Graph Integrated Language Transformers for Next Action Prediction in Complex Phone Calls

Amin Hosseiny Marani and **Ulrike Schnaithmann** and **Youngseo Son** and
Akil Iyer and **Manas Paldhe** and **Arushi Raghuvanshi**
Infinitus Systems, Inc.
{amin.hosseiny, ulie.schnaithmann, youngseo.son, akil.iyer,
manas.paldhe, arushi}@infinitus.ai

Abstract

Current Conversational AI systems employ different machine learning pipelines, as well as external knowledge sources and business logic to predict the next action. Maintaining various components in dialogue managers' pipeline adds complexity in expansion and updates, increases processing time, and causes additive noise through the pipeline that can lead to incorrect *next action prediction*. This paper investigates graph integration into language transformers to improve understanding the relationships between humans' utterances, previous, and next actions without the dependency on external sources or components. Experimental analyses on real calls indicate that the proposed Graph Integrated Language Transformer models can achieve higher performance compared to other production level conversational AI systems in driving interactive calls with human users in real-world settings.

1 Introduction

Building and maintaining complex production quality conversational systems has been an ongoing challenge in industry. One approach to solve complex conversational tasks such as outbound call automation, is to use a dialogue manager (Paek and Pieraccini, 2008; Teixeira et al., 2021) to encode business logic. Conversational systems which use dialogue managers have multiple components which consist of Natural Language Understanding (NLU) (Bocklisch et al., 2017), dialogue state tracking (Mannekote, 2023), next action prediction (Mannekote, 2023), and response generation (Weston et al., 2022; He et al., 2018). Figure 1 describes the process of call automation systems with the aforementioned components.

Handling *next action prediction* is one of the critical tasks dialogue managers take care of, as it affects the response generation directly (David, 2017). *Next action prediction* is the process of analyzing human utterance, current and previous state

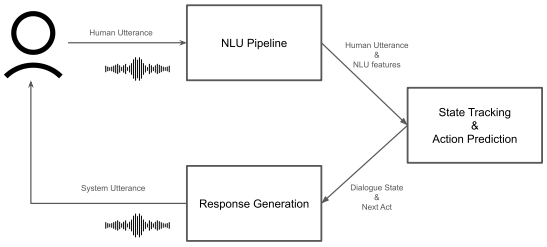

Figure 1: A schematic visualization of dialogue managers' components which utilize an NLU pipeline of models to extract intents and fill slots from human utterances, and predict the next action based on the current and previous state. Finally, the system generates an utterance to respond to the human users (e.g., using LLMs or predefined templates).

of the conversation (i.e., dialogue state tracking) and deciding which action to take, which in many industry settings is returning a specific response template. Figure 2 demonstrates an example of a dialogue manager based conversation automation as a visual navigation assistant for multiple dialogue turns.

Recently, there has been significant progress in the field of Generative AI and Large Language Models (LLMs) for end-to-end conversational systems which alleviate the need for manually engineered dialogue managers (Mannekote, 2023; Snell et al., 2022). However, they sometimes have issues with hallucinations and can underperform in domain specific, targeted conversations such as those that require knowledge graph retrieval (Dziri et al., 2021; Ji et al., 2023).

In most industry settings, templates are used with action prediction to generate the response. By predicting an action, we are determining which response template(s) to return to the user (Mannekote, 2023; Qiu et al., 2022; Urbanek et al., 2019). Action prediction using response templates instead of

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), pages 347–358

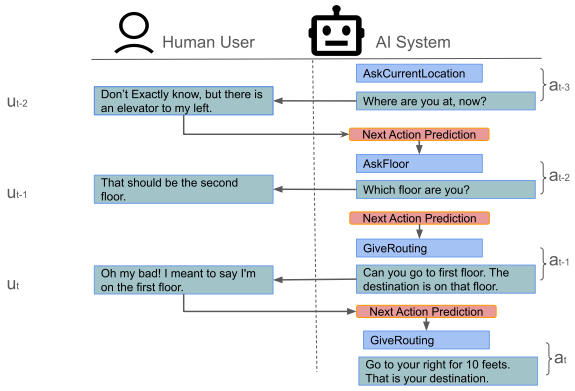

Figure 2: An example of Visual Navigation Assistant as a dialogue manager. At each time-step t, the dialogue manager extracts the entities such as slots and intents from human utterance u_t (i.e., green rectangles on the left side) and predicts the next action a_t (i.e., red rectangles). Using the predicted action (i.e., blue rectangles) the dialogue manager generates a system response (i.e., green rectangles on the right side).

language generation helps prevent hallucinations, adds necessary guardrails for some industry settings, and keeps latency low.

To solve the *next action prediction* problem, different NLP methods from traditional symbolic AI techniques such as Knowledge Graph models (He et al., 2017; de Vries et al., 2018), to more modern transformer based techniques (e.g., Zhou et al., 2023) have been introduced; however, two main challenges still persist. 1) a majority of prior work depends on Slot-Filling (SF) and Intent-Classification (IC) techniques to extract dependencies and relies on external sources (i.e., knowledge or rule based approaches) to find the relationship between the extracted information and actions (Mannekote, 2023; David, 2017). Instability in detecting SF and IC causes incorrect *next action prediction*. 2) many conversational systems handle grounding poorly (David, 2017; Weston et al., 2018; Sutskever et al., 2014); this is when users' responses differ from expected inputs (e.g., referring to a previous point in the conversation, moving backwards to change a previous response, or no action-related slots being detected). For example, in Figure 2, the human user sets a new ground by mentioning the elevator instead of the their location. This information may be slightly different than what a *next action prediction* model expects and can respond to. Lack of grounding in a conversation and more specifically in a model may result in misunderstanding (David, 2017) and can damage the conversation.

This paper introduces an approach to predict the next action without any dependency on information extraction (i.e., SF and IC) or external resources [1] such as ontology (e.g., Altinok, 2018) or knowledge-base (e.g., Vizcarra and Jokinen, 2022) approaches. The proposed models, Graph Integrated Language Transformers, learns co-occurrences of actions and human utterances through a graph component (i.e., Graph Neural Network or a graph embedding layer) and combines it with language transformers to add language understanding in production settings. The model is trained on conversations that followed a Standard Operating Procedure (SOP) [2] without the need for explicit encoding. The proposed model can be trained on any similar dataset that has an inherent action-to-action relationships. The list below summarizes the contribution of this paper.

- Integrating graph information and combining with language transformers to remove dependency on NLU pipelines.
- Adding a graph component (i.e., history of action co-occurrence) to language transformers to predict the next action as one atomic task while also overcoming the token limit by removing the need to keep prior dialogue history.
- Evaluating the proposed *next action prediction* model in a production setting against a system that relies on an NLU pipeline with an explicitly defined dialogue manager (DM system) in Appendix A.

To examine the performance and robustness of the proposed models in real-world settings with noisy input, the evaluation is done in a production setting and goes beyond classification metrics; the evaluation includes industry critical factors such as human experience using the conversational system and considers real-time constraints such as latency of output generation.

2 Related Work

Next action prediction approaches can be categorized in three chief groups. First, structured-

[1] The proposed model is trained using external resources but does not need any external resources after training.

[2] SOP is a document which defines a set of guideline instructions for diverse situations during the conversations.

based approaches that consider sequential relationships between previous actions, other actions, and their requirements. These approaches assume that the current state (i.e., the previous action) is known (Henderson, 2015). On the one hand, local structure-based approaches such as *Question & Answer* systems (Reshmi and Balakrishnan, 2016) consider local adjacency of the actions, utterance features, and next potential actions. On the other hand, global structured-based approaches define problem space using dialogue-grammars or finite-state networks (David, 2017; Wollny et al., 2021). However, none of structured-based approaches provide the ability to train a model and they require expert to design them (Henderson, 2015).

The second group of *next action prediction* approaches are principle-based. These techniques choose next actions based on the filled information rather than sequential order between actions, thus behaving both locally and globally (David, 2017). Slot-filling (SF) and Intent-classification (IC) based techniques (i.e., joined or separate components) are common principle based approaches (Louvan and Magnini, 2020).

Recently, neural models including RNNs and Language Transformers which act solely on input are receiving more attention for SF-IC based techniques (Goo et al., 2018; Chen et al., 2019; Zhang and Wang, 2022). These methods are mainly using dialogue history alongside additional information such as schema of the task (e.g., "hotel booking" or "scheduling a doctor's appointment") e.g., using embedding layers with or without attention layers fused with a language transformer (e.g., Mosig et al., 2020; Mehri and Eskenazi, 2021; Zhang et al., 2021). However, most of these language transformer based techniques were only evaluated on datasets with low number of actions, 10 or less (Mosig et al., 2020; Rastogi et al., 2020), or perform poorly on larger number of actions (i.e., 30 actions) for one top output selection (Chen et al., 2021).

3 Methodology

This section discusses the problem definition of the *next action prediction* task (i.e., Section 3.1), and introduces the proposed models (i.e., Section 3.2).

3.1 Problem Definition

A *next action prediction* model chooses an action a_t given $U_{k:t}$ and $Z_{k:t-1}$ at time t in which U is the set of all utterances from time k (i.e., $k \geq 0$) to time t, and Z is the set of all previously predicted acts. Equation 1 formulates the process of *next action prediction*. In this equation, f denotes any function (e.g., machine learning model or a probabilistic matching technique) that can map thereof inputs to the next action.

$$a_t = f([U_{k:t}, Z_{k:t-1}]) \quad s.t. \quad 0 \leq k \leq t-1 \quad (1)$$

Different techniques approach *next action prediction* differently. Some techniques rely on feature extraction from utterances (i.e., $U_{k:t}$) using NLU techniques (e.g., intents or slots in NLU pipeline of Figure 1); in those cases U_t in Equation 1 becomes utterance and all those extracted features at time t. However, this paper proposes a method that relies only on the very last human utterance and previous actions in Section 3.2.

3.2 Graph Integrated Language Transformers

This paper proposes a graph integrated approach to employ the rich information of graph-like structures, discussed in Section 2 (e.g., SOP, graphs, or rule knowledge bases) and combine it with language transformers. Two different techniques are proposed in this section that each combine language transformers with 1) Graph Neural Networks (GNN) to explicitly encode the graph of actions and other features (GNN-LT), and 2) a graph embedding layer to learn co-occurrences of action history, Graph-aware Language Transformer (GaLT).

Both models additionally use language transformers such as BERT (Devlin et al., 2018), DistilBERT (Sanh et al., 2019), or RoBERTa (Liu et al., 2019) to add language understanding (Devlin et al., 2018) to the *next action prediction*. The GNN-LT models is fed past actions as nodes and features of nodes' connections as edges (i.e., order of the connections, slots, and embedding of the utterance) using a Graph Attention Network (Yun et al., 2019). Thus, GNN-LT explicitly integrates the graph knowledge including the order of the actions and their connections. GaLT employs a graph embedding layer that encodes past actions as node labels directly without the past action names or utterances; therefore implicitly adds the ability to learn the co-occurring utterances and actions without the need to explicitly enforce graph constraints (i.e., actions as nodes, filled slots or other features as edges). Additionally, GaLT acquires fewer training parameters (e.g., $66M$ Distilbert + $1M$ fusion

and fully connected layer = $67M$ in total) in comparison to GNN-LT (e.g., $66M$ Distilbert + $12M$ Graphormer small (Yun et al., 2019) + $1M$ fusion and fully connected layer = $79M$ in total); therefore, GaLT requires less training time and performs much faster in inference.

The language transformer is fed the human utterance alongside the history of actions to implicitly learn the co-occurrence between human responses and follow-up actions taken by the system. Additionally, the language transformer is pre-trained on a much larger dataset of full dialogue turns to learn the context of the utterances and their co-occurring actions. As the dialogue history is removed from the graph integrated language transformer training process, the model is incentivised to focus on action co-occurrence and sequences as graph nodes rather than the dialogue history surrounding them. Keeping only actions as the history of the dialogues (i.e., both in language transformer and graph components) removes dependency to the NLU pipeline (i.e., discussed in Section 1 and 2) and the need to keep the dialogue turns' utterances; thus improving speed of prediction and satisfying the language transformer token limit; e.g., 512 for DistilBERT (Sanh et al., 2019; Devlin et al., 2018). Due to the simplicity of the model, real time inference time requirements are still being met. Figure 3 shows a schematic of the proposed models.

A fusion layer combines both language transformer and graph component features using Equations 2-4. First, Equation 2 computes mean of the hidden features from the language transformer and Equation 3 computes the features of the graph component. Here, W and b are trainable parameters, O is the output of a layer, l and g denote the language transformer and graph component. Then, the fused features will be fed into a fully connected layer to predict the next action. Equation 4 fuses the hidden features of both layers and generates the probability using the $Softmax$ activation layer. The next action will be picked from the list of all actions with respect to their probability of the computed Softmax output. While there are variety of fusion techniques (e.g., concatenation, dot product techniques, or summation techniques), Equation 4 uses $\otimes$; since GaLT and GNN-LT reach to the highest performance via pairwise dot product fusuion.

$$H_l = GELU(W_l \ \ mean(O_l) + b_l) \quad (2)$$

$$H_g = GELU(W_g O_g + b_g) \quad (3)$$

$$H_f = Softmax(W_f(H_l \otimes H_g) + b_f) \quad (4)$$

4 Experimental Setup and Results

This section describes the process of collecting data for training the models, comparing the trained models regarding classification metrics (i.e., $F1$), and evaluating the proposed models as well as the DM system[3], explained in detail in Appendix A, using a human-centered approach.

4.1 Data, Configurations, and Training

To integrate the graph information into GNN-LT and GaLT models, this work utilizes conversational data which follows a Standard Operating Procedure (SOP). These conversations were guided by a human expert or the DM system which employs a human defined SOP. The SOP is a graph like structure with actions as nodes and their connections to next actions based on filled slots, which has been carefully translated into dialogue manager logic. Appendix B discuss the SOP in more details. However, the proposed Graph Integrated Language Transformers were not trained on the SOP explicitly. GaLT and GNN-LT were trained on the data human experts and the DM system collected and generated from the SOP.

To evaluate the proposed models, dialogue turns of phone calls between human-AI and human-human were collected from June to August 2023. The next action for each human dialogue turn was decided and labeled by the DM system with human in the loop supervision. Human domain experts intervened in calls that might fail. The intervention varied from correcting the collected data (e.g., spelling mistakes) to driving the calls in severe cases. To generate a reliable dataset, a team of human experts classified each conversation as successful or unsuccessful on a call level, rather than labeling and reviewing each dialogue turn, due to financial reasons and limited human resources. For the same reason, all of dialogue turns for each call are added to the dataset if it was considered successful [4] or was dropped otherwise. That resulted in $\sim 1M$ records each including one human utterance and one system response. In addition to selecting successful calls, a pre-processing step (described

[3]The current production system that is handling the call automation at the time is called DM system throughout this paper.

[4]If the model managed to prompt the human user to give all information required

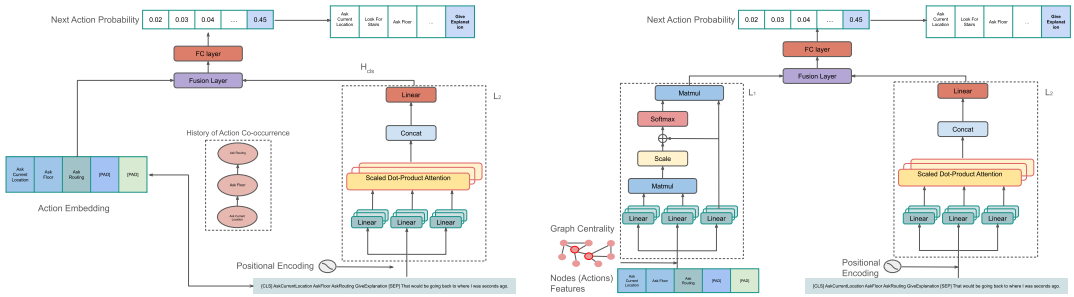

Figure 3: The architecture of the GaLT model (i.e., left figure) and GNN-LT (i.e., right figure). GaLT is fed action history as graph embedding and GNN-LT is fed actions as nodes as well as utterance features as edges; each models then is fused with a language transformer. $L1$ denotes the number of layers in GNN and L_2 denotes the layers of the language transformer.

in Appendix C), is devised to remove undesirable dialogue turns, calls, or actions; e.g., actions that are deprecated and the rest of the call to avoid incorrect connection between actions. This process led to $\sim 600K$ remaining dialogue turns.

Despite filtering out $\sim 400k$ dialogue turns, the language transformers were initially pre-trained on all dialogue turns (i.e., $\sim 1M$) using Masked Language Modeling (MLM) (Devlin et al., 2018) and then fine-tuned on $\sim 600K$ selected dataset for the *next action prediction* task. The dataset was randomly split to 80%-10%-10% for training, validation, and test. Section D summarizes the details of the dataset. Additionally, Section E and Section F lists the system configurations and proposed models' hyper-parameters for training and testing the models.

4.2 Classification Performance Comparison

This section evaluates the proposed models and other techniques using an offline classification evaluation. The process evaluates each technique's performance on the turn-level; next action given a human user's utterance and the previous actions or dialogue history. To measure the performance for each model, F1 Score was computed on the test-set described in Section 4.1.

Table 1 compares the proposed models with other techniques. The dataset, described in Appendix D, consists of 80 next actions (i.e., classes) of imbalanced frequency; thus $F1_{Macro}$ was calculated alongside $F1_{weighted}$. The results suggest that stand-alone models (i.e., language transformers or GNNs) and prompt-based large language models [5] are not able to predict the next action with high performance (i.e., lower $F1_{macro}$). Moreover, this table shows adding the graph embedding of actions in GaLT can improve $F1$ for *next action prediction* more than combining complex GNN models. GaLT also can reach to its high performance with as little as $60K$ dialogue turns(i.e., 10% data size) as described in Appendix H.

Model	$F1_{Weighted}$	$F1_{Macro}$
BERT w/ dialogue history (Mosig et al., 2020)	0.58	0.38
BERT w/ SF (Zhang et al., 2021)	0.79	0.44
BERT w/ action history	0.80	0.63
DistilBERT w/ action history	0.82	0.69
RoBERTa w/ action history	0.78	0.60
GNN (Yun et al., 2019)	0.72	0.52
(sub)*GNN (Yun et al., 2019)	0.72	0.51
GNN-LT(DistilBERT)	<u>0.84</u>	0.72
(sub)*GNN-LT(DistilBERT)	<u>0.84</u>	0.72
GaLT	<u>0.84</u>	<u>0.75</u>

*sub-GNN models are fed only recent actions (e.g., last 5 or 10).

Table 1: Summary of offline classification evaluation across different techniques regarding $F1$. Four categories of models were listed in this table; language transformers (e.g., BERT) with dialogue history or detected filled slots, language transformers with last utterance and recent history of actions (e.g., 5 or 10 last actions), GNN model, and Graph Integrated language transformers (e.g., GNN or graph embedding). The <u>underscore</u> values show the best performance regarding each metric (i.e., columns).

4.3 Human-Centered Evaluation

This section evaluates the best performing model, GaLT, with the DM system using a human-centered approach since the desired outcome of a call can be

[5] This paper also evaluates a prompting only approach using Llama2 (https://ai.meta.com/llama) on the same task and dataset; however, the results are not reported due to poor results in comparisons with other models. The prompt that is used to generate outputs as well as the results are discussed in Appendix G.

Difficulty Level	#Fields Mean (std) DM system	Proposed	#Panels Mean(std) DM system	Proposed
Easy	23.1(6.59)	25.35(6.19)	3.85(0.65)	4.0(0.0)
Medium	18.36(9.23)$^{\cdot}$	23.3(4.45)$^{\cdot}$	3.05(1.39)**	(4.0)0.0**
Hard	18.25(5.49)	21.44(5.98)	3.63(0.99)	3.66(0.94)
Total	20.36(7.97)*	23.79(5.70)*	3.48(1.13)*	3.93(0.42)*

Note: $^{\cdot}p<0.1$, $^{*}p<0.05$, $^{**}p<0.01$, $^{***}P<0.001$

Table 2: Comparing the DM system and proposed models performance regarding product-level metrics, number of fields and panels, across different difficulty levels. The results of t-test are shown as stars ('*') or dots ('.')

achieved through various paths and does not need to be strictly tied to one correct next action (i.e., what was done in Section 4.2). Put precisely, more than one next action can be considered as a correct prediction given the recent actions and the current utterance. To compare GaLT with the DM system, human assessors acted the "role" of the agent receiving outbound calls. They were familiar with the call structure and expected outcome of calls. Two different approaches were designed to compare and evaluate the models; objective product-level and human subjective. Additionally to test the generalizability and robustness of the compared models three call difficulty levels were defined; easy, medium, and hard (i.e., Table 7). As the call difficulty level increases human utterances and provided information get more complex (e.g., mumbling or updating a piece of information). The experimental setup and metrics are described in more detail in Appendix I.

Production Level Metrics Table 2 shows that the proposed models outperforms the DM system regarding both *field number* (i.e., how much information the call collected) and *panel number* [6] (i.e., how far to the end of the call model reached). T-test statistics analysis suggests that the comparisons were significant for *medium* level as well as all levels combined (i.e., '.' and '*' symbols for each pair in Table 2). In addition to the *panel number*, finishing a call successfully (e.g., collecting all information or without human user hanging up) is another important metric (i.e., *E2E* metric). GaLT also improved the *E2E* or number of successfully finished calls by +31.92% (Appendix J shows an extensive comparison).

[6]Panel number indicates the progress a model is made into finishing a call. Panel 0,1,2,3,4, and E2E indicate 0%, 20%, 40%, 60%, 80%, and 100% progress of a call respectively.

Subjective Human Evaluation Additionally, Human agents (i.e., human users who interacted with the models) and reviewers rated each call after finishing that call as described in Appendix I using a 5-point Likert scale rating. The GaLT model received a higher rating average of 2.91 ($std = 1.15$) in comparison to rating average of 2.78 ($std = 1.42$) for the DM system. Comparing the number of positive and negative ratings for each model shows that both models received almost same number of positive ratings but the DM system received higher number of negative ratings. In other words, human assessors rated the proposed models to be more robust. A deeper investigation regarding difficulty levels is done and discussed in Section K.

5 Conclusion

This paper proposes Graph Integrated Language Transformers technique to improve *next action prediction* performance to resolve the dependency on Slot-Filling and Intent-Classification techniques and grounding issue (Mannekote, 2023). The analyses indicate that keeping the action history with order of the actions using a graph embedding layer and combining with language transformers generates higher quality of outputs in comparison to more complex models that include connection details of actions (i.e, GNNs including the connection details through edges). The proposed model(s) improve the *next action prediction* regarding $F1$ as well as product-level metrics and human-centered evaluation. They can improve the robustness regarding *next action prediction* (e.g., less unexpected results or being stuck in a loop) in comparison to other techniques and handle complex tasks better in comparison to the DM system in long noisy phone calls. Additionally, the proposed models can reach to a high performance level with as low as $60K$ dialogue turns. We hope future research can employ a similar method combined with generative AI models to extract the information from human utterances as well as generating custom responses to automate calls without dependency on other components.

Limitations

Although the proposed models can reach to a high performance with as little as $60K$ dialogue turns, it needs re-training or fine-tuning for any new application in a new domain or even with slightest

changes; e.g., adding or removing even one action. Moreover, similar to other neural models, graph integrated language transformers, lack interpretability and may show instability (e.g., predict an action that does not have any relationship to dialogue history).

In addition to these limitations, the evaluation can benefit from further investigation. This paper recruits human agents and employees who were familiar with the DM system. That can lead to a biased assessment and perhaps is the source of inconsistency between human subjective rating and product-level metrics.

Finally, there are next steps to further evaluate graph injection with additional third party GenAI prompt based models. The ability to use certain third party systems was limited at the time of evaluation due to the requirement for this healthcare dataset to stay HIPAA compliant.

References

Duygu Altinok. 2018. An ontology-based dialogue management system for banking and finance dialogue systems. *arXiv preprint arXiv:1804.04838*.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3002–3017.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Traum David. 2017. Computational approaches to dialogue. In *The Routledge Handbook of Language and Dialogue*, pages 143–161. Routledge.

Harm de Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. 2018. Talk the walk: Navigating grids in new york city through grounded dialogue.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nouha Dziri, Andrea Madotto, Osmar R Zaiane, and Avishek Joey Bose. 2021. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2197–2214.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. *arXiv preprint arXiv:1704.07130*.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics.

Matthew S Henderson. 2015. *Discriminative methods for statistical spoken dialogue systems*. Ph.D. thesis, University of Cambridge.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. *arXiv preprint arXiv:2011.00564*.

Amogh Mannekote. 2023. Towards a neural era in dialogue management for collaboration: A literature survey. *arXiv preprint arXiv:2307.09021*.

Shikib Mehri and Maxine Eskenazi. 2021. Schema-guided paradigm for zero-shot dialog. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 499–508.

Johannes EM Mosig, Shikib Mehri, and Thomas Kober. 2020. Star: A schema-guided dialog dataset for transfer learning. *arXiv preprint arXiv:2010.11853*.

Tim Paek and Roberto Pieraccini. 2008. Automating spoken dialogue management design using machine learning: An industry perspective. *Speech communication*, 50(8-9):716–729.

Liang Qiu, Yizhou Zhao, Yuan Liang, Pan Lu, Weiyan Shi, Zhou Yu, and Song-Chun Zhu. 2022. Towards socially intelligent agents with mental state transition and human value. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 146–158.

Arushi Raghuvanshi, Lucien Carroll, and Karthik Raghunathan. 2018. Developing production-level conversational interfaces with shallow semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 157–162.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8689–8696.

S Reshmi and Kannan Balakrishnan. 2016. Implementation of an inquisitive chatbot for database supported knowledge bases. *sādhanā*, 41:1173–1178.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Charlie Snell, Sherry Yang, Justin Fu, Yi Su, and Sergey Levine. 2022. Context-aware language modeling for goal-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2351–2366, Seattle, United States. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Milene Santos Teixeira, Vinícius Maran, and Mauro Dragoni. 2021. The interplay of a conversational ontology and ai planning for health dialogue management. In *Proceedings of the 36th annual ACM symposium on applied computing*, pages 611–619.

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 673–683, Hong Kong, China. Association for Computational Linguistics.

Julio Vizcarra and Kristiina Jokinen. 2022. Knowledge-based dialogue system for the ageing support on daily activities. In *International Conference on Human-Computer Interaction*, pages 122–133. Springer.

Jack Weston, Raphael Lenain, Udeepa Meepegama, and Emil Fristed. 2022. Generative pretraining for paraphrase evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4052–4073, Dublin, Ireland. Association for Computational Linguistics.

Jason Weston, Emily Dinan, and Alexander H Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.

Sebastian Wollny, Jan Schneider, Daniele Di Mitri, Joshua Weidlich, Marc Rittberger, and Hendrik Drachsler. 2021. Are we there yet?-a systematic literature review on chatbots in education. *Frontiers in artificial intelligence*, 4:654924.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *Advances in neural information processing systems*, 32.

Jing Zhang and Yujin Wang. 2022. SRCB at SemEval-2022 task 5: Pretraining based image to text late sequential fusion system for multimodal misogynous meme identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 585–596, Seattle, United States. Association for Computational Linguistics.

Yang Zhang, Vahid Noroozi, Evelina Bakhturina, and Boris Ginsburg. 2021. Sgd-qa: Fast schema-guided dialogue state tracking for unseen services. *arXiv preprint arXiv:2105.08049*.

Pei Zhou, Andrew Zhu, Jennifer Hu, Jay Pujara, Xiang Ren, Chris Callison-Burch, Yejin Choi, and Prithviraj Ammanabrolu. 2023. I cast detect thoughts: Learning to converse and guide with intents and theory-of-mind in dungeons and dragons. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11136–11155.

A Conversational Systems' *Next Action Prediction* Process

Figure 4 shows a schematic overview of how a *next action prediction* model is employed in conversational AI systems. Although different conversational AI systems may use different approaches for NLU analysis or dialogue manager logic, most of the current approaches still use similar mechanism (e.g., Mannekote, 2023; Bocklisch et al., 2017; Raghuvanshi et al., 2018).

B Standard Operating Procedure

The DM system described in Appendix A employs a human defined Standard Operating Procedure

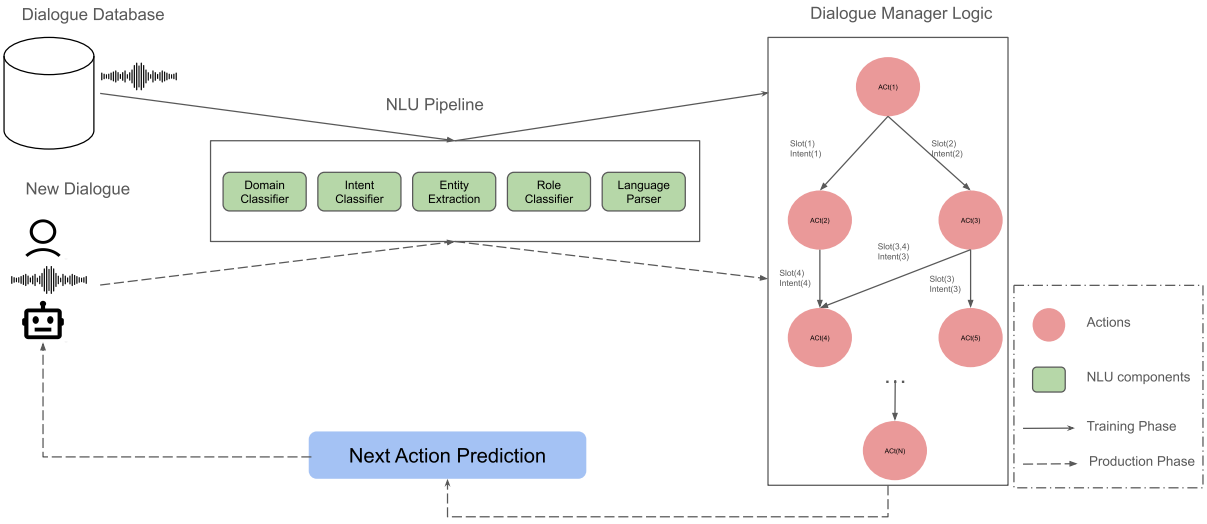

Figure 4: Overview of *next action prediction* process in a conversational AI system. The dialogue manager logic is generated using previous dialogues and via NLU pipeline (e.g., SF and IC). The model predicts next action using incoming utterances, NLU pipeline, and the generated knowledge. The arrows in the image show the connections between data, NLU pipeline, and dialogue manager logic during training (i.e., straight lines) or prediction during production (dotted lines).

(SOP) to guide the conversation based on the last action and conversational context of past slots filled. For example, one of the questions asked by the AI system is "Is this a commercial or government plan?" Depending on the type of plan different paths have to be followed. If it is a government plan, the AI system should ask "Is this Medicare, Medicaid, or Tricare?". If it is a commercial plan, the AI system should next ask about the Rx Number. If the Rx number is the same as a previously provided policy number, the AI system should push back to clarify "Just to confirm, the RX group number and the policy number are the same?". Similarly, throughout the conversation these types of guidelines are defined which are necessary for collecting accurate information in these healthcare calls. Depending on the information provided so far on the call, the SOP may require different confirmations and followup loops similar to the above example.

C Data Preprocessing

The data preprocessing resulted in $> 593K$ records each including one human utterance, the previous and next action as well as the system response. Through preprocessing, three types of records were removed from the dataset:

- records with rare or obsolete [7] next actions and the rest of the call: A low number of next actions, N 10, only appeared less than 50 times across the dataset due to different reasons (e.g., getting merged or updated). While the preprocessing kept the dialogue history up to that moment, the rest of dialogue was dropped since lack of prior information (i.e., deleted records) can be misleading for a *next action prediction* model.

- records with filler actions such as *wait, just a moment, or repeat last sentence* : The preprocessing also dropped these records and actions because, filler actions 1) do not add any meaningful instructions to the graph structure and 2) do not need dialogue history or previous actions to be detected.

  The preprocessing also dropped these for the same reasons stated for *waiting* actions above.

In addition to these steps, utterances split into fragments (i.e., multiple dialogue turns with one same next action) were merged to form one record

[7] No longer has been used in the DM system

Calls	21,220
Dialogue Turns	593,156
Average Turns per call	27.95
Average Tokens per Call	544.16
Average Tokens per Turn	19.47

Table 3: Summary of the dataset regarding number of calls, human utterances (i.e., dialogue turns), and tokens.

Panel	Progress	Actions	Dialogue Turns (%)
0*	0%	17	313214(53%)
1	20%	39	43,095(7%)
2	40%	4	135,52(2%)
3	60%	4	166,068(28%)
4	80%	20	57,227(10%)
Total	-	80	593,156(100%)

*Panel 0: Authentication; finishing a call at this panel means the call has failed.

Table 4: Summary of actions and dialogue turns per panels.

with one desired next action. Although, it is important to handle edge cases such as incomplete sentences for a conversational AI system in call automation, managing those are less relevant to the *next action prediction* models. Moreover, the proposed models handled incomplete sentences well during evaluation.

D Dataset Details

This section summarizes the details of the dataset regarding number of calls and dialogue turns in Table 3 as well as actions and panels in Table 4.

E System Configurations

The experiments in this paper including training and testing phases were done by two Computing Engines of the Google Cloud Platform; One including two "NVIDIA T4 16 GB Memory" GPUs and the other including a "NVIDIA A100 40 GB Memory" GPU. "T4" GPUs were used to train the MLM and GaLT models as well as other language transform approaches while the "A100" unit was used for GNN based approaches as they needed more memory.

F Models' Hyper-parameteres

Table 5 lists the parameters and their values for training the proposed model.

G Prompting Llama2

To evaluate the performance of Prompt Engineering on Large Language Models, Llama 2 was chosen as it was one of the few available ones at the time of experiments that had the proper HIPAA compliance requirements in place which is a requirement for this healthcare dataset to stay PHI compliant [8]. The prompt included basic instructions on the task and the dialogue history between user (i.e., agent) and system (i.e., user). The model was evaluated on the same dataset and achieved an $F1$ score of 0.09. Figure 5 shows a sample snippet of the prompt; it is customized for each request. The contextual information supplied in the prompt included basic instructions, the last actions of dialogue history (i.e., up to 10 turns), a list of next actions and their descriptions. To reduce prompt size and restrict the action search space, the prompt included only a subset of potential next actions. This list was determined by their observed co-occurrence in the dataset. All next actions were included if there were up to 10 co-occurring actions. For cases where there were more than 10, as many actions as required to add up to a cumulative sum of 50% were added to the set.

Parameter	Value (GaLT/MLM)
Epochs	3/30
Batch Size	256/512
Optimizer	AdamW/AdamW
Max. Learning Rate	5e-5/5e-5
Learning Rate Policy	linear/linear
Warmup steps	250/250
Max. Input Sequence Length	NA/128
Masking Probability	NA/15%

Table 5: List of hyper-parameters the proposed models was trained on.

Train Size (%)	$F1_{Weighted}$	$F1_{Macro}$
5,930 (1%)	0.52	0.29
11,860 (2%)	0.75	0.59
59,300 (10%)	0.82	0.69
296,500 (50%)	0.84	0.72
593,156 (100%)	0.84	0.75

Table 6: Effect of training size on the proposed model, GaLT, performance.

H Data Size Effect

Table 6 shows the effect of training size on the proposed models performance. It suggests having $60K$ training data is almost enough to train GaLT to perform close to its best.

[8] https://www.hhs.gov/answers/hipaa/what-is-phi/index.html

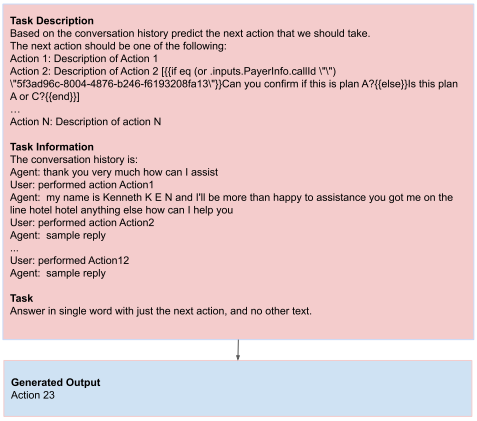

Figure 5: A sample snippet of the prompt input that is fed to the Llama2. Each Action has a name and a description that includes a Golang code on how the requirements of a next action is met. The prompt follows by the dialogue history and the task to generate response for.

I Human-centered Evaluation Setup

A call is considered successful if the conversational AI system was able to prompt the recipient of the call to provide all information fields required for completion of the task. Therefore, the **number of fields** gathered by the system is a direct measure of call success. The outbound call is structured into panels (i.e., **panel number**) which indicate how far into the conversation the system was able to navigate before call breakdown or completion. Therefore, both of these objective metrics indicate better performance the higher they are. This paper computes both of these metrics as objective metrics.

Additionally, after finishing a call, the human agent is asked to rate the call from 1-5 (i.e., 1 being extremely dissatisfied and 5 extremely satisfied) using a Likert scale. In addition to that, two additional human experts review both the objective and subjective assessment. Both the human agent and the reviewers answer an open-ended question of *how they describe their experience with the system* at the end of the process.

To make sure the evaluation is not handled in error-free, lab settings but more similar to real-world settings, different levels of difficulties were defined and considered for each call (e.g., background noise or repeated expressions). Three difficulty levels were defined (i.e., hard, medium, and easy) and each level consisted of a minimum-maximum number of scenarios challenges described in Table 7.

Scenario	Level		
	Easy	Medium	Hard
Agent	0-1	2-3	4-5
Flow	0-1	2-3	4-5

Table 7: Summary of how each difficulty level is made from agent and flow scenarios for calls. Human agents were assigned a difficulty level and could pick a number in the given range of the available scenarios to act out for their call. For each call difficulty level both agent and flow scenarios were picked from the same level.

Two types of scenarios were defined; agent and flow scenarios. The agent scenarios (i.e., 5 conditions) are challenges regarding human users' performance during a call such as mumbling, background noise or repeated expressions. The flow scenarios (i.e., 6 conditions) are specific conditions and edge cases which increase the complexity of the conversation and the information required to be collected to complete the task. For each level, both agent and flow scenarios are selected from the same difficulty level.

J Extended Results of Product-level Metrics

E2E metrics shows the proportion of calls a model can finish successfully. Put differently, reaching to panel 4 alone is not the desired goal but reaching to *E2E* is the main goal of the call automation task. A comparison between the "DM sytem" and the proposed models in Figure 6 shows that the proposed models are perfect (i.e., 100% *E2E* reach) in *easy* and *medium* difficulties. However, the DM system was only able to finish 90% of the easy and 70% of medium calls. The proposed models also managed to finish 78% of calls successfully with *hard* difficulty whereas the DM system were able to finish 62% of calls.

K Extended Results of Subjective Human Evaluation

Figure 7 shows the distribution of the ratings regarding each model. The "DM system" received negative ratings (i.e., strongly negative) twice as much as the proposed model.

Figure 8 shows the distribution of the models regarding the human ratings using a violin plot. The DM system performed better regarding human assessment across *easy* difficulty ($M_{Current} = 3.48 > M_{Proposed} = 3.2; STD_{Current} = 1.08 <$

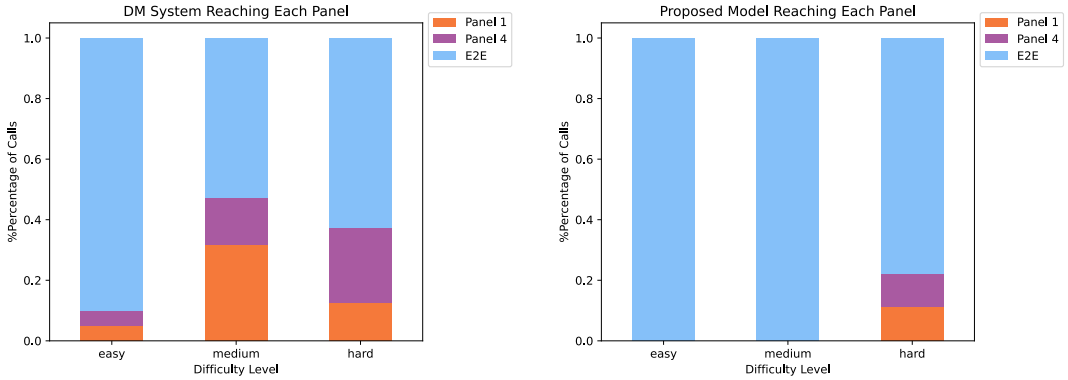

Figure 6: Percentage of calls reaching to each panel for the DM system (i.e., left figure) and the proposed models (i.e., right figure). There were 4 panels and end of a call (i.e., *E2E*) during a call and each model managed to finish only at panel 1, 4, or *E2E* due to lower number of actions in panel 2 and 3.

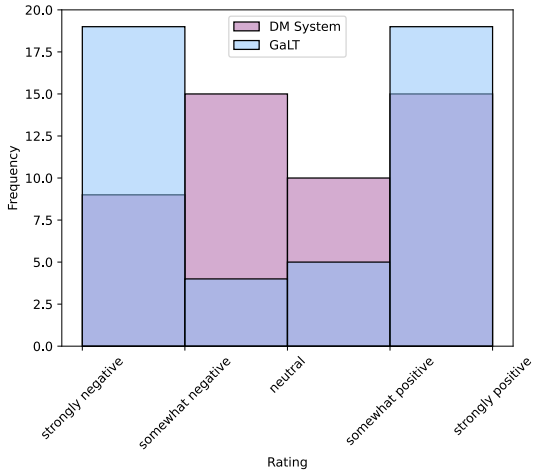

Figure 7: Distribution of human evaluation ratings for calls managed by the DM system and proposed model.

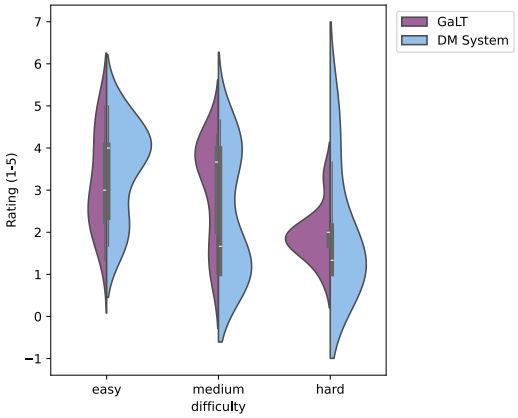

Figure 8: Violin plot of distribution for human evaluation rating regarding difficulty levels.

$STD_{proposed} = 1.11$). The DM system also performed slightly better for calls with *hard* difficulty on average but with a much larger standard deviation ($M_{Current} = 2.00 > M_{Proposed} = 1.93; STD_{Current} = 1.41 > STD_{proposed} = 0.58$). Higher standard variation for *hard* difficulty indicates that the proposed models will generate less unexpected actions or outputs. Moreover, the proposed models outperformed the DM system across *medium* difficulty ($M_{Current} = 2.37 < M_{Proposed} = 3.07; STD_{Current} = 1.41 > STD_{proposed} = 1.14$).

However, t-test statistics of human ratings across different difficulties as well as all levels combined were not significant ($p > 0.1$). These findings suggests, a careful consideration when interpreting these results and perhaps the need for a larger sample to compare ratings for both models.