Using Textual Interface to Align External Knowledge for End-to-End Task-Oriented Dialogue Systems

Anonymous submission

Abstract

Traditional end-to-end task-oriented dialogue systems have been built with a modularized design. However, such design often causes misalignment between the agent response and external knowledge, due to inadequate representation of information and lack of interaction. Furthermore, its evaluation metrics emphasize assessing the agent's pre-lexicalization response, neglecting the quality of the completed response. In this work, we propose a novel paradigm that uses a textual interface to align external knowledge and eliminate redundant processes. We demonstrate our paradigm in practice through MultiWOZ-Remake, including an interactive textual interface built for the MultiWOZ database and a correspondingly re-processed dataset. We train an end-to-end dialogue system to evaluate this new dataset. The experimental results show that our approach generates more natural final responses and achieves a greater task success rate compared to the previous models.

1 Introduction

Task-oriented dialogue (TOD) systems have been extensively studied for various applications that involve natural language interactions between users and machines. These systems are designed to accomplish specific tasks, such as booking a hotel, ordering food, or scheduling an appointment. The traditional paradigm for building such systems is to use a modularized design (Pieraccini et al., 1992; Young, 2006; Young et al., 2013; Zhao and Eskénazi, 2016), where a dialogue state is maintained across modules to track the progress of the conversation and to interact with external databases. It generally incorporates Dialogue State Tracking (DST), database query or API calls, Natural Language Generation (NLG), and lexicalization to create the final system response.

However, the traditional modularized paradigm faces several limitations. Firstly, it struggles to represent and integrate external knowledge effectively, as the modules operate independently, without a common knowledge grounding. Secondly, the traditional paradigm heavily relies on delexicalization, resulting in annotations that are rigid and exhibit inconsistencies. Also, current evaluation metrics primarily focus on assessing the agent's pre-lexicalization response, neglecting the performance of the system as a whole, which compromises the end user experience. Consequently, this modularized design has become a significant impediment in developing more effective end-to-end task-oriented dialogue systems.

To address these limitations, we propose a new TOD paradigm that is Textual Interface Driven (TID) to better represent external knowledge and coordinate the interactions from the agent. We instantiate our proposal using the MultiWOZ (Budzianowski et al., 2018a) dataset to demostrate the differences. As the original MultiWOZ dataset only contains limited annotations collected for the traditional paradigm, we re-process it into MultiWOZ-remake by transforming the annotations into interface states and agent actions. This new dataset simulates agent interactions over the textual interface, ensuring complete alignment of external knowledge representation with agent responses. We also build an end-to-end dialogue agent for this dataset to demonstrate the effectiveness of our proposed paradigm.

In our experiments, we expose the problem of evaluating delexicalized responses with the commonly used metrics of 'Inform' and 'Success'. Instead, we evaluate the final lexicalized responses with BLEU to better reflect the performance of the end-to-end system. To more thoroughly assess the system, we conduct a comprehensive human evaluation. Compared against strong baselines, our system generates more natural responses and achieves a higher task success rate, thereby showcasing the superiority of our proposal.

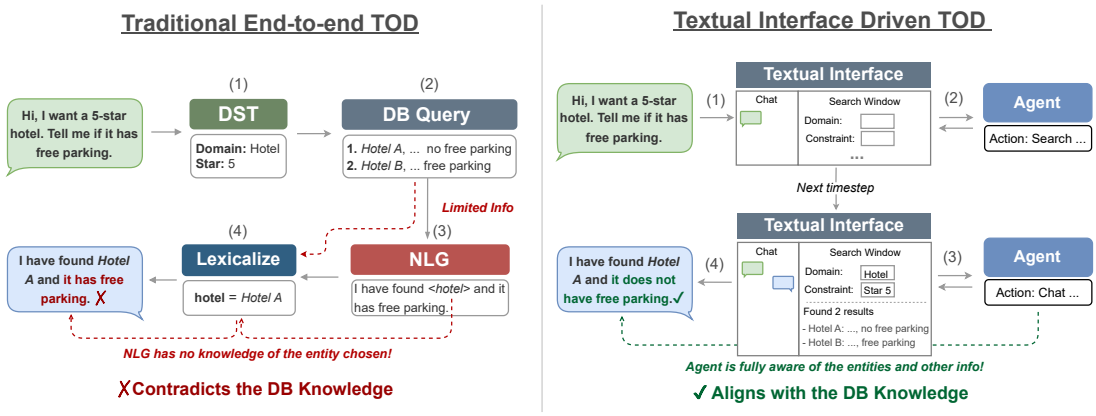

Figure 1: A comparative illustration of Traditional End-to-end TOD systems versus Textual Interface Driven TOD systems. This example highlights how the traditional pipeline may induce misalignment between the generated response and the corresponding database entity. Meanwhile, in our pipeline, the agent can interact with the textual interface iteratively for superior knowledge representation. More details are in Section 3.

2 Related Work

The most common task-oriented dialogue paradigm is the dialogue state paradigm, or slot-filling paradigm (Pieraccini et al., 1992; Young, 2006; Young et al., 2013; Zhao and Eskénazi, 2016). It typically consists of several modular components, including a natural language understanding module that extracts user intents and relevant information for the task (Hashemi et al., 2016; Shi et al., 2016), dialogue state tracking module which tracks the current state of the conversation (Kim et al., 2017), a dialogue policy module for learning dialogue acts, and a natural language generation module to generate the system response.

The MultiWOZ dataset (Budzianowski et al., 2018b) extends this paradigm by providing comprehensive annotations for building different dialogue systems (Wu et al., 2019, 2021; Gu et al., 2021; Lee, 2021; Hosseini-Asl et al., 2020; Yu et al., 2022). However, the traditional task-oriented dialogue system paradigm has limitations in effectively representing the external knowledge. In this work, we address these limitations and remake MultiWOZ with our proposed paradigm.

3 Textual Interface-Driven TOD

Our textual interface-driven (TID) approach effectively circumvents the limitations of the traditional modularized design, where each module requires a specific schema for inter-module communication, leading to ineffective knowledge representation and error propagation throughout the conversation. In contrast, our model leverages a *unified* textual interface, serving as a precise and comprehensive front-end for knowledge representation.

In the following subsections, we outline the implementation of the textual interface using the document tree, and then present a comparative illustration between the traditional paradigm and our proposed one. Finally, we show the construction of an end-to-end dialogue agent for our interface.

3.1 Interface with Document Tree

To better represent information, we utilize a virtual document tree to implement the textual interface, similar to the document object model (Keith, 2006) employed in HTML, where each node can represent part of the document such as a title, text span, or a list. This approach captures the document's structure as a hierarchical, tree-like object. It also helps to separate the presentation of content from its underlying structure and behavior, making it easier to update the interface representation. To preserve formatting and structural information, we further render the document tree into Markdown. Markdown is a lightweight markup language that is used for formatting text. It provides a simple and easy-to-use syntax for creating headings, lists, and other elements, and it is designed to be easy to read and write (Mailund, 2019). This rendered Markdown text will serve as the state representation as the dialogue system's inputs. The detailed explanation and illustrations are in Appendix A.2.

3.2 Comparison with Traditional Paradigm

Figure 1 provides a comparative illustration of the traditional paradigm versus our proposed paradigm.

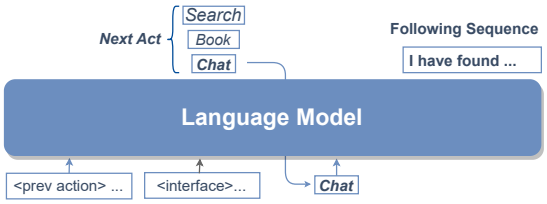

Figure 2: In our end-to-end model, the agent initially predicts the next action, and then generates the following sequence based on that action.

In most of the traditional TOD implementations, there are four main stages. Initially, a user's input is processed by (1) a dialogue state tracking (DST) module, extracting the user's intentions and beliefs. Subsequently, (2) a database (DB) query is conducted using the extracted intents and belief states. Next, (3) a natural language generation (NLG) module creates a delexicalized response, exemplified in the figure. Finally, (4) in the next step, the placeholders in the delexicalized response are replaced with actual entities information derived from the database query.

However, such a design spreads dialogue states across the modules, causing difficulties in syncing database information with the actual generated response. In the provided example, the system's response inaccurately reflects that 'Hotel A' has free parking, yet the NLG module is unaware of the specific entity chosen. Similar misalignment can occur with the DST module, especially when managing booking requests, as it may lack knowledge of the previously selected entity.

In contrast, our interface-driven paradigm avoids misalignment by having a *shared* textual interface to coordinate all the information flow between the user and agent. (1) A user's utterance updates the interface's state. (2) The agent then determines the next action (Search). (3) When enough information is collected, the agent choose to 'Chat' through the interface after the 'Search' action, and (4) the final response is delivered to the user. The agent is fully aware of the entity displayed on the interface and can generate a consistent and cogent response based on its selected entity.

3.3 End-to-End TOD Agent

To interact with the textual interface, we build a model that is compatible with most task-oriented dialogue datasets. The input context contains the previous action and the current textual interface state. The model needs to first predict the next action. It includes three main next actions: *"Chat"*, *"Book"* and *"Search"*. Then, the predicted action is fed back to the model. *"Chat"* continues to output the generated sequence to the chat window, while *"Search"* and *"Book"* updates the search constraint or booking information displayed in the search window with the following generated sequence. This setting is compatible with different language models including encoder-decoder models.

4 MultiWOZ Remake

We remake the existing MultiWOZ dataset (Budzianowski et al., 2018b) to showcase the usefulness of our proposed paradigm. We implemented a textual interface to interact with the database and re-processed the dataset accordingly.

The textual interface follows the interface-driven paradigm and utilizes the document tree design. For each of the seven domains present in MultiWOZ, we design a different sub-section in the interface based on the query domain. The interface as a front-end displays necessary details such as query domain, constraints, the number of entities found, and booking status. An example can be found in Figure 3 in Appendix A.

The original MultiWOZ dataset did not record the selected entities during the conversation, leading to a misalignment between the interface representation and the actual response. Therefore, we need to re-process the dataset to replay the agent's actions on the interface, thereby ensuring alignment between the selected entity and the interface representation. The details of re-processing are shown in Appendix A.3.

5 Experiments

We conducted several experiments on the MultiWOZ test set to evaluate our end-to-end dialogue agent. We tested different back-bone encoder-decoder models including BART (Lewis et al., 2019), T5 (Raffel et al., 2020), and GODEL (Peng et al., 2022) to compare with previous models, by fine-tuning them on the re-processed MultiWOZ Remake training set to compare with the baselines.

For automatic evaluation, Inform and Success have problems in reflecting the performance of TOD systems. One can use a fixed response "[value_name] [value_phone] [value_address] [value_postcode] [value_reference] [value_id]" to easily get the state-of-the-art performance on

Model	Backbone	#Parameters	BLEU
HDSA	BERT_{base}	110M	11.87
MTTOD	GODEL_{base}	360M	13.83
	GODEL_{large}	1.2B	13.06
GALAXY	UniLM_{base}	55M	13.71
Mars	T5_{base}	220M	13.58
Remake	BART_{base}	140M	15.87
	BART_{large}	406M	15.82
	T5_{base}	220M	15.27
	T5_{large}	770M	16.66
	GODEL_{base}	220M	16.55
	GODEL_{large}	770M	**16.92**

Table 1: BLEU results for lexicalized responses.

Model	Backbone	Goal Success (%)
MTTOD	GODEL_{base}	47.6
	GODEL_{large}	38.1
Remake	GODEL_{large}	**90.5**

Table 2: Human Evaluation for Goal Success.

Comparison	Win	Lose	Tie
Remake vs. MTTOD_{base}	57.1%	0.0%	42.9%
Remake vs. MTTOD_{large}	52.4%	4.8%	42.8%

Table 3: Human evaluation for coherence.

MultiWOZ[1]. See more details in Appendix B. Therefore, we only used sacreBLEU (Post, 2018) [2] to evaluate the lexicalized responses of various task-oriented dialogue systems. We also conduct human evaluation for a better comparison.

5.1 Automatic Evaluation

We compared Remake with strong baselines including HDSA (Chen et al., 2019), MTTOD (Lee, 2021), GALAXY (He et al., 2022b), and Mars (Sun et al., 2022). Note that we evaluated the quality of final lexicalized responses. We used the lexicalization script provided by Hosseini-Asl et al. (2020) to fill in the placeholders for the baselines' outputs.

As shown in Table 1, Remake models with the new paradigm achieve better performance than the baseline models (HDSA, MTTOD, GALAXY, and Mars)with the traditional dialog state paradigm. Especially, although HDSA (Chen et al., 2019) has the best reported BLEU score performance with delexicalized responses, it gets worse performance after lexicalization. This observation suggests that our paradigm model can greatly improve the quality of final lexicalized responses.

5.2 Human Evaluation

As mentioned previously and in Appendix B, automatic evaluation metrics can be misleading. Thus, we conduct human evaluations to better evaluate the performance improvement of our model Remake compared to MTTOD, as it is the strong baseline with supervised learning.

The evaluators interact with each model for 21 whole conversations using the same goal instructions from the MultiWOZ dataset. On average, each conversation finishes within ten turns. Then, they rate the models in terms of two metrics: "Goal Success" and "Coherence". Each conversation gets two annotations. "Goal Success" measures if the system can successfully satisfy the user's goal without given any information contradicting to the database. "Coherence" measures if the system responses are coherent and human-like.

Table 2 shows the human evaluation results for goal success. The Remake model demonstrates a significantly higher level of accuracy ($p < 0.01$), achieving 90.5% goal success, compared to the 47.6% accuracy of the MTTOD model. This improvement suggests that the use of an interface can help the system reduce hallucinations and better satisfy the user's request.

Table 3 shows the human evaluation results for coherence. "Win" indicates that the dialogue looks more coherent, whereas "Lose" means the opposite. Remake is significantly more coherent than MTTOD ($p < 0.01$ with paired t-test). We observe that, in the context of an entire conversation, MTTOD struggles to maintain entity consistency, resulting in incoherent dialogues.

6 Conclusion

In conclusion, we have proposed a novel textual interface driven paradigm for building task-oriented dialogue systems. The new paradigm better aligns external knowledge and the final system response. We implemented the paradigm with presenting MultiWOZ-Remake, an interactive interface built for the MultiWOZ database, and a corresponding dataset. Experimental results show that our system in this new paradigm generates more natural responses and achieves a greater task success rate compared against the previous models.

[1]We used the official scoring script: https://github.com/Tomiinek/MultiWOZ_Evaluation

[2]The official BLEU script.

7 Limitations

One major limitation of our proposed paradigm is that the interface controls how the information is displayed to the model and maintains all the states internally. Therefore, the interface becomes the most important component of the system. A poorly designed interface can hurt a model's performance as it limits the model's ability to access the necessary information to make accurate decisions or take appropriate actions.

Another limitation is with the MultiWOZ's evaluation metrics. We mainly evaluate our model with BLEU after lexicalization, as we pointed out the problem of Inform and Success rates. In the future, we will design better automatic evaluation metrics to test the performance of task-oriented dialogue systems.

References

Rahul Kumar Agarwal, Ikhlaq Hussain, and Bhim Singh. 2017. Implementation of LLMF control algorithm for three-phase grid-tied SPV-DSTATCOM system. *IEEE Trans. Ind. Electron.*, 64(9):7414–7424.

Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. Task-optimized adapters for an end-to-end task-oriented dialogue system. *CoRR*, abs/2305.02468.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018a. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018b. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Derek Chen and Zhou Yu. 2022. Sources of noise in dialogue and how to deal with them.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3696–3709. Association for Computational Linguistics.

Jing Gu, Qingyang Wu, Chongruo Wu, Weiyan Shi, and Zhou Yu. 2021. PRAL: A tailored pre-training model for task-oriented dialog generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 305–313, Online. Association for Computational Linguistics.

Homa B Hashemi, Amir Asiaee, and Reiner Kraft. 2016. Query intent detection using convolutional neural networks. In *International Conference on Web Search and Data Mining, Workshop on Query Understanding*.

Wanwei He, Yinpei Dai, Min Yang, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022a. SPACE-3: unified dialog model pre-training for task-oriented dialog understanding and generation. *CoRR*, abs/2209.06664.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2022b. GALAXY: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10749–10757. AAAI Press.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *CoRR*, abs/2005.00796.

Tianjian Huang, Shaunak Ashish Halbe, Chinnadhurai Sankar, Pooyan Amini, Satwik Kottur, Alborz Geramifard, Meisam Razaviyayn, and Ahmad Beirami. 2023. Robustness through data augmentation loss consistency. *Trans. Mach. Learn. Res.*, 2023.

Jeremy Keith. 2006. *DOM scripting: web design with JavaScript and the Document Object Model*. Apress.

Seokhwan Kim, Luis Fernando D'Haro, Rafael E Banchs, Jason D Williams, and Matthew Henderson. 2017. The fourth dialog state tracking challenge. In *Dialogues with Social Robots*, pages 435–449. Springer.

Yohan Lee. 2021. Improving end-to-end task-oriented dialog system with A simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 1296–1303. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Thomas Mailund. 2019. *Introducing Markdown and Pandoc: using markup language and document converter*. Apress.

Tomás Nekvinda and Ondrej Dusek. 2021. Shades of bleu, flavours of success: The case of multiwoz. *CoRR*, abs/2106.05555.

Baolin Peng, Michel Galley, Pengcheng He, Chris Brockett, Lars Liden, Elnaz Nouri, Zhou Yu, Bill Dolan, and Jianfeng Gao. 2022. Godel: Large-scale pre-training for goal-directed dialog.

Roberto Pieraccini, Evelyne Tzoukermann, Zakhar Gorelov, Jean-Luc Gauvain, Esther Levin, Chin-Hui Lee, and Jay G. Wilpon. 1992. A speech understanding system based on statistical representation of semantics. In *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '92, San Francisco, California, USA, March 23-26, 1992*, pages 193–196. IEEE Computer Society.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Kun Qian, Ahmad Beirami, Satwik Kottur, Shahin Shayandeh, Paul A. Crook, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. 2021. Database search results disambiguation for task-oriented dialog systems. *CoRR*, abs/2112.08351.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Yangyang Shi, Kaisheng Yao, Le Tian, and Daxin Jiang. 2016. Deep LSTM based feature mapping for query classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1501–1511, San Diego, California. Association for Computational Linguistics.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. Mars: Semantic-aware contrastive learning for end-to-end task-oriented dialog. *CoRR*, abs/2210.08917.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 808–819. Association for Computational Linguistics.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2021. Alternating recurrent dialog model with large-scale pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1292–1301. Association for Computational Linguistics.

Steve J. Young. 2006. Using pomdps for dialog management. In *2006 IEEE ACL Spoken Language Technology Workshop, SLT 2006, Palm Beach, Aruba, December 10-13, 2006*, pages 8–13. IEEE.

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

Xiao Yu, Qingyang Wu, Kun Qian, and Zhou Yu. 2022. Krls: Improving end-to-end response generation in task oriented dialog with reinforced keywords learning.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. Description-driven task-oriented dialog modeling. *CoRR*, abs/2201.08904.

Tiancheng Zhao and Maxine Eskénazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 1–10. The Association for Computer Linguistics.

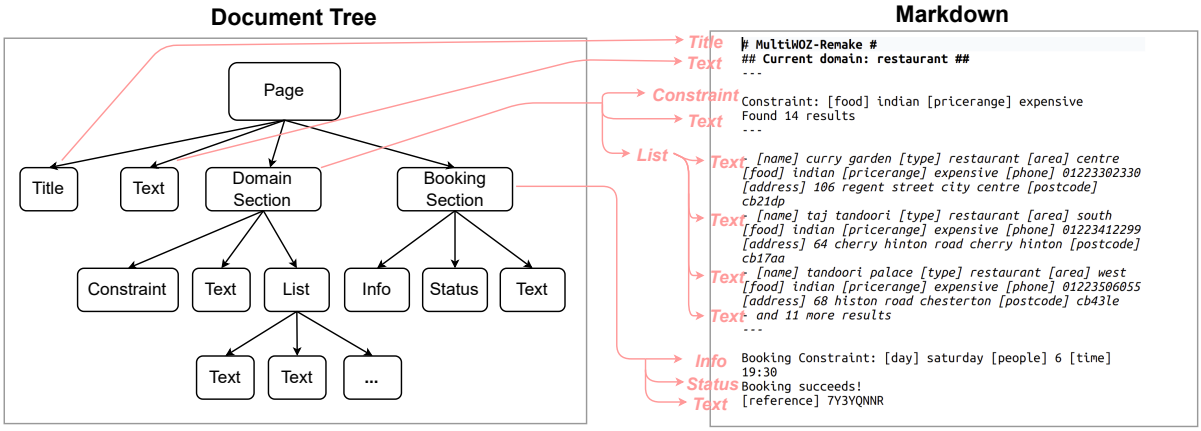

Figure 3: An example of rendering the Document Tree representation into the textual representation in Markdown. Document tree simplifies the manipulation of the dynamic elements in the interface, while Markdown is to display richly formatted text. This approach provides both the flexibility and comprehensibility for the interface.

A MultiWOZ-Remake Details

A.1 Textual Interface

MultiWOZ's interface defines the front-end and the back-end functions. The front-end displays necessary details such as query domain, constraints, the number of entities found, and booking status. It also displays a truncated list of presently searched entities for agent selection. Meanwhile, the back-end handles SQL search calls, utilizing current and prior constraints entered into the interface to identify appropriate entities, and it also verifies booking availability and, if successful, returns a reference number.

The proposed interface for the MultiWOZ database implements two types of search commands: searching with constraints, and booking using provided information. To simplify the complexity of the command, the interface uses the incremental belief state between two turns as the query action. In the back-end, a cumulative belief state is used to perform the actual SQL search.

The interface can be interacted with by providing a command *"[domain] [slot] value"* or *"[booking] [slot] value"*. *"[slot] value"* is optional if only domain switch is performed. The command would update the internal constraint for querying the database and refresh the dynamic elements showing in the interface. For example, in Figure 3, the interface state can be reached by performing two actions: *"[restaurant] [food] indian [pricerange] expensive"* and *"[booking] [day] saturday [people] 6 [time] 19:30"*. It is important to note that there can be multiple different paths of actions to reach the same interface state, allowing for flexibility in how the agent interacts with the interface.

A.2 Document Tree

Figure 3 provides how MultiWOZ's interface is implemented and it highlights the transformation between the Document Tree and Markdown. This illustration provides a visual representation of how the different components of the interface correspond to the Document Tree and Markdown. The right part of figure 3 shows an example of the interface in the restaurant domain. It highlights the flexibility of using Document Tree to manipulate dynamic elements such as "Domain Section", "Booking Section", or "List", and then it can be rendered into the Markdown text that is comprehensible for both humans and language models.

A.3 Data Re-Processing Details

We use MultiWOZ 2.2 (Zang et al., 2020) as it provides necessary annotations to help us re-process the dataset. Specifically, we track entities from previous dialogue history, ensuring alignment between the query domain, search constraints, and selected entities. In particular, the entity chosen during booking should correspond with the actual booked entity. However, it is possible that the mentioned entities in the training data's response are truncated from the list to avoid a long context. To avoid this situation, we re-arranged the database search results so that the mentioned entities are always shown in the interface display, which minimizes the hallucination with the correct entity grounding. Also, when multiple domains are involved in a single turn, we divide this turn into multiple actions

Backbone	Prev Act	Next Act	Search	BLEU
BART_{base}	✗	62.0	3.8	11.24
BART_{large}	✗	57.8	4.2	13.51
BART_{base}	✓	92.1	77.4	15.87
BART_{large}	✓	**95.2**	77.5	15.82
GODEL_{base}	✓	95.0	76.6	16.55
GODEL_{large}	✓	95.1	**79.7**	**16.92**

Table 4: Action prediction results and ablation studies. "Next Act" means the next action's prediction accuracy. "Search" means the search query accuracy.

Model	Inform	Success
MTTOD	85.9	76.5
GALAXY	85.4	75.7
Mars	88.9	78.0
KRLS	89.2	80.3
fixed response*	**89.2**	**88.9**

(a) End-to-end models

Model	Inform	Success
Gold (human)	93.7	90.9
KRLS	93.1	83.7
MarCO	**94.5**	87.2
GALAXY	92.8	83.5
fixed response*	92.5	**92.3**

(b) Policy optimization

Table 5: Inform and Success scores. * a fixed response is used: "[value_name] [value_phone] [value_address] [value_postcode] [value_reference] [value_id]"

to ensure completeness.

For the booking data processing, we used multiple sources of information, including the span annotation for the booking reference number, dialogue as annotations to provide booking status, and information from belief states to determine whether a booking takes place at the current turn. Additionally, we aligned the interface's representation with the recorded booking outcome, whether it is a success or failure. Therefore, the interface can correctly display the booking status when handling the booking action.

A.4 Re-Processing Inconsistency

It is important to note that there can be inconsistency between the training data and the re-processed data by replaying the trajectories on the interface. If some entities in the response cannot be inferred from the history context, we recognize it as a inconsistent dialogue. This normally happens due to the annotation errors (Chen and Yu, 2022) in MultiWOZ, or the complex scenarios when multiple domains are involved.

There are 2373 out of 10438 dialogues that are potentially inconsistent during data processing. Then, we randomly sampled 250 dialogues and manually classify them to check the consistency. We observed that single-domain dialogues have 74% consistency and multiple-domain dialogues have 43% consistency, suggesting that multiple-domain dialogues are more complex. Also, we found that the consistency is relatively disproportional to the length of turns. To minimize noise, these dialogues are excluded from training.

B Problems of Inform and Success

Task-oriented dialogue systems often use Inform and Success to evaluate the quality of response generation. However, they are designed for delexicalized responses like "[value_name] is a restaurant...", and it needs further lexicalization process to fill in the placeholders like "[value_name]". As a result, they are not reflecting the real quality of the final response.

Furthermore, we question the validity of the current Inform and Success metrics implementation. They check the cumulative belief states for placeholders and whether the response contains a reference to calculate the scores. Consequently, a model that generates more placeholders achieves a deceptively better performance.

To illustrate this, we use the same fixed response "[value_name] [value_phone] [value_address] [value_postcode] [value_reference] [value_id]" for every turn when evaluating on the standardized evaluation script (Nekvinda and Dusek, 2021) to report the performance. We compare both end-to-end and policy optimization models. In the end-to-end setting, we use the dialogue state prediction from Mars (Sun et al., 2022).

Table 5 shows the results with this fixed response. Surprisingly, the fixed response achieves state-of-art performance on the Inform and Success score compared to the baseline models. It is questionable whether Inform and Success can measure the true performance of the system, and they may mislead the existing models. We urge future researchers to stop reporting Inform and Success until a better evaluation metric is proposed.

Figure 4: Dialogue examples that address the problems of ambiguity and hallucination. We compare Remake and MTTOD's outputs. **Green** represents the correctly generated spans. **Red** represents the incorrect ones.

Models	Joint Acc.
SimpleTOD (Hosseini-Asl et al., 2020)	55.8
AG-DST (Agarwal et al., 2017)	57.3
Space-3 (He et al., 2022a)	57.5
D3ST (Zhao et al., 2022)	58.7
DAIR (Huang et al., 2023)	60.0
TOATOD (Bang et al., 2023)	63.8
Remake + Exact match	54.2
Remake + Fuzzy match	**74.8***

Table 6: Dialogue state tracking results. * means the joint accuracy after multiple interactions.

C Dialogue State Tracking

Dialogue State Tracking (DST) is an important module in traditional end-to-end dialogue systems. There have been many previous works (Hosseini-Asl et al., 2020; Agarwal et al., 2017; He et al., 2022a; Zhao et al., 2022; Huang et al., 2023; Bang et al., 2023) studied this problem, and they have achieved great performance on this task.

However, in this work, DST is not our main focus, as we can apply any existing DST module and replace the search query in our system. Our method focuses more on the alignment of the final response generation. Also, our textual interface implementation can support fuzzy match and interactive searching, so that the agent can find the correct entities even with small typos or missing some slots. As shown in Table 6, if we apply fuzzy match, the joint goal accuracy can boost up to 74.8. Note that other DST methods can potentially use the same method to improve the performance, but they often do not consider the end-to-end scenarios where fuzzy match can be helpful. It needs more investigation in the future to study how the accuracy of DST is reflected in the final response. For example, booking slots may require more attention, but name typos may be less important and can tolerate some typos.

D Next Act Prediction

We perform ablations of our model for the Next Act prediction accuracy and the Search prediction accuracy. The search query accuracy measures if the system generates the correct sequence when searching the database when performing *"Search"* action. Table 4 shows the final results. The backbone with GODEL-large achieves the best overall performance. Note that without the previous action

in the context, the model is unaware of its previous action and performs not well, which suggests the importance of the history state in our paradigm.

We also conducted the error analysis for the wrongly predicted search action with three categories prediction error, annotation error, and ignore type, and they are denoted as: Type I: Prediction Error where the model makes a wrong prediction, Type II: Annotation Error; and Ignore: Error which can be ignored.

Error Types	Percentage
Type 1: Prediction Error	40.0%
Type 2: Annotation Error (Labeling)	6.0%
Type 2: Annotation Error (Discourse)	2.0%
Ignore	52.0%

Table 7: Percentage of different errors.

For the prediction error, the common mistake is forgetting to predict one of the intents requested by the user. Sometimes, this can be due to mispredicting attributes that require reasoning. Also, searching for "train" domain requires attributes like destination and departure to be all revealed. For example, if the user says "I want to book the restaurant for tomorrow.", then the agent needs to transfer that into the actual value represented in the database. For the annotation errors, we further divided them into labeling errors, ontology and inconsistencies, and discourse errors as suggested by Chen and Yu (2022). The labeling errors occur when the states are under-labeled or over-labeled while discourse attributes are when the dialogues show occurrences of inconsistency or incoherence.

We randomly select 50 errors and classify them into those categories. Table 7 shows the results. We can observe that most errors can be ignored. However, the model still accounts for a large portion of the errors, suggesting that the model needs further improvement in terms of search.

E Case Studies

Figure 4 shows two dialogue examples chatting with Remake and MTTOD, respectively. It demonstrates the common problems of the traditional dialog state paradigm. The first problem is handling ambiguity in the user's utterance, which is previously studied by Qian et al. (2021). MTTOD cannot handle such requests very well as the lexicalization process involves no understanding. The same situation can happen when the user says "what about another restaurant?"

Another type of problem is hallucination. Models like MTTOD often use the number of the returned database results to represent the grounding of the database. As a result, it cannot handle complex questions from the user. In this example, "Cityroomz" does not offer free parking at all, but MTTOD hallucinates to provide the wrong information to the user. It suggests the necessity of using our paradigm to provide knowledge grounding for the model to avoid this case.