sometimes also called "belief states" contains a set of *(slot, value)* pairs for each turn of the dialog history. Existing data-driven methods and neural models for individual dialog modules (NLU, DST, NLG) and end-to-end dialog systems show promising results, but they need large amounts of task-specific training data, which is rarely available for new tasks. For task-specific DST, collecting dialog state labels can also be costly, requiring domain experts to indicate all possible *(slot, value)* pairs for each turn of the dialogues. A typical task-oriented dialog system contains an ontology for each domain, with a pre-defined set of slots and all possible values for each domain. In real-world applications, defining all possible slots and values for DST is difficult due to new rising domains and users' continuous needs.

Prompt-based learning *("pre-train, prompt, and predict")* is a new paradigm in NLP which aims to predict the probability of text directly from the pre-trained LM (Liu et al., 2021). This framework is powerful as it allows the language model to be *pre-trained* on massive amounts of raw text, and by defining a new prompting function the model can perform *few-shot* or even *zero-shot* learning. The large pre-trained language models (PLMs) are supposed to be useful in *few-shot* scenarios where task-related training data is limited, as they can be "probed" for task-related knowledge efficiently by using a prompt. One example of such large pre-trained language models is GPT-3 (Brown et al., 2020) - *"Language Models are Few-Shot Learners"*. Madotto et al. (2021) created an end-to-end chatbot (Few-Shot Bot) using *prompt-based few-shot learning* (no gradient fine-tuning) and achieved comparable results to those of state-of-the-art. Prompt-based learning for few-shot DST with limited labeled domains is still under-explored.

Recently, Yang et al. (2022) proposed a new prompt learning framework for few-shot DST. This work designed a *value-based prompt* and an *inverse prompt* mechanism to efficiently train a DST model for domains with limited training data. This approach doesn't depend on the ontology of slots and the results show that it can generate unseen slots and outperforms the existing state-of-the-art few-shot methods. The goal of this thesis is to further explore this prompt-based few-shot learning framework for DST by implementing these three tasks: (1) Prompt learning framework for few-shot DST - reproduce the results from Yang et al. (2022). Can the DST knowledge be probed from PLM? (2) Evaluation and analyses of belief state predictions. This task will answer what improvements can be observed from prompt-based methods and the drawbacks of this approach. (3) Extend this prompt-based DST framework to utilize various *multi-prompt* learning methods. Can different *multi-prompt* techniques help the PLM better understand the DST task? These research methods are formally described in the later sections of this proposal.

2 Background & Related Work

2.1 Dialog State Tracking (DST)

Task-oriented dialog systems, both modular and end-to-end systems, solve a wide range of tasks (ticket booking, restaurant booking, etc.) across different domains. Since task-oriented dialog systems require strict response constraints as they aim to accurately handle the user messages, modular systems were proposed to generate responses in a controllable way. A typical modular-based system uses a modular pipeline, which has four modules that execute sequentially - Natural Language Understanding (NLU), Dialog State Tracking (DST), Policy Learning (POL), and Natural Language Generation (NLG). In this thesis, the focus is on the DST module of the modular-based dialog system. The Dialog state tracker infers the belief states (or user goals) from every turn of the dialog history and provides this information to the next module. For example, consider the user message - *"Plan a train trip to Berlin this Friday"* - the DST Module is supposed to extract belief states (*slot, value*) pairs as follows: {(*destination, Berlin*), (*day, this Friday*)}.

2.2 Pre-trained Language Models (PLMs)

Large pre-trained language models are trained on huge amounts of textual data and are used to solve a variety of NLP tasks. Pre-trained transformer-based language models such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018) have achieved state-of-the-art performance on many tasks. GPT-2 (Radford et al., 2019) is a state-of-the-art auto-regressive language model trained on large amounts of open web text data. GPT-2 is trained with a simple objective to predict the next word, given all previous words within some text. The training objective of the pre-trained LMs plays an important role in determining its applicability to particular prompting tasks (Liu et al., 2021). For example, left-to-right auto-regressive LMs may be particularly suitable for *prefix* prompts.

The baseline model of this thesis SOLOIST (Peng et al., 2021) uses a 12-layer GPT-2. SOLOIST uses the publicly available 117M-parameter GPT-2 as initialization for task-grounded pre-training. The prompt learning model of this thesis will use the SOLOIST to learn the DST task.

2.3 SOLOIST

SOLOIST (Peng et al., 2021) is a task-oriented dialog system that uses transfer learning and machine teaching to build task bots at scale. SOLOIST uses the *pre-train, fine-tune* paradigm for building end-to-end dialog systems using a transformer-based auto-regressive language model GPT-2 (Radford et al., 2019), which subsumes different dialog modules (i.e., NLU, DST, POL, NLG) into a single model. In a *pre-train, fine-tune* paradigm, a fixed *pre-trained* LM is adapted to different downstream tasks by introducing additional parameters and *fine-tuning* them using task-specific objective functions. In the pre-training stage, SOLOIST is initialized with the 12-layer GPT-2 (117M parameters) and further trained on large heterogeneous dialog corpora. The primary goal at this stage is to learn task completion skills such as DST and POL. Belief state prediction (DST) is one of the tasks in the task-grounded pre-training, which will be utilized in this thesis. In the fine-tuning stage, the pre-trained SOLOIST model can be used to solve new tasks by just using a handful of task-specific dialogs.

In this thesis, the pre-trained SOLOIST will be utilized in the baseline model. In the fine-tuning stage, a multi-domain task-oriented dialog dataset will be applied to solve the belief state prediction task. The predictions and results from this task can be used to compare with the prompt-based model for detailed analyses.

2.4 *Pre-train, Prompt, and Predict (PPP)* Paradigm

Prompt-based learning (also dubbed *"pre-train, prompt, and predict"*) is a new paradigm that aims to utilize PLMs more efficiently to solve downstream NLP tasks (Liu et al., 2021). In this paradigm, instead of adapting pre-trained LMs to downstream tasks via objective engineering, downstream tasks are reformulated to look more like those solved during the original LM training with the help of a textual *prompt*. To perform prediction tasks, the original input x is modified using a *template* into a textual *prompt* x' that has some unfilled slots, and then the PLM is used to probabilistically fill the unfilled information to obtain a final string $\hat{x}$, from which the final output y can be derived.

For example, to recognize the emotion in the text, where *input* $x =$"I missed the bus today.", the *template* may take the form such as "[X] I felt so [Z]". Then, *prompt* x' would become "I missed the bus today. I felt so [Z]" and ask the PLM to fill the slot [Z] with an emotion-bearing word. There are two main varieties of prompts: *cloze prompts*, where the slot [Z] is to be filled in the middle of the text, and *prefix prompts*, where the input text comes entirely before [Z]. In general, for tasks that are being solved using a standard auto-regressive LM, prefix prompts tend to be more helpful, as they mesh well with the left-to-right nature of the model.

In this way, by selecting the appropriate prompts, the pre-trained LM can be used to predict the desired output, sometimes even without any additional task-specific training. In this thesis, prompt-based methods will be utilized to train and help PLM understand the DST task.

2.5 Prompt learning for DST

Existing work by Lee et al. (2021) uses slots as prompts, along with the natural language descriptions of the schema for generating corresponding values. This slot-based prompt DST approach uses encoder-decoder LM with a bi-directional encoder. This method relies on the known ontology of the slots and requires a lot of training data for fine-tuning PLM. In real-world applications, defining all possible slots is difficult due to new domains and users' continuous needs. Yang et al. (2022) proposed a new prompt learning framework that uses values as prompts and doesn't rely on the ontology of the slots. This thesis will apply the value-based prompt approach for few-shot DST.

3 Methods

The main goal of thesis is to explore the prompt learning framework for few-shot DST designed by Yang et al. (2022) and propose some improvements. This thesis work can be subdivided into three tasks: (1) Apply prompt learning framework for few-shot DST, (2) Evaluation and analyses of belief-state predictions, (3) Multi-prompt learning methods.

3.1 Prompt learning framework for few-shot DST

This task aims to reproduce the results from Yang et al. (2022) and apply some minor improvements by utilizing multi-prompt methods. There's no publicly available implementation of this prompt learning framework. This task implements the prompt learning framework using Soloist baseline.

Dataset The baseline and prompt-based methods are evaluated on MultiWOZ 2.1 (Eric et al., 2019) dataset. MultiWOZ 2.0, originally released by Budzianowski et al. (2018), is a fully-labeled dataset with a collection of human-human written conversations spanning over multiple domains and topics. Eric et al. (2019) added some fixes and improvements to dialogue utterances and released MultiWOZ 2.1, which contains 8438/1000/1000 dialogues for training/validation/testing respectively. Yang et al. (2022) excluded two domains that only appear in the training set. Under few-shot settings, only a portion of the training data will be utilized to observe the performance in a low-resource scenario.

SOLOIST Baseline SOLOIST (Peng et al., 2021) is the baseline for the prompt-based approach. SOLOIST is initialized with the 12-layer GPT-2 (Radford et al., 2019) and further trained on two task-oriented dialog corpora (Schema and Taskmaster). The task-grounded pre-training helps SOLOIST to solve two dialog-related tasks: *belief state prediction* and *response generation*. For the baseline implementation, the pre-trained SOLOIST will be fine-tuned on MultiWOZ 2.1 dataset and perform the belief predictions task for DST. The main focus of this thesis is on prompt-based methods, however, the SOLOIST baseline implementation is required for comparing the belief state predictions and performance of prompt learning.

Value-based Prompt A general idea for generating (*slot, value*) pairs is to use slots in the prompts and generate corresponding values (Lee et al., 2021). For example, given the utterance - "*Plan a trip to Berlin*" and slot (*destination*), the prompt to the PLM could become "*Plan a trip to Berlin. destination = [z]*" and the PLM is expected to generate *[z]* as "*Berlin*". However, this approach relies on the ontology of the slots, and the fixed set of slots can change in real-world applications. Yang et al. (2022) proposed a *value-based prompt* that uses values in the prompt and generates corresponding slots. This method doesn't require any pre-defined set of slots and can also generate unseen slots. Consider this prompt template: "*belief states: value = [v], slot = [s]*", the prompt function f can be of form $f(v) =$ *[utterances] belief states: value = [v], slot = [s]*, given the value candidate $v =$ "*London*", the PLM should be able to generate *slot [s] = "destination"*. The overall training objective of value-based prompt generation is maximizing the log-likelihood of slots in the training dataset D:

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

where $P\left(s_t \mid c_t, f\left(v_t\right)\right)$ is the probability of slot s_t given dialog history c_t and prompt-function f filled with value v_t for each turn t. The loss $\mathcal{L}$ from this step will be combined with the loss from the next step in order to compute the final loss. While training, the values from the annotated training dataset are utilized to construct prompts.

Inverse Prompt The *inverse prompt* mechanism is used to generate values by prompting slots. After generating slot s using value-based prompts (previous step), the generated slot s is presented to the inverse prompt function I. The inverse prompt aims to generate the value v' which is supposed to be close to original value v. The template for inverse prompt function is I = "*belief states: slot = [s], value = [v]*". This inverse prompt can be considered as an auxiliary task for this prompt-based approach, which can improve the performance by helping the PLM understand the task and tune the slot generation process. The loss function $\tilde{\mathcal{L}}$ for the inverse prompt mechanism:

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

The final loss $\mathcal{L}^*$ can be computed by combining loss from value-based prompt $\mathcal{L}$ and inverse prompt loss $\tilde{\mathcal{L}}$:

$$\mathcal{L}^* = \mathcal{L} + w * \tilde{\mathcal{L}}$$

where w is a decimal value (0,1) and can be used to adjust the influence of inverse prompt.

Training For training the above prompt-based approach, SOLOIST (117M params) pre-trained model will be utilized and fine-tuned on the prompt-based slot generation process. To fine-tune SOLOIST, MultiWOZ 2.1 dataset is used, the dialog history and values are directly given to the prompts. Inverse prompt is only used during the training phase. For evaluating the prompt-based model ability to generate slots under low-resource data settings, few-shot experiments will be performed while training. Experiments will be conducted by choosing random samples of the training data (1%, 5%, 10%, and 25%) for each domain. Few-shot experiments will be performed on both SOLOIST baseline and the prompt-based model.

Testing In the testing phase, only value-based prompts are utilized for slot generation. While testing, candidate values are not known. Following the existing work (Min et al., 2020), values can be extracted directly from the utterance. First POS tags, named entities, and co-references are extracted. A set of rules can be used to extract candidate values given the POS and entity patterns like considering adjectives and adverbs, filtering stop words and repeated candidates.

3.2 Evaluation & Analyses

Evaluation Metrics The standard metric joint goal accuracy (JGA) will be adopted to evaluate the belief state predictions. This metric compares all the predicted belief states to the ground-truth states for each turn. The prediction is correct only if all the predicted states match the ground-truth states. Both slots and values must match for the prediction to be correct. To omit the influence of value extraction, Yang et al. (2022) proposed JGA*, the accuracy is computed only for the belief states where the values are correctly identified. These evaluation metrics can answer the following questions: **Q1**: How do the prompt-based methods perform overall compared to SoTA SOLOIST? **Q2**: Can the prompt-based model perform better under the few-shot settings? **Q3**: Does JGA* has a better score than JGA?

Analyses of belief state predictions The main goal of this task is to analyze belief state predictions. The predictions from SOLOIST baseline and prompt-based methods will be compared and analyzed to identify the improvements and drawbacks. A detailed error analyses will be performed on the wrong belief state predictions.

3.3 Multi-prompt learning methods

The *value-based* prompt described in the previous sections utilize a *single* prompt for making predictions. However, a significant body of research has demonstrated that the use of multiple prompts can further improve the efficacy of prompting methods (Liu et al., 2021). There are different ways to extend the single prompt learning to use multiple prompts. This task will explore three multi-prompt learning methods: *Prompt ensembling*, *Prompt augmentation*, and *Prompt decomposition*. This task aims to answer the following questions - **Q1**: Can combining different *multi-prompt* techniques help the PLM better understand the DST task? **Q2**: How do various hand-crafted prompt functions influence the prompt-based model?

Prompt Ensembling This method uses multiple *unanswered* prompts during the inference time to make predictions (Liu et al., 2021). This idea can leverage the complementary advantages of different prompts and stabilize the performance on downstream tasks. Yang et al. (2022) applied prompt ensembling for the value-based prompt to effectively utilize four different prompts. A simple way for ensembling is to train a separate model for each prompt and generate the output by applying the weighted averaging on slot generation probability. The probability of slot s_t can be calculated via:

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

where $|K|$ represents the number of prompt functions, f_k is the k-th prompt function, α_k is the weight of prompt k. This task will utilize prompt ensembling differently from Yang et al. (2022), by combining other multi-prompt methods. In this task, experiments will be performed on various prompt templates to find the most effective and suitable prompts in combination with other multi-prompt methods.

f_1	belief states: value = [v], slot = [s]
f_2	belief states: [v] = [s]
f_2	[v] is of slot type [s]
f_4	[v] is the value of [s]
$\vdots$	

Table 1: Examples of different prompt functions for ensembling

Prompt Augmentation *Prompt Augmentation*, sometimes called *demonstration learning* (Gao et al., 2021), provides a few additional *answered prompts* that can demonstrate to the PLM, how the actual prompt slot can be answered. Sample selection of answered prompts will be manually hand-picked from the training data. Experiments will be conducted on different sets of samples. Table 2 provides an example for prompt augmentation.

I want to book a cheap hotel.	*cheap* is of slot *price range*
Plan a train trip to Berlin.	*Berlin* is of slot *destination*
Find me an Italian Restaurant.	*Italin* is of slot *food*
Recommend a movie at Cinemaxx.	*Cinemaxx* is of slot [s]

Table 2: Examples of prompt augmentation with answered prompts

Prompt Decomposition For utterances where multiple slot values should be predicted, directly using a single prompt for generating multiple slots is challenging. One intuitive method is to breakdown the prompt into sub-prompts, and generate the slots for each sub-prompt separately. For each candidate value in the utterance, construct a *value-based* prompt and generate the slot. This approach will be utilized in both training and testing phases. This sort of *prompt decomposition* has been explored by Cui et al. (2021) for named entity recognition(NER) task.

Utterance:	Book a flight to Stuttgart tomorrow evening.
Prompt 1:	belief states: *Stuttgart* = [s]
Prompt 2:	belief states: *tomorrow* = [s]
Prompt 3:	belief states: *evening* = [s]

Table 3: Prompt decomposition example

4 Work Plan

This section outlines the work plan for the thesis. Main tasks, sub-tasks and deadlines are provided in the below table 4.

Main Tasks	**Sub-tasks**	**Deadlines**
Baseline Implementation (Section 3.1)	Environment Setup SOLOIST Baseline	22.08.2022 (3 weeks)
Prompt Learning (Section 3.1)	Value-based Prompt Inverse Prompt Mechanism Training & Testing	19.09.2022 (4 weeks)
Evaluation & Analyses (Section 3.2)	Evaluation Metrics Graphs & Visualizations Prediction Analysis Error Analysis	10.10.2022 (3 weeks)
Multi-prompt Learning (Section 3.3)	Prompt Ensembling Prompt Augmentation Prompt Decomposition	07.11.2022 (4 weeks)
Thesis Writing (First draft)	LaTeX Typesetting	19.12.2022 (6 weeks)
Thesis Writing (Final submission)	LaTeX Typesetting Corrections & Fixes	16.01.2023 (4 weeks)

Table 4: Thesis work plan with deadlines

Other important dates:

- Thesis Start Date: **01.08.2022**
- Proposal Submission: **27.07.2022**
- Thesis Deadline: **01.02.2023**

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 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*, pages 5016–5026, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1547. URL https://aclanthology.org/D18-1547.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.161. URL https://aclanthology.org/2021.findings-acl.161.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tür. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *CoRR*, abs/1907.01669, 2019. URL http://arxiv.org/abs/1907.01669.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference* *on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.295. URL https://aclanthology.org/2021.acl-long.295.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. Dialogue state tracking with a language model using schema-driven prompting. *CoRR*, abs/2109.07506, 2021. URL https://arxiv.org/abs/2109.07506.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586, 2021. URL https://arxiv.org/abs/2107.13586.

Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. Few-shot bot: Prompt-based learning for dialogue systems. *CoRR*, abs/2110.08118, 2021. URL https://arxiv.org/abs/2110.08118.

Qingkai Min, Libo Qin, Zhiyang Teng, Xiao Liu, and Yue Zhang. Dialogue state induction using neural latent variable models. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3845–3852. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/532. URL https://doi.org/10.24963/ijcai.2020/532. Main track.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. SOLOIST: building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824, 2021. doi: 10.1162/tacl_a_00399. URL https://aclanthology.org/2021.tacl-1.49.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Yuting Yang, Wenqiang Lei, Juan Cao, Jintao Li, and Tat-Seng Chua. Prompt learning for few-shot dialogue state tracking. *CoRR*, abs/2201.05780, 2022. URL https://arxiv.org/abs/2201.05780.