Unified Approach for Scalable Task-Oriented Dialogue System

Manisha Thakkar, Nitin Pise
Department of Computer Engineering and Technology, Dr Vishwanath Karad, MIT World Peace University, Pune, India

***Abstract*—Task-oriented dialogue (TOD) systems are currently the subject of extensive research owing to their immense significance in the fields of human-computer interaction and natural language processing. These systems assist users to accomplish certain tasks efficiently. However, most commercial TOD systems rely on handcrafted rules and offer functionalities in a single domain. These systems perform well but are not scalable to adapt multiple domains without manual efforts. Pretrained language models (PLMs) have been popularly applied to enhance these systems via fine-tuning. Recently, large language models (LLMs) have made significant advancements in this field but lack the ability to converse proactively in multiple turns, which is an essential parameter for designing TOD systems. To address these challenges, this paper initially studies the impact of language understanding on the overall performance of a TOD system in a multi-domain environment. Furthermore, to design an efficient TOD system, we propose a unified approach by leveraging LLM with reinforcement learning (RL) based dialogue policy. The experimental results demonstrate that a unified approach using LLM is more promising for scaling the capabilities of TOD systems with prompt adaptive instructions with more user friendly and human-like response generation.**

Keywords—Task-oriented dialogue system; unified; adaptive multi-domain; large language models; prompts

I. Introduction

Creating a dialogue system that has intelligence to converse like a human and assist in task completion is challenging. Depending upon the functional positioning of these systems in practice, are classified into two distinct types, Chit-chat systems also known as Open-domain dialogue systems, and Task-oriented dialogue (TOD) systems. Open-domain dialogue systems are not bound to any specific goal completion, and have flexibility to talk about any arbitrary topic, such as movies, sports, politics, etc. Open-domain dialogue systems are usually trained on large-scale social media data to engage users in human-like casual conversations. For example, ELIZA [1], which is the first open-domain dialogue system that plays the role of therapist; Parry [2], which acts like a psychology patient; and the recent chatbot, Xiaobing from Microsoft, which is a smart and emotionally aware open-domain dialogue system.

On the other hand, TOD systems are closed domain systems and have specific goals to be completed efficiently by assisting users. For instance, for tasks such as booking a flight or a taxi, scheduling an appointment, ordering food, etc., TOD systems are expected to ask questions proactively to accomplish well-defined user goals in minimum dialogue turns. This helps real users perform another important task to increase productivity. In the real world, these systems are utilized in various applications, such as QA systems at help desks to answer basic questions, and as pedagogical agents to assist in learning languages. On a day-to-day basis, users seek help from pretrained TOD systems such as Google Mini, Apple's Siri, Amazon's Echo, etc., to operate smart home devices, play music, and ask general questions to obtain answers [3]. In this study, our main focus was on TOD systems.

Most commercial dialogue systems have excelled in their ability to support singular domain functionalities [4]. To design such systems meticulously, handcrafted rules are used to understand the meaning of the sentence, to track the dialogue state in each turn, and to select the appropriate response. Domain experts participate in updating these rules to support each new task or domain. Each domain has a structured ontology that contains a set of predefined slot-value pairs. Consider an example of a restaurant TOD system that offers basic inquiry and booking related tasks. As shown in Table I, the domain ontology contains predefined slots for basic inquiry and booking tasks. Slot-value information is semantically represented as dialogue acts (DAs), which are updated in each turn during the slot-filling process. Any DA is either an inform act, a request act, or a greet act. Inform acts are used to inform user constraints from user queries to the dialogue system. Request acts are used by dialogue system to obtain additional information from user to fill needed slots, and greet acts are used to greet the user.

As shown in Table I, in basic DAs, primary information about restaurants is requested or informed by utilizing basic slots such as address, postcode, phone no, food type, price range (cheap, moderate, expensive), etc. Mandatory information for booking tasks such as number of people, number of days, and booking reference number are utilized by booking-related DAs.

Traditionally, the pipeline architecture of a TOD system has four components: natural language understanding (NLU), dialogue state tracking (DST), dialogue policy (POL), and natural language generation (NLG), as depicted in Fig. 1.

The NLU is responsible for recognizing user intentions by applying tokenization, and extracting information about domains, intents, slots, and values from user queries. This information is represented as a semantic frame, which is given to DST. This component keeps track of the slot-value pairs by maintaining a belief state in each turn along with the dialogue history. The current DST information is given as input to the

POL, which decides the next appropriate action, i.e., system DA, such as acknowledging user about the task completion or requesting additional information to fill mandatory slots. Finally, the NLG module generates a natural language response according to the system DA [6].

TABLE I. RESTAURANT DOMAIN ONTOLOGY [5]

Dialogue Act-type	basic acts	inform /request/ select/recommend/not found
	booking related acts	request booking info / offer booking/ inform booked / decline booking
	greet acts	welcome /greet / bye / reqmore
Slots	basic slots	address / postcode / phone/ name/ no of choices / area / price range / type / food
	booking slots	no of people / reference no / no of days

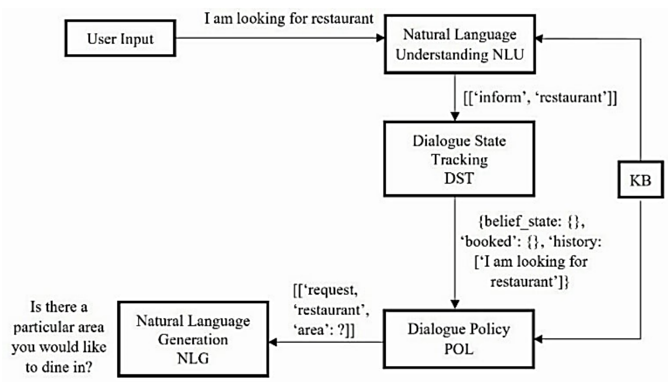


Fig. 1. Pipeline architecture of the TOD system.

To accomplish the intended task while comprehending user goals presents a formidable challenge. Commercial TOD systems are usually designed for specific domains using tools such as Microsoft's Power Virtual Agents (PVA) or Google's Dialog Flow. Consider a complex practical scenario where the user is asking for directions to the movie hall, and later, in the same dialogue session, the user wants to book a taxi to reach the hall. Here, both the navigation and taxi domains have a different set of (slot, value) pairs and actions. Contextual understanding is essential for understanding user intention for effective conversation in such dynamic scenarios. Recent advancements using PLMs and LLMs have made promising achievements in addressing complex real-world problems in natural language processing (NLP). This paper aims to achieve following objectives:

- To study and analyze the impact of language understanding on the overall performance of a TOD system in multi-domain conversation.
- To design a scalable TOD system with state-of-the-art NLU approaches and an RL-based dialogue policy instead of handcrafted rules.
- To enhance the TOD system by utilizing a unified LLM with instruction prompts for the NLU, DST and NLG tasks by boosting the convergence of RL-based dialogue policy with few samples of task demonstrations.

The paper is organized as follows: In Section II, the Literature Survey discusses previous work and recent advancements in the related field. Section III, Designing TOD Systems for multi-domain dialogues, elucidates design approaches for components of the dialogue system pipeline. In Section IV, Experimental Setup, provide details about the dataset, toolkit utilized, system configuration, and results of the experiments. Section V, Evaluation, presents a comprehensive comparison of the performance of dialogue systems designed with various approaches. Discussion is given in Section VI. Finally, Section VII concludes the paper.

II. LITERATURE SURVEY

TOD systems assist users in completing user intended tasks efficiently in a proactive manner. Traditionally, the TOD system components NLU, DST, POL are designed with handcrafted rules and a predefined sequence of words by the designers. Although rule-based systems perform well, scaling such systems is tedious and costly due to the necessity of re-designing rules to support new tasks or domains. Therefore, rule-based TOD systems are usually designed with limited task coverage in a specific domain with predetermined dialogue flow. Additionally, due to template-based response generation, these systems are less engaged with restricted language variability. Repeated or dumb responses are often generated by such systems in case of errors that cause user frustration. Therefore, when these systems are deployed for customer support, users often opt to converse with human agents directly.

Various statistical approaches have been studied to enhance the understanding of dialogue systems. Word presentation techniques such as bag-of-words (BoW), continuous BoW (CBoW), term frequency (TF), inverse document frequency (IDF), n-grams, and word2vec have been utilized to extract the meaning of the given input. The tasks of domain identification, intent detection, and policy selection are often treated as classification problems and slot labeling is treated as a sequence classification problem. Many studies have attempted different machine learning approaches for these tasks. The researchers [7] studied intent classification and slot-value labeling using an support vector machine (SVM) classifier. The researchers [8] studied intent classification by applying different machine learning approaches, including naïve Bayes, and SVM coupled with BoW. However, these approaches using machine learning and static embedding for word representation exhibit the following limitations.

- Curse of Dimensionality: The numeric representation of text results in a sparse matrix, which requires exponentially large amounts of memory, which impacts computational efficiency and model performance.
- Lack of Contextual information: In static embeddings, each word is embedded in isolation. Therefore, understanding the meaning of a word according to its context defined by its own position and that of other words, is not considered, which is crucial for designing NLUs.
- Dependence on the large amount of annotated data: TOD systems designed with traditional machine learning approaches rely on a large amount of annotated

task-specific data to achieve better task performance. Practically, using such data is not feasible.

- Human feedback is not undertaken: Traditional machine learning approaches do not consider human feedback, which is an essential parameter for selecting the next appropriate action in TOD systems. Therefore, such systems are not able to improve their performance from experience.

Due to deep learning (DL) advancements, input word representation has evolved from static embeddings to contextual word embeddings; for instance, pretrained bidirectional encoder representations from transformers (BERT) models consider contextual information in both the left and right directions. Additionally, various deep learning encoder-decoder based architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs) and long short term memory (LSTM), have been widely applied to solve various real-world problems in computer vision and conversational systems. [9] proposed a customer-facing dialog system by combining RNNs with domain-specific knowledge. Although RNNs have been widely used in TOD systems due to their ability to handle sequential data [10][11], have the following limitations. RNNs can capture dependencies exclusively in one direction, thereby neglecting the consideration of previous word dependencies while determining the meaning of subsequent words. Also, RNNs suffer from vanishing gradients while handling long-term dependencies, which are later mitigated with the help of gated recurrent units (GRUs) with additional memory units [12]. To this end, bi-directional LSTMs have been widely utilized to handle long- term dependencies in sequential data in many applications such as language translation tasks in NLP [13]. In their study, [14] performed joint classification of domain, intent and slot labeling by using bi-directional LSTM. The researchers [8] studied intent classification using SVM and bi-directional LSTM and found that bi-directional LSTM approach outperformed the traditional machine learning approaches.

Attention mechanisms significantly improved the encoder-decoder architecture [15]. This mechanism computes attention weights, which determine the amount of attention to be given to each word in the input sequence at each step. Transformers [16] are breakthrough advancements that use self-attention mechanisms for dependency modeling. BERT is an encoder-only transformer model, developed by Google AI that considers bi-directional context to predict masked words; thus, BERT has become a preferred choice for NLU tasks. BERT is pretrained on a large corpus from wiki and e-books, which offers generalization capabilities for basic understanding. In practice, the scarcity of human annotated data is one of the reasons for the work, proposed by [17] used BERT to demonstrate the ability of pretrained limited generalizability of these data to the NLU [14].

In their contextual embeddings for few-shot learning scenario. [18], [19] proposed joint training of intent classification and slot filling using an attention mechanism to significantly enhance the performance of dialogue systems. The study [20] demonstrated that BERT-based intent recognition outperforms other deep learning models including LSTM and RNN. The researchers [21] studied dialogue state tracking by applying BERT instead of using rule-based DST. Later, in these advancements, generative pretrained transformer (GPT)-based autoregressive decoder-only models, due to their ability to generate diverse responses, have received increased amounts of attention. The study [22] utilized GPT-2 for various TOD tasks, which resulted in more engaging and human-like responses. GPT-based models are popularly utilized for enhancing TOD tasks specifically response generation in NLGs [20], [23]. Until recently, PLMs with contextual embedding have been used as a starting point and subsequently fine-tuned for downstream tasks.

Recently, there has been a paradigm-shift from traditional model fine-tuning to prompt-tuning by efficiently utilizing a unified framework. In prompt-tuning, the network weights of the LLMs are frozen and a few task-specific demonstrations along with task prompts, are utilized to generalize with ease in few-shot settings. In their work, [24] used task-specific instruction prompts and approached various text processing tasks, such as sentiment analysis, question-answer generation, classification, etc., as text generation problems and referred to their model text-to-text transfer transformer (T5). Instruction based tuning of LLMs is gaining attention due to their improved communication capabilities achieved by providing hints to these LLMs about tasks [25]. LLMs have revolutionized dialogue with enhanced productivity across various industry domains. However, these models lack proactive communication, which is an essential parameter for handling multi-turn dialogue.

Selecting the next action in a dialogue flow to achieve the user goal in minimum dialogue turns is essential for evaluating the POL of a TOD system. Although rule-based policies perform well with fixed dialogue flow are not scalable for adapting to changes in the user goals. In real-world scenarios, the user is often uncertain about their goals at first place and wants to explore all available options. Additionally, dialogue POL should have the ability to learn new knowledge even after deployment. To design such scalable dialogue policies, researchers have studied the optimal action selection problem as a sequence of decision-making problems. In various studies, POL is implemented as a partially observable Markov decision process (POMDP) [26], by designing an RL-based dialogue agent to select the next action from the current dialogue state [27]. Such RL-based dialogue agents aim to maximize cumulative rewards by considering human feedback. These agents require more training cycles to learn from trial-and-error in a user-agent setting. Therefore, instead of real users, user simulators are used to train specifically in the initial stage of learning [28].

As PLMs and LLMs are large sized with huge number of parameters, should be efficiently used in TOD systems. This proposed work employs BERT contextual embedding for NLU and studies its impact on the overall performance of a TOD system. To achieve this goal, three distinct systems are configured using BERT, an SVM classifier and an RNN-based approaches for NLU and assessed their effectiveness. In further experiments, the rule-based POL is replaced with RL-based

POL and lastly, a unified approach using T5, and Llamma-2 is leveraged for the NLU, DST and NLG tasks.

III. Designing TOD System in Multi-domain Setting

In a diverse multi-domain environment, conversations include multiple tasks from different domains. The multi-domain ontology as shown in Eq. (1) has a set of slot-value pairs that are already defined in the respective domains to provide different functionalities.

$$(Multi\text{-}domain)\ MD_{ontology} = \left\{ domain_1: \begin{bmatrix} (slot_1,\ value_1), \\ (slot_2,\ value_2), \\ \vdots \\ (slot_i,\ value_i) \end{bmatrix}, domain_2: \begin{bmatrix} (slot_1,\ value_1), \\ (slot_2,\ value_2), \\ \vdots \\ (slot_i,\ value_i) \end{bmatrix}, \vdots\ domain_i: [(slot_i,\ value_i)], \right\} \quad (1)$$

The dialogues between the user and dialogue agent are either single-turn or multi-turn. The user utterance is represented as $User_u$ and the system utterance is represented as $System_u$. Therefore, a multi-turn conversation is shown as Eq. (2),

$$(Multi\text{-}domain,\ multi\text{-}turn)\ MDMT_{Dialogue} = \left\{ dialogue_i : domain_i: turn_i \left\{ User_{u\ i}, System_{u\ i} \right\} \right\} \quad (2)$$

The pipeline of NLU, DST, POL, and NLG process these dialogues by performing tokenization, extracting the semantic meaning, accessing database information, and generating responses. All these components can be trained and optimized separately using different approaches or by adopting an end-to-end (E2E) approach. In the E2E approach, two or more TOD components are combined for training and optimization using deep learning models. The following subsections describe the design of these components.

A. Natural Language Understanding (NLU)

The NLU identifies user intent from the input user query. BERT, a pretrained contextual embedding model, trained on two default training objectives, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks, is integrated into the NLU. In MLM, initially input tokens are randomly masked, and the model predicts the vocabulary ID of the masked tokens based on both left and right contexts. The input representation in BERT is the concatenation of word embeddings, position embeddings and segment embeddings. The first token of every sequence is a special classification token [CLS], which is pivotal for classifying intent. Another special token [SEP] is the last token in each sequence that separates two sentences. As shown in Fig. 2, intent recognition is approached as a classification problem to predict the intent class y^i. On the other hand, slot-filling is considered as a sequence labeling task to tag the input word sequence, $X = \{x_1, x_2, x_3, \cdots x_n\}$ with the slot label sequence given as $y^s_n = \{y^s_1, y^s_2, y^s_3, \cdots y^s_n\}$. The NLU represents this information in a semantic frame called dialogue act (DA).

Given the input token sequence as X,

$$X = \{ x_1, x_2, x_3,\ \cdots x_n \}$$

The output of BERT is H

$$H = \{ h_1, h_2, h_3,\ \cdots h_n \}$$

Based on the hidden state (h_1), Weights (W) and Bias (b) of the classification token [CLS], intent can be predicted as,

$$y^i = softmax\left(W^i h_1 + b^i\right) \quad (3)$$

The remaining hidden states from $h_2, h_3, \cdots h_n$ are used for slot filling as shown in Eq. (4). Each tokenized input word is given to the tokenizer, and the hidden state of the first token is fed to the softmax layer for classification. The slot filling prediction function is represented as,

$$y^s_n = softmax(W^s h_n + b^s)\ where\ n \in 1 \cdots N \quad (4)$$

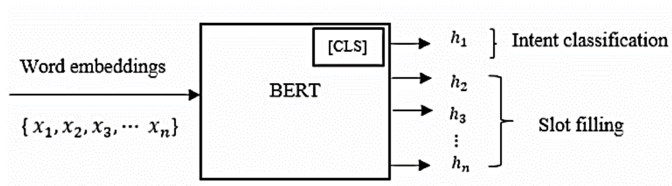


Fig. 2. BERT for generalization in the NLU.

The objective function for joint training of intent classification and slot filling is given as,

$$P\left(y^i, y^s_n \mid X\right) = P\left(y^i \mid X\right) \prod_{n=1}^{N} P\left(y^s_n \mid X\right) \quad (5)$$

The objective is to maximize the conditional probability $P\left(y^i, y^s_n \mid X\right)$ by minimizing the cross-entropy loss. In the case of slots with binary values such as Yes/No binary cross- entropy loss is used.

B. Dialogue State Tracking (DST)

DST utilizes information from NLU in a semantic frame known as dialogue acts (DAs) and maintains the belief state along with dialogue history. DST, which updates the belief state (intent, domain, slot, value) in each turn, is widely addressed as a classification task. The objective function for DST is to minimize the cross-entropy loss for both slot and slot value predictions. DST contains information about the constraints of the user, database search results, current user DAs, and previous system DAs. The following example demonstrates belief state updates at each turn in a multi-turn conversation. In proposed work, initially rule-based DST is used and subsequently the DST is approached as a text generation problem by employing the T5 model.

Turn 1 User: Can you find me a restaurant in the east?
belief state $_{t1}$= ["Inform", “restaurant", “**Location** ", "**east**"]

Turn 2 System: Sure, what type of cuisine are you looking for?
belief state $_{t2}$= ["Inform”: “restaurant", “Location”: "east", “**Cuisine**”:”?”]

Turn 3 User: I would like to have Indian food.
belief state $_{t3}$= ["Inform": “restaurant", “Location”: "east", “**Cuisine**”: “**Indian**”]

C. Dialogue Policy (POL)

The objective of dialogue policy is to accurately predict the next action by using the current dialogue state and generating system DAs in each turn with corresponding slot-value pairs.

In rule-based policy, the entire dialogue flow is hand coded whereas in RL-based dialogue policy, the training occurs in an agent-environment setting, which considers user feedback in terms of rewards. The RL-based dialogue agent aims to maximize the cumulative reward and improve from the experience. As these agents learn using trial and error, thousands of interactions are required for stabilization. Therefore, to train RL-based agents, user simulators that mimic real users are often required to interact before actual deployment for real users (Shi et al., 2019). In an agenda-based user simulator, the user goal is decomposed into slot-value pairs, whereas the agenda is maintained in a stack-like structure (Schatzmann et al., 2007). At a finite time-step T, the dialogue policy π is trained to maximize the cumulative reward in each turn. The cumulative reward as shown in Eq. (6) is assigned to an agent after dialogue completion. The optimal policy $\pi^*$is obtained using either value-based or policy-based methods. In our experiment, both rule-based and RL-based approaches are used to model dialogue policy.

$$R_t^{\pi} = \sum_{i=0}^{T-t-1} \gamma^t \; r_{t+i+1} \tag{6}$$

D. Natural Language Generation (NLG)

Once the policy determines system DA, the NLG task occurs in the following two steps: content planning followed by sentence realization. The content planning emphasizes ‘what to say’, and sentence realization focuses on ‘how to say in the correct manner’. The sentence realization is achieved using a de-lexicalization process in which system DAs are mapped to de-lexicalized sentences. This approach allows generation of dynamic sentences in different scenarios without hard-coded values.

In NLG, this template-based approach is commonly used to select the most appropriate template from the candidate set of already designed templates for response generation, are less engaging with limited language variability [29]. In proposed work a template-based NLG is initially used, and further text generation approach is used by employing T5 and LLM models.

E. Unified Approach for TOD System Pipeline using an LLM

Recently, all text processing problems have been approached as text generation problems. T5, and Llama2-chat-hf models from Hugging Face library, are used as unified frameworks for the NLU, DST and NLG tasks, as depicted in Fig. 3. Llama 2 chat is fine-tuned and optimized LLM for dialogue handling [30].

In prompt-tuning, only a small number of parameters are required to optimize the prompt that adapts an LLM to customized tasks or domains with frozen weight by preserving the general language understanding ability of LLM. To utilize the power of LLMs, instruction prompts are used to adapt NLU, DST, and NLG tasks in the TOD system.

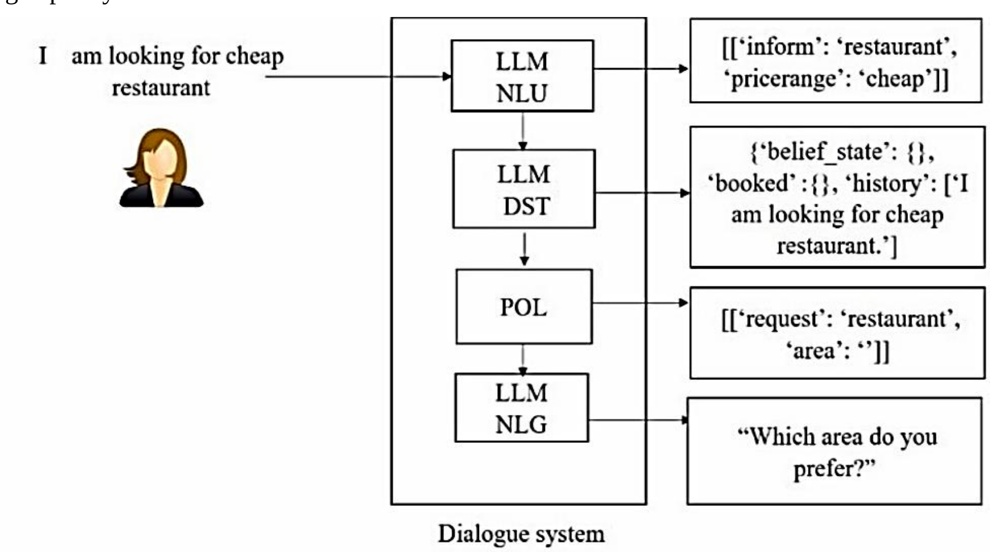


Fig. 3. Unified approach for the TOD system pipeline.

IV. Experimental Setup

Proposed work is focused on attaining scalability to support multiple domain conversation. To fulfill this essential requirement, a multi-domain, multi-turn human-to-human conversation dataset is selected after a survey [30] for designing a scalable TOD system. This section provides details about the utilized dataset, tools, system-configuration, and experimental results.

A. Dataset

A benchmark dataset MultiWOZ 2.1 is used for our research experiments. MultiWOZ 2.1 is a large dataset containing annotations for dialogue states, system dialogue acts, and user goals for training and evaluating dialogue systems in the context of tourist-related conversations. The MultiWOZ dataset consists of approximately 30 (domain, slot) pairs, encompassing over 45,000 values. The dataset has a size of 10,000 instances and covers various domains including Hotel, Hospital, Train, Taxi, Police, Postcode, and Restaurant [5].

MultiWOZ 2.1 contains more than 3,400 single-domain dialogues, and 7,032 multi-domain dialogues spanning across at least 2 to 5 domains. Most of the dialogues contain 10 turns on an average to meet the complexity of real-world scenarios.

B. Tools

Many tools have been implemented and integrated into an IDE for building dialogue systems. Often, these tools incorporate a rule-based dialogue manager (DM) with a built-in NLU component having a rigid structure. PyDial, ParlAI, Plato, Rasa, DeepPavlov, and ConvLab are examples of open-source tools with neural network-based dialogue managers that offer more flexibility and scalability. Convlab-3 [31] offers a range of state-of-the-art models for various TOD components and user simulators. In our study, Convlab-3 is used for experiment setup.

C. System Configuration

In the first experiment, distinct system agents are configured, each employing a different NLU approach. This allowed us to investigate the influence of the NLU on the overall performance of the dialogue system. All our experiments are performed using NVIDIA Tesla P100 GPU using the Google Colab platform with subscription.

An agenda-based user simulator is utilized for modeling a user agent, which is integrated with the BERT-base uncased model for NLU to evaluate the performance of system agents. The BERT-base uncased model utilizes a self-attention transformer-based encoder with 12 layers and 12 attention heads with a hidden size of 768 and a total of 110 M parameters.

Our experiment comprises five dialogue systems named SA1, SA2, SA3, SA4, and SA5. Each system has a rule-based DST and a template-based NLG. SA1, SA2, and SA3 have rule-based POL, whereas SA4 and SA5 have RL-based POL.

To understand the impact of NLU on the overall performance of dialogue system, different models for NLU. In SA1, BERT-base uncased model is used for joint training of intent classification and slot filling, whereas in SA2, utilizes the RNN-based joint neural model called Multi-Intent Language Understanding (MILU) for joint prediction of domain, intent, slot, and value. SA3 uses an SVM (support vector machine) classifier in the NLU [32] which is designed to manage complex semantic tuples (intent-slot-value) and classify them based on n-gram features. The listing of example code is referred from [32] for experiment setup as shown in Listing 1 . Similarly, in an extended experiment SA4 and SA5 TOD systems are designed by utilizing BERT and T5 for NLU tasks respectively with RL-based POL by referring to Listing 2 from [33].

In the second experiment, a unified approach is proposed to design a TOD system using T5 model. The NLU, DST and NLG are modeled using T5 with an RL-based dialogue policy. Further, this experiment is extended by utilizing the LLM from Huggingface library, meta-llama/Llama-2-13b-chat-hf with instruction prompts. Here, two LLM based systems are used to play the roles of 'user' and 'system'. The off-policy algorithm VTRACE [34] from checkpoint is utilized to model RL-based POL in both experiments.

```
# import necessary modules
# Create models for each component
# Parameters are omitted for simplicity
sys_nlu = BERTNLU(...)
sys_dst = RuleDST(...)
sys_policy = RulePolicy(...)
sys_nlg = TemplateNLG(...)

# Assemble a pipeline system named "sys"
sys_agent = PipelineAgent (sys_nlu, sys_dst, sys_policy, sys_nlg,
name="sys")

# Build a user simulator similarly but without DST user_nlu =
BERTNLU(...)
user_policy = RulePolicy(...)
user_nlg = TemplateNLG(...)
user_agent = PipelineAgent(user_nlu, None, user_policy, user_nlg,
name="user")

# Create an evaluator and a conversation environment
evaluator = MultiWozEvaluator()
sess = BiSession(sys_agent, user_agent, evaluator)

 # Start simulation sess.init_session()
sys_utt = ""
while True:
sys_utt, user_utt, sess_over, reward = sess. next_turn ( sys_utt)
if sess_over:
        break
print(sess.evaluator.task_success()) print(sess.evaluator.inform_F1())

# Use the analysis tool to generate a test report
analyzer = Analyzer(user_agent, dataset="MultiWOZ")
analyzer.comprehensive_analyze(sys_agent, total_dialog =1000)
# Compare multiple systems
sys_agent2 = PipelineAgent(MILU(...), sys_dst, sys_policy, sys_nlg,
name="sys") analyzer.compare_models(agent_list=[sys_agent,
sys_agent2], model_name=["bertnlu", "milu"], total_dialog=1000)
```

Listing 1. Example code from [32]

D. Results

Table II demonstrates the performance of each NLU for different multi-domain user queries. Table II highlights the domain and slot-value information for each query. BERT NLU outperforms the other two methods by demonstrating excellent performance in understanding long queries.

TABLE II. OUTPUT OF NLU IN DIALOGUE SYSTEMS USING BERT, SVM AND MILU APPROACHES

Query	Conversation	BERT NLU (SA1)	MILU (SA2)	SVM NLU (SA3)
1	I am looking for cheap food	[['Inform', 'Restaurant', 'Price', 'cheap']]	[['Inform', 'Restaurant', 'Price', 'cheap']]	[['Inform', 'Restaurant', 'Price', 'cheap']]
2	Can you suggest me Indian restaurants in westzone	[['Inform', 'Restaurant', 'Food', 'Indian'], ['Inform', 'Restaurant', 'Area', 'west zone']]	[['Inform', 'Restaurant', 'Food', 'Indian']]	[['Inform', 'Restaurant', 'Food', 'Indian']]
3	Give me address police station and contact details	[['Request', 'Police', 'Addr', '?'], ['Request', 'Restaurant', 'Addr', '?']]	[['Inform', 'Police', 'none', 'none']]	[['bye', 'general', 'none', 'none']]
4	I want to reach London kings cross by train TR1111	[['Inform', 'Train', 'Dest', 'London kings cross'], ['Inform', 'Train', 'Id', 'TR1111']]	[['Inform', 'Train', 'Dest', 'London kings cross']]	[['Inform', 'Train', 'Dest', 'London kings cross']]
5	Can you suggest me Indian restaurants in westzone with free parking. Also check for free Wi-Fi	[['Inform', 'Hotel', 'Parking', 'yes'], ['Inform', 'Hotel', 'Internet', 'yes'], ['Inform', 'Restaurant', 'Food', 'Indian'], ['Inform', 'Restaurant', 'Area', 'westzone']]	[['Inform', 'Restaurant', 'Food', 'Indian'], ['Inform', 'Hotel', 'Parking', 'yes'], ['Inform', 'Hotel', 'Internet', 'yes']]	[['Inform', 'Hotel', 'Internet', 'yes']]

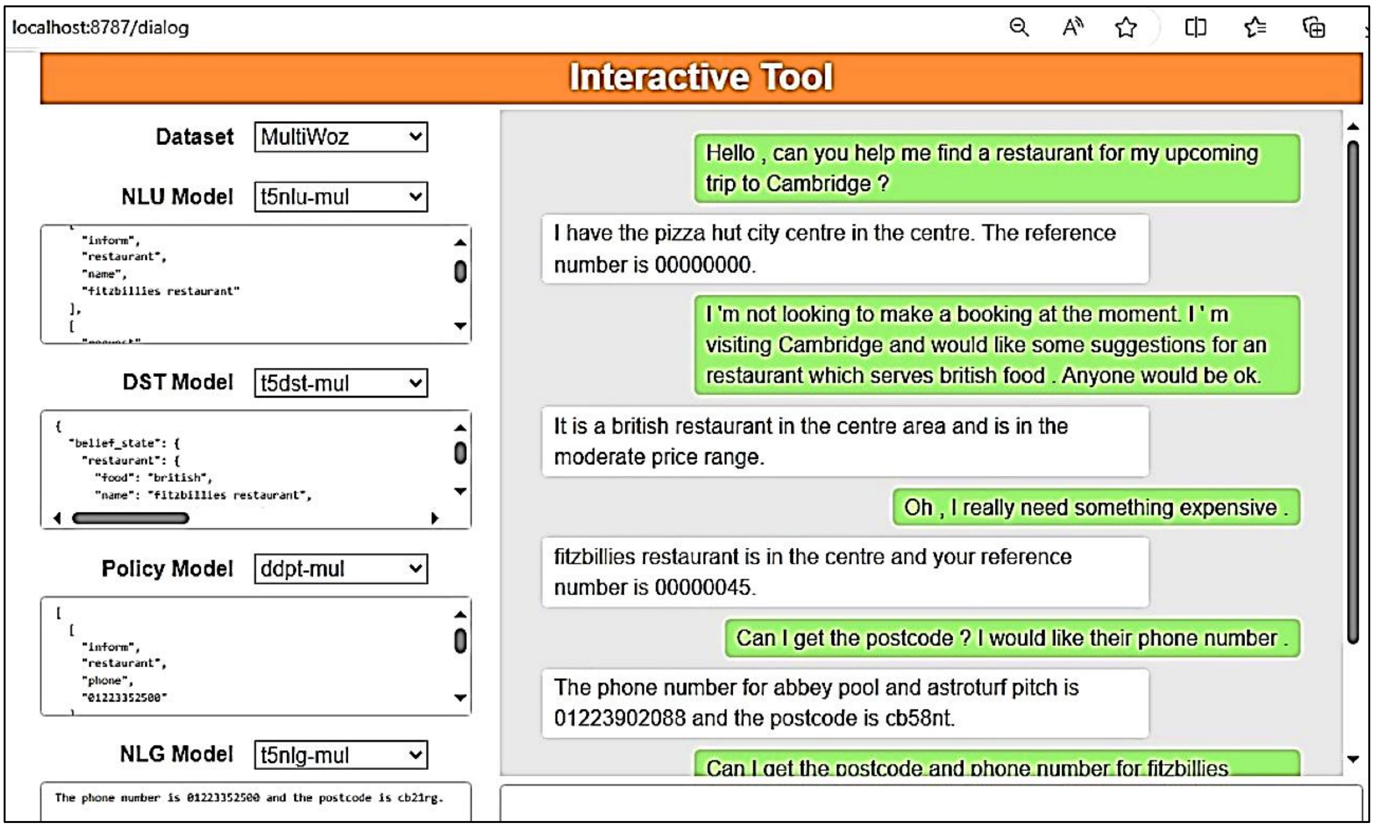


Fig. 4. Unified approach for modeling TOD tasks NLU, DST and NLG using T5.

In the second experiment, a unified approach with T5-small is used which was fine-tuned for both user and system utterances on the MutliWOZ2.1 dataset from the hugging face library. In this configuration, an RL-based policy is used for designing the TOD system. This encoder-decoder model completes NLU, DST and NLG tasks by approaching these tasks as text generation problems. The interaction between the system agent and real user is shown in Fig. 4. Further, the T5 model is replaced with the open-source Llama-2 model by utilizing task-specific instruction prompts to adapt the NLU, DST, and NLG tasks with sample demonstrations. Role-based instruction prompts are used in the meta-llama/Llama-2-13b-chat-hf model to define two roles, a system agent, and a user simulator to generate proactive multi-turn dialogues. The multi-turn interaction between the system agent and user simulator is shown in Table III. Domain information is highlighted for understanding.

TABLE III. UNIFIED APPROACH FOR MODELING TOD TASKS NLU, DST AND NLG USING LLAMA-2 13B-CHAT-HF

User	Assistant
	Hello, I am helpful assistant. How may I help you?
Hello! I am looking for a train from Norwich to Cambridge on Wednesday after 18:45. Can you help me find one and make a booking for one person? Also, Do you know of any places in Cambridge that offer multiple sports? Could you provide the phone number and postcode?	
	Hi there! I'd be happy to help. For a train from Norwich to Cambridge on Wednesday after 18:45, you can take the Greater Anglia train departing at 19.05 from Norwich railway station. To book, you can visit the Greater Anglia website or call their customer service at 0345 600 7245. As for sports facilities in Cambridge, the University of Cambridge Sports Centre offers a range of sports and can be reached at 01223336666 (postcode CB2 1RR)
Thank you so much! I've found the train I need and I'm going to book it now. Can you please provide me with reference number or the booking confirmation so I can have it handy? And thank you for the information on sports facilities in Cambridge, I'll definitely look into that [END]	
	Sure thing! Your train booking reference number is GA-123456. Please have this ready when you contact Greater Anglia to confirm your reservation. If you need any further assistance, feel free to ask. Enjoy your trip to Cambridge!

V. EVALUATION OF TOD SYSTEMS

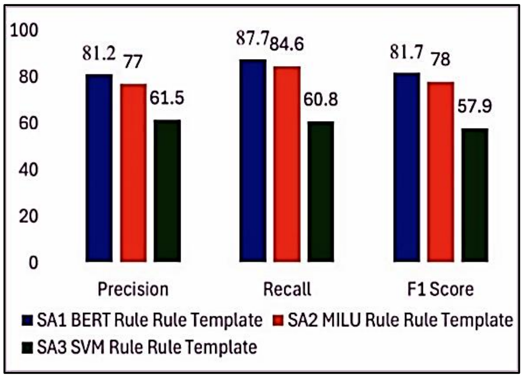
To measure the overall correctness of the search tasks in the inform act, such as domain, intent classification, and slot-filing, Precision, Recall and F1-score are used. Book rate (BR) measures the accuracy of the system for fulfilling booking requests and generating reference numbers. The success rate (SR) measures the successful completion of user goals in all required domains [35]. A user goal is considered successful when both the book rate and the recall of the informed act are 1. The average number of dialogue turns required for task completion impacts user satisfaction. The widely adopted evaluation matrices for TOD system are facilitated in convlab toolkit.

As depicted in Table IV, the SA1 system with the BERT NLU outperforms other rule-based system configurations. However, for the RL-based dialogue policy, the SA5 system with the T5 NLU outperforms the BERT-based system configuration. Compared to the other configurations, SA5 has the maximum completion rate.

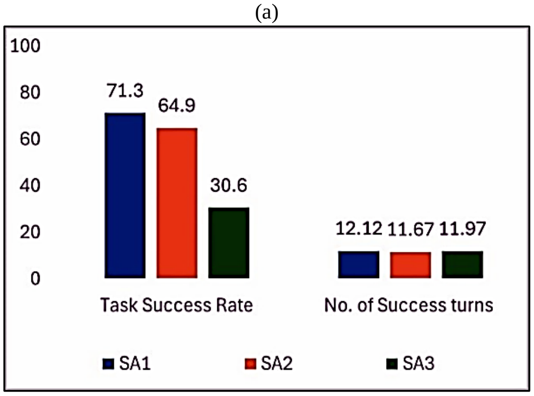
Fig. 5(a) depicts the NLU performance for the SA1, SA2 and SA3 dialogue systems and demonstrates that the BERT-based NLU has better precision, recall and F1-score than the MILU and SVM-based NLU. However, as depicted in Fig. 5(b), the BERT-based system resulted in an improved task success rate, with a slight increase in the average number of turns to achieve success compared to that of SA2 and SA3. SA1 is still taking longer to complete the task. As the NLU component is enhanced with state-of-the-art models including BERT and T5, for the next experiment rule-based dialogue policy is replaced with RL-based policy to adapt dynamic dialogue flow to converse in multi-domain setting.

TABLE IV. AUTOMATIC EVALUATION OF TOD SYSTEMS USING AN AGENDA-BASED USER SIMULATOR

System ID	System Configuration				Inform			Complete Rate	Task Success Rate	Book Rate BR	Average No. of Success turns/Average turns
	NLU	**DST**	**POL**	**NLG**	**P**	**R**	**F1**				
SA1	**BERT**	Rule	**Rule**	Template	**81.2**	**87.7**	**81.7**	**78.9**	**71.3**	**88.4**	12.12/16.51
SA2	MILU				77	84.6	78	73.7	64.9	83.3	**11.67/16.56**
SA3	SVM				61.5	60.8	57.9	44.4	30.6	51.4	11.97/16.59
SA4	BERT	Rule	**RL based POL**	Template	64.2	86	70	71	30	62.2	**17.06/25.16**
SA5	**T5**				64	**93.7**	**72.8**	**85**	**53**	**84.3**	21.05/25.32



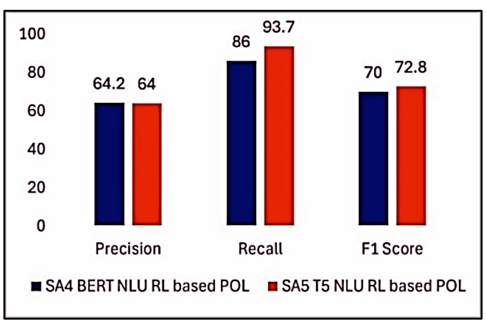
(a)



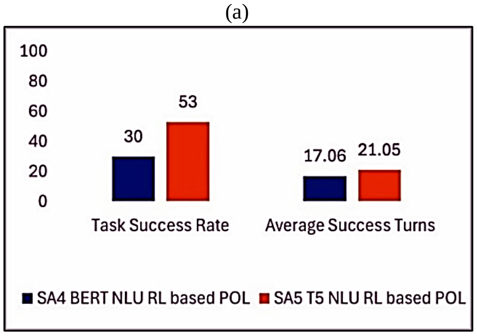
(b)

Fig. 5. (a) NLU performance in SA1, SA2 and SA3 (b) task success rate and average success turns in SA1, SA2 , and SA3.

As depicted in Fig. 6(a), the T5-based NLU outperforms the BERT-based NLU, whereas Fig. 6(b) indicates that SA5 has an improved task success rate at the cost of an increased number of average success turns, indicating slow convergence compared to that of SA4.



(a)



(b)

Fig. 6. (a) NLU performance in SA4 and SA5 (b) task success rate and average success turns in SA4, and SA5.

VI. Discussion

In real-world scenarios, designing a proactive, multi-turn dialogue system to understand and satisfy user goals in minimum dialogue turns is a complex task. Additional complexity is introduced when the user goal contains tasks from multiple domains. This led us to analyze the impact of different NLUs on the performance of the TOD system. Transformer based approaches outperformed traditional machine learning algorithms including SVM[8] and RNN[11], followed by additional experiments with RL-based policies instead of using rule-based policies to avoid handcrafted rules. BERT- based NLU with the rule-based policy demonstrated improved performance with a slight increase in the average number of successful dialogue turns. Furthermore, the transformer-based models including BERT and T5 are used for the NLU with the RL-based policy. The T5 NLU with the RL-based policy resulted in significant improvements in the recall, F1-score, complete rate, and book rate with an increase in the average number of successful dialogue turns. This indicates limited improvement in fast convergence of task completion. Therefore, although NLU performance is boosted, RL-based POL requires investigation for improvement in task performance through warm-up and task-specific pretraining to achieve the task in minimum dialogue turns.

Additional experiments are performed to improve TOD system performance by using a unified approach instead of improving the performance of individual components. This has a major obstacle to the availability of annotated task-specific data for pretraining. Until recently, fine-tuning PLMs achieved promising performance on TOD tasks. However, an entire PLM model with many parameters is required for gradient updating to adapt to each downstream task; for instance, the large number of BERT-base-uncased NLUs is 110M. Domain-specific pretraining is required to achieve better performance because the PLMs are trained on general purpose data.

Our experiments found that even after improving the performance of a single component, the overall performance improvement in TOD systems is not guaranteed. On the other hand, a unified approach using a single PLM or LLM with shared parameters across all tasks is more preferred approach. To adapt to a new task or domain, prompt-enabled models such as T5 and Llama-2 are more efficient as only a small number of parameters are updated in prompt tuning with just a few demonstrations. On the other hand, benchmark TOD systems designed using fine-tuning approach such as UBAR, GALAXY, MinTL needs to load entire PLMs such as GPT-2(1.5B), UniLM (340M), BART-large (440M) respectively to update large number of parameters to adapt each new task or domain [35].

The responses generated in the LLM model Llama-2 have human-like language variability, which demonstrates its ability to design more user-friendly TOD systems in the future. Other LLMs, such as GPT3.5 and GPT-4, have provided many scalability and multimodality features, but these models are accessible only with paid subscriptions.

VII. Conclusion and Future Work

Understanding user intentions from natural language text in a dynamic environment remains an inherently challenging task. Various existing TOD systems contain rule-based components designed to function in single domain and offer limited tasks to users. To design scalable TOD system for conversation which includes multiple domains with different tasks to offer is challenging. In line to our first objective to study the impact of NLU, we have configured SVM, RNN and state-of-the pretrained language models such as BERT and T5 instead of using handcrafted (rule-based) NLU to understand user intention in multi-domain tourist conversation environment. We found that configuring BERT and T5 in NLU enhances the performance of an individual component but does not guarantee an overall performance improvement in TOD system.

In further step, we extended our experiment with best performing NLUs (BERT, T5) in a pipeline TOD and replaced the rule-based designing dialogue policy with more scalable approach by employing reinforcement learning (RL) algorithm to adapt in multi-domain conversation. We found that handing large state-action spaces requires large commuting power, and training RL-based dialogue policy in such a large dynamic environment takes many training cycles by the dialogue agent to learn from scratch. As the agent gains experience using trial and error method, establishing a stable dialogue policy is time-consuming. Also, to train these RL-based agents a reliable user simulator is required with added design efforts. In our experiments, we utilized already existing agenda-based user simulator for automatic evaluation of TOD systems provided in the toolkit. We achieved very less task success rate which indicates, further investigation is needed to boost or warm-up the performance of dialogue policy using methods such as task-specific pretraining, fine-tuning, inverse reinforcement learning (IRL), and imitation learning (IL) approaches.

Recently, unified approach is utilized popularly by employing large language models to perform all TOD tasks but fine-tuning these models is costly. Instead of fine-tuning language models for each individual task recent trend encourages to utilize prompt-enabled large language models. In this paper, soft prompts are generated using system instructions to achieve proactive multi-turn dialogues by assigning different roles to LLMs, such as assistant and system agents. The proposed approach is adaptive and generates more human-like responses compared to other systems, paving the way for scalable and user-friendly dialogue systems.

References

[1] J. Weizenbaum, “ELIZA-A computer program for the study of natural language communication between man and machine,” Commun. ACM, vol. 9, no. 1, pp. 36–45, 1966, doi: 10.1145/365153.365168.

[2] K. M. Colby, F. D. Hilf, S. Weber, and H. C. Kraemer, “Turing-like Indistinguishability Tests for the Calidation of a Computer Simulation of Paranoid Processes,” Artif. Intell., vol. 3, pp. 199–221, 1972, [Online]. Available: https://api.semanticscholar.org/CorpusID:31542633.

[3] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan, “Towards Scalable Multi-domain Conversational Agents: The Schema-Guided Dialogue Dataset,” 2019, [Online]. Available: http://arxiv.org/abs/1909.05855.

[4] M. Henderson, B. Thomson, and J. Williams, “The second dialog state tracking challenge,” SIGDIAL 2014 - 15th Annu. Meet. Spec. Interes. Gr. Discourse Dialogue, Proc. Conf., no. June, pp. 263–272, 2014, doi: 10.3115/v1/w14-4337.

[5] M. Eric et al., “MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines,” Lr. 2020 - 12th Int. Conf. Lang. Resour. Eval. Conf. Proc., pp. 422–428, 2020.

[6] Z. Zhang, R. Takanobu, Q. Zhu, M. Huang, and X. Zhu, “Recent advances and challenges in task-oriented dialog systems,” Sci. China Technol. Sci., vol. 63, no. 10, pp. 2011–2027, 2020, doi: 10.1007/s11431-020-1692-3.

[7] F. Mairesse, M. Gaˇ, and T. Street, “Spoken Language Understanding From Unaligned Data Using Discriminative Classification Models C , F . Jurˇ c ´ ı c ˇ ek , S . Keizer , B . Thomson , K . Yu , and S . Young,” vol. 2.

[8] J. Schuurmans and F. Frasincar, “Intent Classification for Dialogue Utterances,” IEEE Intell. Syst., vol. 35, no. 1, pp. 82–88, 2020, doi: 10.1109/MIS.2019.2954966.

[9] J. D. Williams, K. Asadi, and G. Zweig, “Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning,” ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., vol. 1, pp. 665–677, 2017, doi: 10.18653/v1/P17-1062.

[10] C. Geishauser et al., “Dynamic Dialogue Policy for Continual Reinforcement Learning,” Proc. - Int. Conf. Comput. Linguist. COLING, vol. 29, no. 1, pp. 266–284, 2022.

[11] S. Lee et al., “ConvLab: Multi-domain end-to-end dialog system platform,” ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Syst. Demonstr., pp. 64–69, 2019, doi: 10.18653/v1/p19-3011.

[12] C. S. Wu, A. Madotto, E. Hosseini-Asl, C. Xiong, R. Socher, and P. Fung, “Transferable multi-domain state generator for task-oriented dialogue systems,” arXiv, no. 808, pp. 808–819, 2019.

[13] S. Merity, N. S. Keskar, and R. Socher, “Regularizing and Optimizing LSTM Language Models,” arXiv, 2017.

[14] Y. B. Kim, S. Lee, and K. Stratos, “ONENET: Joint domain, intent, slot prediction for spoken language understanding,” 2017 IEEE Autom. Speech Recognit. Underst. Work. ASRU 2017 - Proc., vol. 2018-Janua, pp. 547–553, 2018, doi: 10.1109/ASRU.2017.8268984.

[15] A. Vaswani et al., “Attention is all you need,” Adv. Neural Inf. Process. Syst., vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[16] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186, 2019.

[17] C. S. Wu, S. Hoi, R. Socher, and C. Xiong, “TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue,” EMNLP 2020 - 2020 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf., pp. 917–929, 2020, doi: 10.18653/v1/2020.emnlp-main.66.

[18] A. Gupta, J. Hewitt, and K. Kirchhoff, “Simple, fast, accurate intent classification and slot labeling for goal-oriented dialogue systems,” arXiv, no. September, pp. 46–55, 2019.

[19] S. Louvan and B. Magnini, “Recent Neural Methods on Slot Filling and Intent Classification for Task-Oriented Dialogue Systems: A Survey,” COLING 2020 - 28th Int. Conf. Comput. Linguist. Proc. Conf., pp. 480–496, 2020, doi: 10.18653/v1/2020.coling-main.42.

[20] V. Khan and T. A. Meenai, “Pretrained Natural Language Processing Model for Intent Recognition (BERT-IR),” Human-Centric Intell. Syst., vol. 1, no. 3–4, p. 66, 2021, doi: 10.2991/hcis.k.211109.001.

[21] V. Balaraman and B. Magnini, “Domain-Aware Dialogue State Tracker for Multi-Domain Dialogue Systems,” vol. 1, 2020, [Online]. Available: http://arxiv.org/abs/2001.07526.

[22] P. Budzianowski and I. Vuli, “Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems,” no. Wngt, pp. 15–22, 2019.

[23] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap., vol. 1, pp. 328–339, 2018, doi: 10.18653/v1/p18-1031.

[24] KEKEKE et al., "T5: Exploring the limits of transfer learning with a unified text-to-text transformer," J. Mach. Learn. Res., vol. 21, pp. 1–67, 2020.

[25] V. Hudeček and O. Dušek, "Are LLMs All You Need for Task-Oriented Dialogue?," 2023, [Online]. Available: http://arxiv.org/abs/2304.06556.

[26] S. Young, M. Gašić, B. Thomson, and J. D. Williams, "POMDP-based statistical spoken dialog systems: A review," Proc. IEEE, vol. 101, no. 5, pp. 1160–1179, 2013, doi: 10.1109/JPROC.2012.2225812.

[27] J. Gao, M. Galley, and L. Li, "Neural approaches to conversational AI," ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. Tutor. Abstr., pp. 2–7, 2018, doi: 10.18653/v1/p18-5002.

[28] Y. Dai, H. Yu, Y. Jiang, C. Tang, Y. Li, and J. Sun, "A Survey on Dialog Management: Recent Advances and Challenges," arXiv, 2020, [Online]. Available: http://arxiv.org/abs/2005.02233.

[29] B. Byrne et al., "Taskmaster-1: Toward a realistic and diverse dialog dataset," arXiv, pp. 4516–4525, 2019.

[30] H. Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models," 2023, [Online]. Available: http://arxiv.org/abs/2307.09288.

[31] Q. Zhu, C. Geishauser, H. L. Carel, X. Zhu, J. Gao, and M. Gaši, "ConvLab-3 : A Flexible Dialogue System Toolkit Based on a Unified Data Format."

[32] Q. Zhu et al., "ConvLab-2: An Open-Source Toolkit for Building, Evaluating, and Diagnosing Dialogue Systems," 2020, [Online]. Available: http://arxiv.org/abs/2002.04793.

[33] Q. Zhu et al., "ConvLab-3: A Flexible Dialogue System Toolkit Based on a Unified Data Format," EMNLP 2023 - 2023 Conf. Empir. Methods Nat. Lang. Process. Proc. Syst. Demonstr., pp. 106–123, 2023, doi: 10.18653/v1/2023.emnlp-demo.9.

[34] L. Espeholt et al., "IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures," 35th Int. Conf. Mach. Learn. ICML 2018, vol. 4, pp. 2263–2284, 2018.

[35] J. Deriu et al., Survey on evaluation methods for dialogue systems, vol. 54, no. 1. Springer Netherlands, 2021.