Dialogue Management in Perspective: An Overview of Evolving Approaches Going into the Era of Large Language Models

Onur Deniz Güler
Technical University of Munich
deniz.gueler@tum.de

1 Introduction

Conversational agents have been in use for more than half a century. In addition to responding to humans via natural language, managing the dialogue and accessing knowledge bases is an important task of conversational agents. The process known as Dialogue Management (DM) encompasses the comprehensive understanding of the inherent logical progression observed in human-to-human dialogues. It involves considering the accumulated dialogue history, referenced entities, previously agreed-upon statements, and the user utterance at each turn to steer the ongoing dialogue.

DM approaches have historically focused on managing Task-Oriented Dialogue (TOD) (Jacqmin et al., 2022), which aims to automate the execution of domain-specific tasks such as reserving a table at a restaurant or booking a flight. Despite the prevalence of contemporary open-ended dialogue systems that rely on large language models (LLMs) at their core, industrial conversational agents are primarily designed to act as TOD agents, operating on DM modules founded upon relatively antiquated yet firmly established methodologies (Quan et al., 2021). In domain-specific TOD applications, DM has traditionally had two separate consecutive modules named Dialogue State Tracking (DST) and Dialogue Policy (DP). DST was conceived as the module responsible for keeping track of the flow of information through turns of dialogue and DP as the module deciding on the next actions to take in response to the human agent.

The primary contribution of this paper is the synthesis of a comprehensive overview of prominent DST and DP approaches found in the literature, accompanied by selected illustrative methods for some of these approaches. This endeavor aims to present a holistic perspective of DM in the era of rapidly advancing research on end-to-end DM in both TOD systems and open-ended dialogue systems with LLMs. The pursuit of end-to-end methods stimulates the need to examine DST and DP from an integrated and all-encompassing standpoint. To achieve this perspective, we separately examine the evolution of DST and DP within the TOD paradigm which has a rich DM background in the literature. We then examine a very recent end-to-end DM approach in detail to understand where the research stands today.

2 Dialogue Management Methods

DM methods can broadly be separated into two distinct categories (Kwan et al., 2023). The first category encompasses DM methods which are an indiscernible part of an end-to-end natural language processing (NLP) model, exemplified by ChatGPT, where the DM process seamlessly blends into the autoregressive LLM (Ouyang et al., 2022). These methods are particularly useful for open-ended dialogues and are being extensively investigated for both chatbot development (Huang et al., 2020) and TOD systems (Bordes et al., 2017a), as surveyed by (Jacqmin et al., 2022). The second category comprises modular DM methods with modules developed independently to be integrated into a pipeline.

The modular approach traditionally has four modules called natural language understanding (NLU), dialogue state tracking (DST), dialogue policy (DP) and natural language generation (NLG). The DST and DP modules form the DM module. NLU's task is to extract structured information from user utterances, and NLG's task is to convert system actions into natural language responses. We thoroughly define DST and DP in their respective sections. Several researchers, including Lee et al. (2021), Bordes

et al. (2017b), and Quan et al. (2021), have investigated diverse combinations of these modules, forming larger singular entities within the dialogue system's pipeline. Figure 1 presents an overview of such possible arrangements observed in modular DM systems.

In the context of the DST module, there has been a notable transition from rule-based methodologies towards machine learning-based methods. Within the machine learning paradigm, there is a growing inclination to revert back to generative models. This shift aims to address the domain limitations inherent in discriminative approaches, which were initially proposed to replace earlier generative models in DST. For DP, methods evolve from rule-based methods to supervised learning methods, to later on be dominated by reinforcement learning (RL) approaches. We include a short review of of rule-based methodologies in our overview, as these approaches continue to be favored in commercial applications owing to their inherent explainability (Quan et al., 2021). Furthermore, lack of data in some domains renders the rule-based methods the only option.

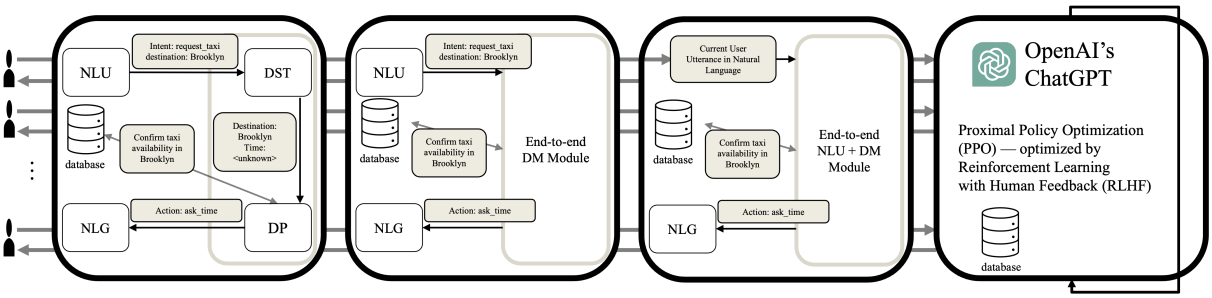

Figure 1: Overview of various granularities of the modules of a TOD system. From left to right: (1) A fully modularized TOD system. (2) A TOD system with an end-to-end DM. (3) A TOD system with an end-to-end NLU and DM. (4) ChatGPT's *autoregressive LLM* approach with all components merged into an end-to-end workflow.

3 Dialogue State Tracking

DST methods in TOD systems have attempted to tackle varieties of the same problem at different scales. They all have a ubiquitous view of the core problem at hand. At each dialogue turn of the dialogue system; i.e, when the system must produce a response to what the user has just uttered, the dialogue state which incorporates the *belief state* and the *dialogue action* must be updated according to the user intent. User intent represents what the user wants to communicate to and receive from the system (e.g., sharing their cuisine preferences with the system to get restaurant suggestions).

The *belief state* of the system represents the knowledge that is accumulated by the DST during the conversation. The constituents of the belief state are *(domain, slot, value)* tuples, where the *(domain, slot)* pairs are part of an ontology predefined by system designers to represent the necessary slots of information required to complete a task catering a specific domain. For example, an ontology representing the task of restaurant bookings would have *(domain, slot)* pairs such as *(restaurant, reservation-day)*, *(restaurant, number-of-people)*. A slot might have predefined values to select from, or just have a type (e.g., *integer* for a slot that admits user's age) but no restrictions on the value.

The task of the simplest conceivable DST would be to receive *(domain, slot, value)* extractions from the NLU accompanied by the user intent at each turn, and complement each of the possible *(domain, slot)* pairs from the ontology with a (potentially *null* or *dont-care*) value as the dialogue progresses. This task is called *slot filling*, and the main challenge of the task is to interpret the output of the NLU to determine whether a useful value for a given slot was uttered by the user. If an uttered value requires updating a previously filled slot, how exactly to update this slot is to be determined by the DST. At each dialogue turn, the updated belief state guides the DP module to decide on the next actions to take.

Even though the core DST task of slot filling is defined similarly by most methods, the abstraction level of the dialogue history, belief state history, and most recent user utterance received at the DST input varies (Balaraman et al., 2021). Some DST modules have access to the entire dialogue and dialogue state

history at all times, but some only has access to the last state and the most recent user utterance. Using complex inputs provides robustness to the module, while making the task much more complex.

Performance in DST is commonly measured by Joint Goal Accuracy (JGA) as defined by Williams et al. (2013). JGA measures how well the model predicts the belief state at each dialogue turn, and is equal to the proportion of dialogue turns which have exactly the same belief state as the groundtruth at that turn. For training and evaluations of DST, variations of the MultiWOZ dataset (Budzianowski et al., 2018) are used. These datasets are the main benchmark for DST, and they comprise belief states at all dialogue turns across dialogues collected from conversations between two humans in various domains.

3.1 Rule-based Dialogue State Tracking

In earlier DST approaches, most models used hand-crafted rules to determine how to update the belief state. These rules might involve a straightforward process of utilizing the *(domain, slot, value)* tuple obtained from the NLU to ascertain the value of the corresponding *(domain, slot)* in the belief state (Larsson and Traum, 2000). Additionally, if any slots previously filled are mentioned in the new utterance, the belief state s_{t-1} from the previous time step $t-1$ would be updated accordingly. Alternatively, the rules might encompass a more intricate approach, incorporating techniques such as regular expressions (Bobrow et al., 1977), parsers, and context-free grammars; with some methods occasionally incorporating probabilistic modelling into the rules.

Rule-based approaches continue to be utilized in scenarios where there is insufficient data available to train a machine learning model. Moreover, these approaches offer the advantage of being more interpretable than black-box models, as the underlying logic is designed and implemented by domain experts, ensuring explicit incorporation of domain understanding. For these reasons, rule-based DM is still a common practice in commercial conversational agents. However, they cannot address the innate uncertainties in natural language, and it is much harder to introduce larger dialogue history windows to these models, as the complexity of the hand-crafted rules increase with increasing input complexity. Moving towards probabilistic models were suggested as early as 1997 by Pulman (1997), and recently developed DST methods ubiquitously favor machine learning based approaches.

3.2 Discriminative ML Based Dialogue State Tracking

Discriminative machine learning methods for DST aim to model a distribution over the belief state to determine the *(domain, slot, value)* tuples for each such tuple in the ontology. These models can broadly be summarised to represent the following distribution:

$$P(S_t, S_{t-1}, ...S_{t-K}|F_t, F_{t-1}, ..., F_{t-T}) \tag{1}$$

where S_t represents the belief state at time t, and F_t represents the features extracted at time t. T can be equal to t or $zero$ in edge cases, meaning that the model accesses the entire feature history or only the last utterance, respectively. K can be equal to t or $zero$ in edge cases, meaning that the state predictions are correlated (as in a sequential model like a recurrent neural network) or they are independent, respectively. The *(domain, slot, value)* tuples can be assumed to be independent from each other, or interdependencies (e.g. hotel stars and price range in a hotel booking TOD) can also be modelled into the distribution such that the model captures the correlation between the slots, e.g. by using self attention (Ye et al., 2021).

Features can be engineered using certain values from the dialogue history and the belief state history. Non-engineered features like the entire dialogue in natural language up until the current time step can also directly serve as input. The *model* can be a neural network, a ranking model, maximum entropy linear classifier, a multinomial logistic regression model. In the case of neural networks, using shared weights across models while admitting slot-specific features; or using multiple weights with output dimensions matching the number of possible values for a given slot are two of the explored options.

Metallinou et al. (2013) devise one of the renown discriminative methods using engineered features to summarise dialogue history. The method's hand-crafted features are extracted from the output of the Automated Speech Recognition (ASR) and Spoken Language Understanding (SLU) modules' hypotheses. For any given slot in the domain, the DST model admits three categories of features from the historical

ASR and SLU hypotheses: (i) *base features* based on the current turn (e.g. the probability of the slot being equal to one of its possible values), (ii) *history features* based on past turns (e.g., number of times of observances of the slot being equal to one of its possible values), (iii) *confusion features* which heuristically estimate ASR errors. On top of these three categories using the hypotheses, meta information such as the number of unique SLU suggestions from the entire dialogue history are incorporated into the features as well.

Given the engineered features, Metallinou et al. (2013) use a maximum entropy model (Berger et al., 1996) where the probability distribution of a multiclass label is modelled through feature functions describing the relationship between the features and possible labels. Maximum entropy models are multiclass extensions of binary log-linear models which in turn are maximum likelihood estimators in essence, and the distribution governed by these models are expressed as:

$$P(y|\mathbf{x}, \mathbf{w}) = \frac{\exp\left(\sum_{j \epsilon J} w_j f_j(\mathbf{x}, y)\right)}{\sum_{y \epsilon Y} \exp\left(\sum_{j \epsilon J} w_j f_j(\mathbf{x}, y)\right)} \tag{2}$$

where w_j represent the model parameters and f_j represent the feature functions. This approach, which is representative of methods using discriminative models admitting fixed-sized hand-crafted features precedes discriminative DST approaches using transformer based language models. Recurrent neural networks are employed to model the belief state distribution using sequences as well (Henderson et al., 2014), (Lee, 2013). For brevity, we omit these methods in this paper.

A more modern, transformer architecture based approach is explored by TOD-BERT (Wu et al., 2020). This approach looks into extending the pre-training of BERT (Devlin et al., 2019) using TOD corpora to improve downstream linear probing performance of BERT on downstream TOD tasks such as NLU, DST, DP, and NLG. In the context of DST, they insert a slot projection layer G_j per *(domain, slot)* pair j on top of pre-trained TOD-BERT embeddings, and extract cosine similarity between the mean embedding of the tokens forming the entire dialogue history up until current time step t, and all values v_i^j that the *(domain, slot)* pair j can span. For a given slot j, and its possible value i, the similarity measure is defined as:

$$S_i^j = Sim(W_j(TODBERT(X)), TODBERT(v_i^j)) \in \mathbb{R}^1, \tag{3}$$

where X represents the entire dialogue utterances. The value with the highest cosine similarity is returned as the prediction. The projection layer essentially behaves as a per-slot shallow fully connected neural network with GELU (Hendrycks and Gimpel, 2016) non-linearities. The method forms a baseline for pre-trained language model (PLM) based discriminative transfer learning by achieving mediocre evaluation results (Jacqmin et al., 2022). Note that this does not hinder the method's success, since the main goal of the method is to show the efficacy of pre-training with TOD corpora, in contrast to BERT which was trained on general corpora. Figure A.1 in the appendices gives a side by side comparison of the two examined discriminative methods.

3.3 Generative ML Based Dialogue State Tracking

Generative models were initially proposed to address inherent uncertainties in natural language and speech. The integration of probabilistic methods these models introduced gained preference over rule-based systems, primarily due to their capacity to alleviate rigidity and tedious engineering processes associated with the latter. As examined by Young et al. (2013), several generative techniques relied on the utilization of Partially Observable Markov Decision Process (POMDP) to model dialogues as dynamic Bayesian networks. These methods incorporated multiple independence assumptions to facilitate parameter simplification. These independence assumptions, which overlooked the interdependencies between states, contributed to the abandonment of these models in favor of discriminative methods that effectively model the distribution of belief states, incorporating correlations. It is worth noting here that DP is also embedded in the Markov process, meaning that these earlier generative methods come closer to end-to-end methods than their discriminative contemporaries.

There has recently been a resurgence of generative approaches, with a completely different approach to generative modeling. Generative LLMs are employed to receive dialogue history in natural language, and generate predictions for the *(domain, slot)* pairs in the ontology. Lee et al. (2021) use T5 (Roberts et al., 2019), a sequence-to-sequence LLM, with prompt engineering. The method achieves the state-of-the-art (SOTA) result of its time, $JGA = 57.6\%$ on one of the literature standard datasets, MultiWOZ 2.2 (Budzianowski et al., 2018). In this method, the ontology is converted to per slot natural language descriptions to fit the prompt to be used with T5. For example, a *(domain, slot)* pair *(train, destination)* from the ontology with possible values *(London, Berlin, Paris)* gets a natural language description similar to "destination location of the train to be booked, with possible values London, Berlin, Paris". At conversation time, the value of each slot is predicted by prompting T5 with a concatenation of the entire dialogue history until the current time step, with the natural language description of the slot to be predicted. Figure A.2 in the appendices displays this process.

These prompt-based methods, and their analogues, exhibit the significant advantages inherent in generative models. Due to the autoregressive nature of language models like T5 conducting next token prediction at the decoder stage, there exists the potential to extract previously unobserved values for a particular slot in the ontology from an unfamiliar user utterance. This enables an easier implementation of dynamically changing values for the slots, even in domains which have not been encountered before. In contrast to discriminative models characterized by predefined discrete value sets, generative models have the capability to predict continuous values. Lee et al. (2021) demonstrate that NLU and DST modules can come together to create a more extensive module that can effectively process natural language with less restrictive context window constraints. This advancement brings the module closer to task-awareness, making way to its possible extension to open-ended DM.

4 Dialogue Policy

DP modules have the distinct task of receiving the current dialogue state and determining the next action to take. These actions comprise employing the NLG module to generate a response (e.g., asking the user for confirmation), performing database queries and API calls to access knowledge bases, starting or ending a dialogue. Dialogue state is formed by the belief state tracked by the DST, and the past dialogue actions taken by DP.

In addition to rule-based DP methods, two main paradigms are observable in the DP literature: supervised learning, and reinforcement learning. In the learning based methods, the DP could be defined as a multi-class classifier predicting a single action from all possible system actions, or a multi-label classifier predicting one or more actions simultaneously. In recent years, there has been a notable preference in the literature for reinforcement learning methods.

4.1 Rule-based Dialogue Policy

A trivial rule-based DP module would prompt the user with questions until a belief state that makes it possible to complete the task is achieved. In earlier TOD systems, this approach was indeed employed. More sophisticated rule-based approaches have also been explored. In those, system developers hand-craft the possible scenarios. It is a tedious task to account for all possible scenarios that could occur in a real dialogue between a human and a TOD agent, hence it is costly to implement hand-crafted rules. However, as in the case of DST, in situations where there is no training data, or when the system explainability is task-critical, rule-based DP modules are preferred over machine learning based ones.

4.2 Supervised Learning-based Dialogue Policy

Supervised learning approaches used in DP modules train on datasets like MultiWOZ (Budzianowski et al., 2018) which provide the dialogue states and the entire dialogue utterance history at each time step. Since the task is of a sequential nature, using models that can process the relations between time steps is preferred. In an example where a Long Short-Term Memory (LSTM) network is used (Gritta et al., 2021), the DP module models the policy as follows:

$$\pi_\theta(a_t|S_t, S_{t-1}, ...S_{t-T}) \tag{4}$$

where a_t represents the next action(s) to take, and S_t through S_{t-T} capture the past dialogue states up-to history length T. The action with the highest probability given the modelled distribution is predicted. The dialogue states must be featurized through some encoding, and it could be as simple as one-hot encoding of whether a *(domain, slot)* pair in the ontology is filled or not. The model learns all possible actions in its training data, and its output size is the number of distinct actions observed in the dataset. In the case of systems which can take more than one action within the same time step, structuring the output layer as a multi-label classification layer reduces complexity from an exponential boundary to a linear one (Gritta et al., 2021). The performance is measured through accuracy of taken actions.

Quan et al. (2021) attempt to upgrade a well established rule-based DP module of Carina Dialogue System (CDS) with a supervised learning approach using TOD-BERT (Wu et al., 2020). The original CDS has an end-to-end rule-based approach where NLU, DST and DP are incorporated into one single DM module. The new approach replaces the NLU and DM modules of the original system with an end-to-end TOD-BERT based DM. TOD-BERT embeds the entire dialogue history until the current time step into *context features*. These features are fused with the dialogue state and passed through a discriminative classifier, which predicts a single action from all valid actions.

Each predicted action is a *mini* dialogue turn, where the system can do database queries, state updates (e.g., inform a *(domain, slot)* pair with a value). Action predictions are continued with the DM module self-triggering itself with mini dialogue turns, until it decides to end its turn and pass it over to the user. The developers of this approach create an updated version of the CamRest676 dataset (Wen et al., 2017), (Wen et al., 2016), and achieve high accuracy (95.73%) on mini-turn action predictions, with a small training dataset. This approach exemplifies the strengths of incorporating a pre-trained task-oriented LLM into the pipeline, and approximates further towards a fully end-to-end TOD system.

4.3 Reinforcement Learning-based Dialogue Policy

RL was realized to be a possible DP optimization method as early as 1998, when Levin et al. (1998) structured the DP module as a Markov Decision Process (MDP). RL is a common way of optimizing MDP based systems (Graßl, 2019), and MDP based DP modules were explored before supervised learning methods dominated the machine learning based approaches. However, with the advent of deep learning based RL methods, the RL paradigm started to be reconsidered in DP, and recently has become favored over supervised learning approaches (Kwan et al., 2023). RL approaches mitigate the nearsightedness of supervised learning methods (Henderson et al., 2008), where the agent action predictions are made based only on dialogue history, without any explicit optimizations for dialogue future.

Kwan et al. (2023) concisely present RL in the context of DP. When the underlying model of the DP module is an MDP, the system is completely defined by the (S, A, P, R, γ) tuple, its accompanying *value function*, and the learnt *policy*. S, A, P, R are vectors of dimensions t, such that each dimension represents a time step. S represents dialogue states, tracked by DST. A represents dialogue actions. These actions could be restrained to comprise only agent actions, or could also comprise user actions in multi-agent RL scenarios. γ is the discount factor to regulate the reward.

P represents the transition model of the environment which is directly the user in the case of DP. The user awards or penalizes the RL-based DP module through their utterances. In some approaches, the user itself is modelled as a conversational agent with its own MDP system; the user and the agent concurrently learn a policy (Georgila et al., 2014). The concurrent learning approach removes the need for simulated user responses generated by sequence-to-sequence models. These simulations are created in an effort to provide data where otherwise actual human speeches are scarce for deep RL training. However, they fail to fully capture real-world behaviour of humans. Thus, concurrent RL has recently been favored in DP.

R represents the reward, and the reward function varies between approaches. Manually designing a reward function suited for DP was found to be promising. The optimal policy π is learnt through reinforcement learning to maximize the value function V, which is defined as the expected reward from the environment over all time steps of the dialogue:

$$V^{\pi}(s) : \mathbb{E}\left[\sum_{t=0}^{T} \gamma^{t} r_t | s_0 = s\right] \tag{5}$$

As surveyed by Kwan et al. (2023), we observe many RL approaches in the literature, all of which come up with a method customizing one or more of the parameters S, A, P, R, and π. Most of these approaches treat the DP problem from the traditional individual-module-in-pipeline perspective, and optimize it individually. There has recently been a completely end-to-end approach using RL on custom-defined reward and policy functions. We take a detailed look at this approach in Section 5.

5 An End-to-End TOD System: Task-optimized Adapters

As we have examined in previous sections, pre-trained language models are gaining prevalence in TOD methods, with renown models like TOD-BERT (Wu et al., 2020) trained specifically on dialogue corpora, used as a backbone for transfer learning, with learning performed separately for each downstream task of a TOD system. These models abide by the modular approach where each module of the pipeline is optimized separately, and then concatenated to collaborate as a TOD system. However, the independent optimizations of these modules make it hard to align for the global target of a successful dialogue. Besides, it is harder to propagate user feedback through the pipeline, and at inference time, errors propagate from one module to the other, decreasing overall performance. End-to-end systems has thus been desired in TOD systems; conditioned, until recently, on the given that there is large amounts of data to achieve this ambitious goal of optimizing all modules at once.

Bang et al. (2023) make a very recent contribution to the pursuit of end-to-end TOD systems. They devise a method to efficiently train an end-to-end model with NLU, DM, and NLG modules optimized with the same training goal using a pre-trained language model as the backbone. DST and DM modules are indiscernibly merged into a DM module, which they shortly refer to as DST. The DST output steers both the NLG module, and generates database queries to retrieve information. Their approach introduces a new way of incorporating PLMs and reinforcement learning into end-to-end TOD systems, and their results are promising for stimulating a research trend towards parameter efficient training methods using PLMs trained on general corpora.

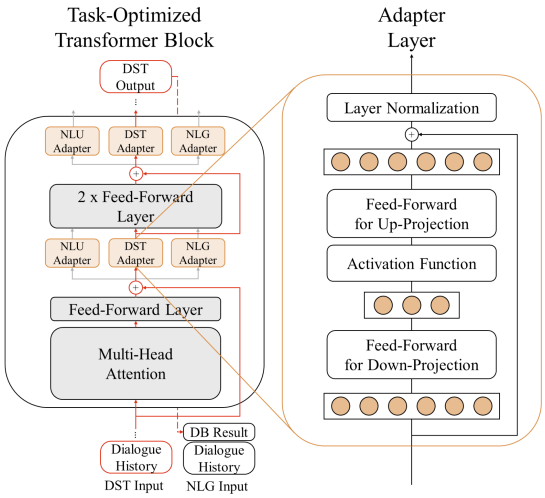

Figure 2: Transformer block architecture of the Task-Optimized Adapters (Bang et al., 2023).

The main problems that Bang et al. (2023) address in the PLM-based end-to-end approaches can be summarized in 4 points. (1) Pre-training of LLMs with task-oriented dialogue corpora require large amount of data and compute time, hence it is not possible to pre-train for each TOD module separately in a parameter efficient way. (2) Debugging each TOD module separately is harder. (3) Task-specific

variables of TOD systems are completely disregarded in pre-training. (4) The shared model parameters of a PLM introduces interference among each module task.

The two main solutions they come up with are to introduce adapters right after the attention layers in the transformer blocks of the sequence-to-sequence LLM T5 (Roberts et al., 2019), and perform RL with a novel reward based on the literature-standard DST metric Joint-Goal Accuracy (JGA). Bang et al. (2023) achieve SOTA performance ($JGA = 63.79\%$) in the MultiWOZ 2.2 dataset (Budzianowski et al., 2018). This is a slight improvement over the currently available models. However, task-oriented adapters are lightweight, fast, and trained in a parameter efficient manner. Thus, the achieved improvement in performance holds significance, irrespective of its relatively modest magnitude.

Training only the adapters, instead of a complete fine-tuning of the LLM, addresses problem (1). Adapters have considerably fewer parameters than the frozen LLM. This helps reduce the need for large data, and adds the benefit of parameter efficient fine-tuning. Since the adapters are separately accessible, debugging their performance issues are easier, solving problem (2). The JGA reward in RL induces task-optimized learning, which helps separate the the parameters to mitigate problems (3) and (4). This reward model also reduces the dependency of the optimization target on next-token-prediction based errors which autoregressive models suffer from while attempting to promote coherence and fluency.

The novel adapter that the method introduces comprises two fully-connected layers with a non-linearity in between. For each task i in NLU, DST, NLG they insert an adapter after the multi-head attention layer in each transformer block j (Figure 2). This enables sharing the PLM parameters, while learning task specific parameters efficiently. The fully-connected layer receives attention scores and residual input token embeddings H_j, and scales the features down and then up again to feed the rest of the transformer block with A_{ij}:

$$A_{ij} = (W_{up} * RELU(W_{down} * H_j) + H_j) \tag{6}$$

The scaled-down dimension h was used as a hyperparameter in training. The loss function J they use in reinforcement learning, the policy function J_{policy} , and the rewards for DST and NLG are given as:

$$J(\theta) = \alpha * J_{policy}(\theta) + (1 - \alpha) * CrossEntropyLoss(y, \hat{y}) \tag{7}$$

$$J_{policy} = -logP(\hat{y}) * Reward(y, \hat{y}) \tag{8}$$

$$Reward_{DST} = JGA(y, \hat{y}) + 1 \tag{9}$$

$$Reward_{NLG} = (1 - \beta) - E[BLEU(y_u, \hat{y_u}) + \beta * Success(y, \hat{y}) + 1 \tag{10}$$

We observe that the method uses two different policy losses conditioned on JGA achieved by the model, and the BLEU (Papineni et al., 2002) score of the NLG outputs. This balances out the categorical cross-entropy loss used in autoregressive prediction, while emphasizing the behavioural differences of the DST and NLG modules. The model focuses not only on generating correct natural language *(domain, slot, value)* tuples but also attaining a reward to track the belief states correctly. In training, the non-differentiable nature of the reward function is mitigated using the REINFORCE (Sutton et al., 1999) method. In their experiments, they observe that simple supervised learning is not enough on its own to bring the performance of task-optimized adapters to task-agnostic dialogue-corpora-based PLMs. However, with the introduction of the JGA reward model in training the method achieves SOTA results in DST.

In this approach, the NLU, DM, NLG modules are brought together within one LLM, with knowledge-base access incorporated into the end-to-end system through queries requested by the DST output as shown in Figure 2. The method employs the inherent NLU capabilities of an LLM, and steers its state tracking behaviour and natural language responses through a RL training regime with little data. The

issue of weight sharing for the different tasks are resolved, while a proof of concept for efficient end-to-end TOD systems are established. At its current state, the method fails to address the need for domain-agnostic generalizability and the necessity of domain expertise to determine *(domain, slot, value)* tuples. However, the introduction of JGA as a reward metric hints at possibilities of incorporating the rich findings of the DM literature to advance the pursuit of end-to-end TOD systems.

6 Conclusion

Dialogue Management is a crucial part of both task-oriented and open-ended dialogue systems. We have examined the evolution of the literature starting from rule-based approaches to very recent LLM-based approaches. There is a circular pattern of favoring methods over one another, as the NLP domain proceeds. We clearly see the steps towards end-to-end optimized TOD systems at various stages of DM development, and in today's early era of LLMs, we still see research trends grounding their methods on the traditional task-oriented DM-paradigm, metrics, and datasets due to the knowledge-grounded, manageable framework these assets provide. With end-to-end methods starting to achieve SOTA results in this framework, it is possible to see current open-ended LLM chatbots adopting some of these findings into their autoregressive dialogue management approaches to increase explainability and facilitate knowledge-grounded dialogue.

References

Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 239–251, Singapore and Online. Association for Computational Linguistics.

Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. Task-optimized adapters for an end-to-end task-oriented dialogue system. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7355–7369, Toronto, Canada. Association for Computational Linguistics.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay, Donald A. Norman, Henry Thompson, and Terry Winograd. 1977. Gus, a frame-driven dialog system. *Artificial Intelligence*, 8(2):155–173.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017a. Learning end-to-end goal-oriented dialog. In *International Conference on Learning Representations*.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017b. Learning end-to-end goal-oriented dialog. In *ICLR*. OpenReview.net.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 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*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kallirroi Georgila, Claire Nelson, and David Traum. 2014. Single-agent vs. multi-agent techniques for concurrent reinforcement learning of negotiation dialogue policies. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 500–510, Baltimore, Maryland. Association for Computational Linguistics.

Isabella Graßl. 2019. A survey on reinforcement learning for dialogue systems. *viXra*.

Milan Gritta, Gerasimos Lampouras, and Ignacio Iacobacci. 2021. Conversation graph: Data augmentation, training, and evaluation for non-deterministic dialogue management. *Transactions of the Association for Computational Linguistics*, 9:36–52.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, 34(4):487–511.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. pages 360–365.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Trans. Inf. Syst.*, 38(3).

Léo Jacqmin, Lina M. Rojas Barahona, and Benoit Favre. 2022. "do you follow me?": A survey of recent approaches in dialogue state tracking. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 336–350, Edinburgh, UK. Association for Computational Linguistics.

Wai-Chung Kwan, Hong-Ru Wang, Hui-Min Wang, and Kam-Fai Wong. 2023. A survey on recent advances and challenges in reinforcement learning methods for task-oriented dialogue policy learning. *Machine Intelligence Research*, 20(3):318–334.

Staffan Larsson and David R. Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Nat. Lang. Eng.*, 6(3–4):323–340.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sungjin Lee. 2013. Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451, Metz, France. Association for Computational Linguistics.

Esther Levin, R. Pieraccini, and Wieland Eckert. 1998. Using markov decision process for learning dialogue strategies. pages 201 – 204 vol.1.

Angeliki Metallinou, Dan Bohus, and Jason Williams. 2013. Discriminative state tracking for spoken dialog systems. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 466–475, Sofia, Bulgaria. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

S. G. Pulman. 1997. Conversational games, belief revision and bayesian networks. In *CLIN VII: Proceedings of 7th Computational Linguistics in the Netherlands meeting, Nov 1996*, pages 1–25.

Jun Quan, Meng Yang, Qiang Gan, Deyi Xiong, Yiming Liu, Yuchen Dong, Fangxin Ouyang, Jun Tian, Ruiling Deng, Yongzhi Li, Yang Yang, and Daxin Jiang. 2021. Integrating pre-trained model into rule-based dialogue management. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(18):16097–16099.

Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J. Liu, Sharan Narang, Wei Li, and Yanqi Zhou. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. Technical report, Google.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 1057–1063, Cambridge, MA, USA. MIT Press.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. Conditional generation and snapshot learning in neural dialogue systems. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2153–2162, Austin, Texas. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 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)*, pages 917–929, Online. Association for Computational Linguistics.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, WWW '21, page 1598–1608, New York, NY, USA. Association for Computing Machinery.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

A Appendices

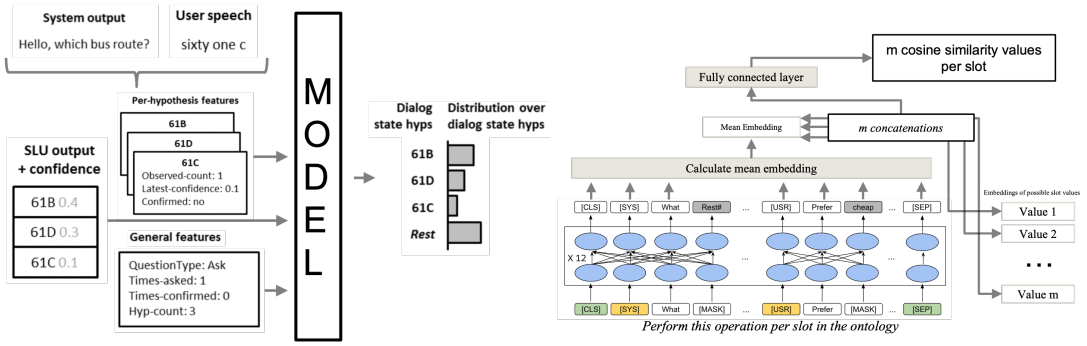

Figure A.1: Approach summary diagrams: (1) Maximum entropy model by (Metallinou et al., 2013) and (2) TOD-BERT by (Wu et al., 2020).

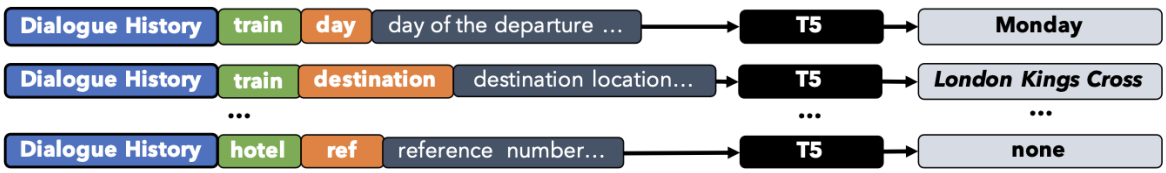

Figure A.2: Per slot natural language descriptions and dialogue history processed by T5 to predict the slot value. Diagram by Lee et al. (2021).