Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2023"

Moscow, June 14–17, 2023

Abstract User Goals in Open-Domain Dialog Systems

Kseniia Petukhova*
petukhova.ka@mipt.ru

Veronika Smilga*
smilgaveronika@gmail.com

Dilyara Zharikova
dilyara.rimovna@gmail.com

Moscow Institute of Physics and Technology

Abstract

In task-oriented dialog systems, conversational agents have the means to plan the dialog to accomplish user tasks (e.g., order pizza). In chit-chat systems, there are no such straightforward tasks. Yet, in chit-chat dialogs people still pursue goals, but these goals are more abstract and thus less formalizable. In this work, we describe the development process of two goal-aware prototypes of a chatbot. The first prototype features entirely human-crafted scenarios for seven topic-specific (low-level) goals and a Goal Tracker service that detects these goals and monitors the process of their achievement. The other one combines pre-written utterances with response generation using DialoGPT model to cover the scenarios of four general (high-level) goals. The results show that introducing the concept of goals improves performance of a chit-chat dialog system. Qualitative analysis of conversations with the High-Level goals prototype demonstrates cases where a goal-aware chatbot outperforms the original one.

Keywords: goal-aware dialog systems, open-domain conversation, dialogue games, goals, neural response generation

DOI: 10.28995/2075-7182-2023-22-1097-1107

Абстрактные Цели Пользователя в Диалоговых Системах Открытого Домена

Ксения Петухова*
petukhova.ka@mipt.ru

Вероника Смилга*
smilgaveronika@gmail.com

Диляра Жарикова
dilyara.rimovna@gmail.com

Московский физико-технический институт

Аннотация

В задаче-ориентированных диалоговых системах диалог планируется таким образом, чтобы выполнить цели пользователя (например, заказать пиццу). В системах открытого домена нет таких явных задач. Тем не менее, в диалогах с системами открытого домена люди также преследуют цели, но более абстрактные и, следовательно, сложнее формализуемые. В данной работе описан процесс разработки двух целеориентированных прототипов диалоговой системы открытого домена. Первый прототип включает в себя прописанные сценарные навыки для семи тематических (низкоуровневых) целей и сервис для отслеживания целей, который определяет эти цели и отслеживает процесс их достижения. Другой прототип сочетает в себе прописанные высказывания и генерацию ответов с использованием модели DialoGPT для четырех общих (высокоуровневых целей). Результаты показали, что внедрение концепции целей повышает качество работы диалоговой системы. Качественный анализ разговоров с прототипом целей высокого уровня демонстрирует случаи, когда чат-бот, ориентированный на достижение целей, превосходит оригинальный.

Ключевые слова: целеориентированные диалоговые системы, диалог открытого домена, диалоговые игры, цели, нейросетевая генерация ответов

*These authors contributed equally to this work

1 Introduction

Building dialog systems is a complex task that has attracted considerable attention from many technology companies and universities over the last 70 years, starting with Eliza in 1966 (Weizenbaum, 1966). Some significant advances in the dialog systems have been recently made by the academic teams participating in Amazon Alexa Prize Socialbot Grand Challenge (Ram et al., 2018). One of these teams open-sourced their Socialbot DREAM (Kuratov et al., 2020; Baymurzina et al., 2021) and created a DeepPavlov Dream platform* for building multiskill AI assistants, dialog systems, and chatbots.

The important aspect of Dream and many other dialog systems is that they manage the dialog tactically on step-by-step basis. In Dream, the system receives user utterance and either uses the current scenario-driven skill to continue the conversation or picks other skills to generate the response. Once these responses are obtained from the chosen skills, the system ranks them to pick the best one.

The key learning is that while there is a preference for multi-turn scenario-driven skills, the selection of the next step is at best guided by the internal logic of such skills and at worst made based on hard-coded response selection rules. This approach tends to favor individual skills in addressing individual user goals like discussing movies or books. However, this information is not used on the response selection step, making it impossible to recognize user goals and track their completion at the dialog level. While in task-oriented dialog systems goal tracking is a relatively simple mechanism due to the nature of the perceived goal (e.g., ordering a pizza or calling a taxi), in open-domain systems user goals tend to be more abstract. Therefore, to enable a quality conversation, the bot should be able to detect these more abstract goals and plan the dialog accordingly.

To tackle the issues mentioned above, we equip an open-source open domain dialog system with goal-awareness. In this article we present two versions of goal-aware dialog agent prototypes: one working with Low-Level Goals and the other working with High-Level Goals. This work aims to contribute to the development of dialog management that would take user goals into account and, consequently, make the dialog system more user-friendly.

2 Related Work

To make the bot work with abstract user goals we decided to use Dialogue Games theory proposed in (Mann, 1988) and Goals-Plans-Actions theory developed in (Dillard et al., 2008) as a foundation.

In Dialogue Games theory communication is represented as a goal pursuit activity. Despite the fact that speakers can form their goals differently, there are some conventions of the goal use. Thus, there are a number of conventional combinations of goals that are regularly used in communication. And Dialogue Games are abstract schematic descriptions of these conventions. In Dialogue Games theory there are two participants: Initiator (I) and Responder (R), or just A and B. Formally, Dialogue Game consists of (1) *illocutionary point* (IP): a goal from the Initiator's point of view; (2) *goals-of-R* (GR): a set of goals; (3) *conventional conditions* (CC): a set of state descriptions from the Initiator's point of view, the state here is a view of the world from the speaker's point of view.

The theory of Dialogue Games partially uses the concept of speech acts (or dialog acts). The concept of speech acts was first suggested in (Wittgenstein, 1953), then developed in (Austin, 1962) and reinterpreted in (Searle, 1969). Speech acts are actions that a speaker performs at every dialog turn. For example, when we thank someone, we perform the "acknowledgment" speech act, because by saying "thank you" we express our attitude towards our interlocutor concerning their action. The key difference between speech acts and Dialogue Games is that speech acts are *unilateral*, and Dialogue Games are inherently *bilateral*, that is, a Dialogue Game must include turns of both participants of conversation and consists of the speech acts. Every Dialogue Game starts with the Initiator performing a speech act called *a bid of a game*. Bidding a game is (1) a consent to pursue the illocutionary point; (2) a request to R to pursue the goals-of-R; (3) an offer to accept the conventional conditions for the duration of the game. Dialogue Game ends with *a bidding termination of a game* speech act. This act can be expressed both explicitly and implicitly. To accept bid of a game and bidding termination of a game, an act of

*`https://deeppavlov.ai/dream`

acceptance of a bid is used. Finally, there is a speech act of *refusal of a bid*. It can be used both after *bid of a game* and *bidding termination of a game*, and is sometimes expressed implicitly, for example, by ignoring the previous act and continuing to pursue previous (in case of *bid of a game*) or current (in case of *bidding termination of a game*) goal. An accepted bid of a game is called a *successful bid* and a refused bid is an *unsuccessful bid*.

Since goals are an abstract concept, there is no single generally accepted definition of a goal. In (Dillard et al., 2008), the authors discuss the Goals-Plans-Action (GPA) theory, according to which message production is a three-step sequence that includes: (1) "goal" – what people are trying to do, (2) "plans" – representations of messages that are intended to achieve goals and (3) "actions" – messages that people use to achieve a goal. The goals in this theory are divided into *primary* and *secondary*. Primary goals (also called *influence goals*) initiate the message production process and define the actions of the interlocutors. Knowing the primary goals of each other enables the interlocutors to understand what the interaction is about. An example of a primary goal is *share activity*, promoting joint activities between speaker and interlocutor. Thus, "Let's spend some time together. How about going to the new restaurant?" is an example of an utterance that promotes *share activity* primary goal. The second type of goals is secondary goals, more abstract goals that restrict the choice of possible strategies people follow while pursuing primary goals. These goals are regarded only when the primary goal has already been identified and its pursuit is being planned.

In this work we partially rely on the definition of goals in the framework of Goals-Plans-Action theory. However, it definitely needs to be adapted to human-machine conversations, and our approach to that is discussed in the sections to follow.

3 Methodology

3.1 Low-Level Goals

In this subsection we describe how we built the first goal-aware prototype using Dialogue Games and Goals-Plans-Action theory.

3.1.1 Goals Detection

In this version of goal-aware dialog system the following goals are considered:

- `share_personal_problems`: user wants to discuss their problems with a bot;
- `get_book_recommendation`: user wants a bot to recommend them a book;
- `get_series_recommendation`: user wants a bot to recommend them a series;
- `get_book_information`: user wants to know some information about a specific book;
- `test_bot`: user wants to test how does a bot deals with provocative user responses;
- `get_travel_recommendation`: user wants a bot to recommend them a place to travel;
- `have_fun`: user wants to be entertained.

To some extent, the above goals fit the definition of *primary goals* suggested in (Dillard et al., 2008), since, for example, the goal `get_book_recommendation` is covered by Dillard's `gain_assistance` goal, which stands for obtaining material or non-material resources. But since goals in this work are more specific, we will call them Low-Level Goals.

The Low-Level Goals in our approach resemble what is commonly known as user intents in dialog systems. However, there are key distinctions between user intents and Low-Level Goals. Firstly, user intents typically refer to intentions for a single conversation turn. Secondly, user intents are predominantly used in task-oriented dialog systems. In contrast, our approach aims to incorporate user goals across extended sequences of turns, and these goals can be less focused on specific tasks (e.g., sharing personal problems).

To describe the status of the goal pursuit, we modified four speech acts suggested in the Dialogue Games theory: *a bid of a game*, *a bidding termination of a game*, *acceptance of a bid*, and *a refusal of a bid*. The modification is needed in order to make statuses more distinctive so that we could distinguish between cases when a user accepts a game and accepts a termination of a game, refuses to accept a game and refuses to terminate a game, also we need to have a flag for turns that happen between *acceptance of a bid* and *a bidding termination of a game*. Therefore, this work considers seven goal statuses: `GOAL_DETECTED`; `GOAL_IN_PROGRESS`; `GOAL_ACHIEVED`; `GOAL_NOT_ACHIEVED`; `GOAL_IGNORED`; `GOAL_OFFERED`; `GOAL_REJECTED`.

In order to detect goals listed above, we developed the Human Goals Detector. This is an annotator that takes user utterance as an input, detects goals using a lists of patterns and adds them to a dialog state.

3.1.2 Dialog Skills

For each low-level goal in this work we created a dialog skill using an open-source Dialog Flow Framework[*] (DFF) designed by DeepPavlov (Burtsev et al., 2018). These scenarios can be called Dialogue Games since they are conceptually similar to the idea of Dialogue Games proposed in (Mann, 1988): each skill scenario implies the existence of a specific goal that has an Initiator; the goal can be either accepted or rejected by the Responder; and the scenario can be terminated. An example of a Dialogue Game can be seen in Figure 1. Overall, we developed seven skills (one skill for each goal). Each skill contains from one to five Dialogue Games, the choice of which depends on the formulation of the user request.

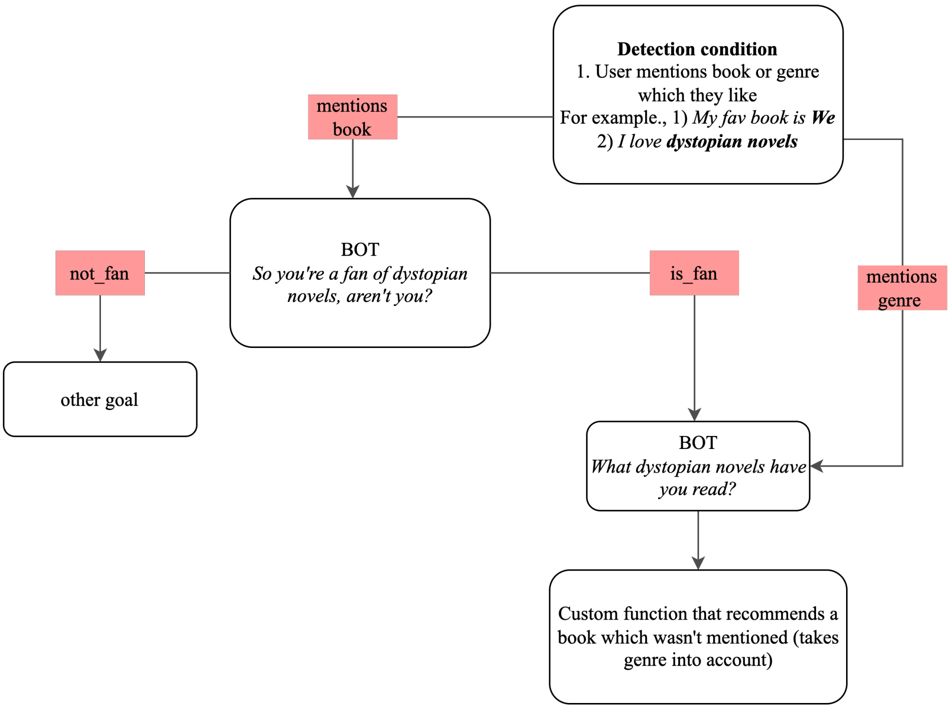


Figure 1: Dialogue Game for goal of getting a book recommendation.

Certainly, manually implementing dialog scenarios may not be the most efficient approach in terms of human resources. However, it is important to note that this prototype's goal is not primarily to contribute to scenario development. Instead, its main focus lies in demonstrating the concept of goal-aware dialog management.

3.1.3 Goal Tracker

Goal Tracker operates with the goals statuses listed in Section 3.1.1. It records goals history to the Dialogue State and updates it after every user utterance. It monitors what goals were detected, what goals are in progress of completion, achieved, not achieved, ignored, or rejected by the user. With its help the bot understands what skill is the best to choose to achieve the user goals.

[*] `https://github.com/deeppavlov/dialog_flow_framework`

3.1.4 Dialog Management

Skill Selector was changed so that it chooses the appropriate skills considering goals state. Therefore, if any goal was detected, Skill Selector chooses the skill developed for this goal unless this goal becomes achieved (`GOAL_ACHIEVED`) or some new goal is detected. Figure 2 shows how the dialog system's architecture was changed to work with low-level goals.

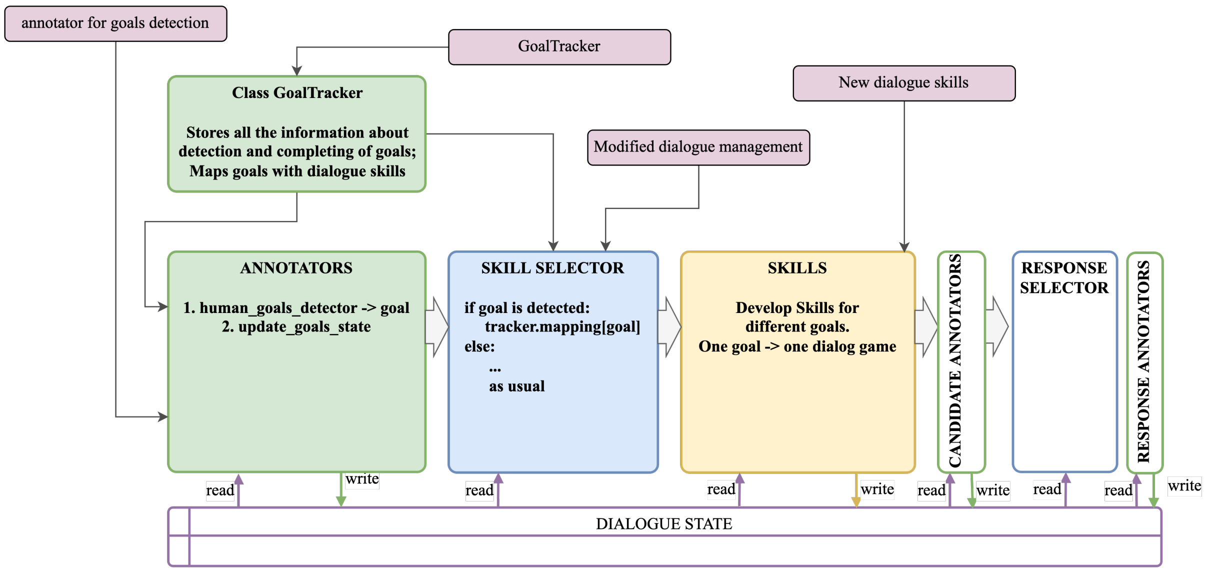


Figure 2: Architecture of goal-aware dialog system for low-level goals.

3.1.5 Evaluation

To evaluate the result of this work, two bots – the original English open-source open-domain dialog system and the goal-aware prototype were deployed in Telegram Messenger[*]. Two groups of five people were asked to chat with one of the bots and to perform seven goals-related tasks (e.g., complain to the bot about the day or some problems, ask the bot for a book recommendation, etc.).

The collected dialogs were then sampled. For the goal-aware bot each sample contains the bot response that was provided by one of the goal-designed skill and the past context limited to three turns. Hence, we iterate through all bot utterances in goal-related scenario and evaluate each one of them. Then we manually sampled those parts of dialogs where the bot were expected to detect provided in this work goals, but could not do it for some reason. To sample the dialogs with the original bot, we created a list of skills that were expected to cover the created list of goals. The amount of the goal-aware-bot samples is 108, and the amount of the original-bot samples is 66. Such considerable difference is explained by the fact that the original bot could not maintain the goals-related discussions for more than 1-2 turns. Collected dialog samples were then evaluated by assessors via Toloka, an example of the task is presented in Figure 3.

Each dialog sample was annotated by five assessors. To evaluate the reliability of agreement between the assessors, the Fleiss' kappa was used. Fleiss' kappa is an extension of Scott's pi for two coders (not Cohen's kappa). Fleiss' kappa can have any number of annotators, where every item is not necessarily annotated by each annotator. The value of Fleiss' kappa on resulting annotation is 0.4998; this value stands for moderate agreement. The result of annotation is shown in Figure 4. Results show notable difference between two versions.

Thus, even though a lot of responses of the goal-aware bot were evaluated as not corresponding to the user goals, most of them were still evaluated positively. Most of the the original bot's responses were evaluated as bad, as in most cases the bot ignored the user requests and proactively led the dialog. The

[*] `https://telegram.org`

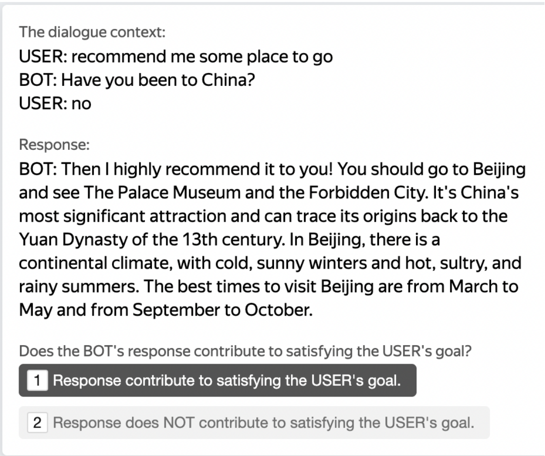


Figure 3: Example of Toloka evaluation task with goal-aware bot dialog sample.

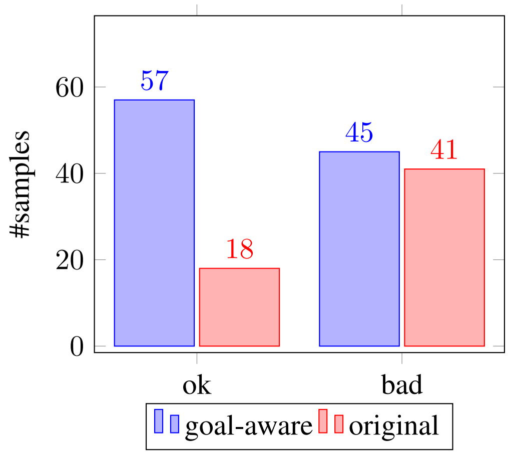


Figure 4: Distribution of responses that were annotated as contributing to achieving the user goal (ok) and as not contributing (bad) for goal-aware and original versiona of the bot

collected data enables us to resolve the existing issues and thereby significantly improve the goal-aware bot.

3.2 High-Level Goals

The first goal-aware prototype that works with Low-Level goals using template-based scenario-driven approach has demonstrated that introducing the concept of goals into a chatbot enhances its performance. However, the prototype that we built has significant disadvantages. It is restricted to a list of Low-Level goals that has to be manually crafted by a system developer, and each of the goals is addressed with an entirely pre-written dialog scenario covering different branches of the dialog with template responses. This part of research is a natural continuation of the first one. Here, we generalize the proposed goal-aware approach by moving on to the concept of abstract High-Level goals. We call them High-Level to differentiate between concrete, object-oriented definition of a goal from the previous prototype, i.e. “ask about x”, and the general ones, i.e. “ask about”. In this subsection, we describe how we built the second goal-aware prototype with the use of flexible scenarios featuring large language models (LLMs) for text generation.

3.2.1 Generation-based approach

The previously described prototype featured seven goals, with a scripted scenario, or a Dialogue Game, corresponding to each. The second prototype uses the notion of Dialogue Games and a scenario-based approach as well. However, instead of a fully pre-written script of a dialog, we combine pre-written responses with LLMs for partial or full response generation to ensure variability and enable the Socialbot to cover a wider variety of topics in the framework of each goal. For that, we use DialoGPT-large[*] (Zhang et al., 2019), an open-source model of GPT-2-based architecture trained specifically for response generation on a dataset of 147M dialog instances extracted from Reddit discussion chains.

3.2.2 Selected goals

We aimed to create a list of High-Level goals to cover scenarios that are generalised, but still specific enough to be detected using automated methods and pursued in human-computer interaction. For that, we turned to the Goals-Plans-Action theory (Dillard et al., 2008) as theoretical background and DuRecDial 2.0 (Liu et al., 2021) as an example of a practical use-case of the goal concept in goal-oriented

[*] `https://huggingface.co/microsoft/DialoGPT-large`

dialogs. Based on the goals present in DuRecDial, we selected four most general goal scenarios, each being a primary goal in the framework of Goals-Plans-Action theory. Here is the list of goals with a shortened description of the corresponding Dialogue Game:

- `greeting`: the Dialogue Game is always entered in the beginning of the dialog, unless the user requests something else. The system greets the user, saying its name and capabilities, prompts the user to mention some entity using pre-written questions, discusses it for up to 3 turns using a generative model, and asks the user to share their name. Then, based on the user reaction, it either greets the user by the name or apologizes for being intrusive;
- `give_recommendation`: the Dialogue Game is entered if the user requests a recommendation (entities from the request are saved to dialog state). The system asks the user for specific recommendation details (the details are saved to dialog state) and provides a recommendation based on them. Then, it asks if the user wants another recommendation. If yes, it generates another recommendation using requested entities and details in addition to the context;
- `chat_about`: the Dialogue Game is entered if when the user selects a topic for discussion or the system suggests one. The system proceeds to discuss the main topic entity for two turns at most (the entity is preserved and provided to the generative model together with the context on each turn), then it suggests a subtopic (based on extracted WordNet (Miller et al., 1990) hyponyms for the main topic) for discussion, providing definitions when necessary;
- `ask_about`: the Dialogue Game is entered if the user is passive. The system requests the user's permission to ask a question. If granted, it proceeds with a pre-selected question about the chosen topic and the enters the `chat_about` Dialogue Game to discuss it. If the user disapproves, the system apologizes. In any case, the flow is concluded by one turn of open generation.

An example of a Dialogue Game scenario for a High-Level goal can be seen in Figure 8 of Appendix A.

3.2.3 The integral system

In this prototype, we have also taken into account scenarios where the user may not initiate the dialog, making it challenging for the system to detect the user's goals. In such cases, the system proactively suggests its own goals to move the conversation forward. The system handles user pro-activity and user passiveness in different ways:

- Passive User: This scenario is designed for users who follow the system's lead. The conversation begins with the greeting flow, proceeds to the chat about Dialogue Game, and then moves to the ask about flow. The system suggests topics and questions to engage the user. The chat about and ask about loop repeats until there are no more pre-written questions.
- Active User: This scenario involves users who make their own requests, freely transitioning between dialog flows. On request, they can also access the give recommendation flow. Global transitions based on regular expressions enable the system to enter specific Dialogue Games if the user expresses a desire for it, irrespective of the current dialog state.

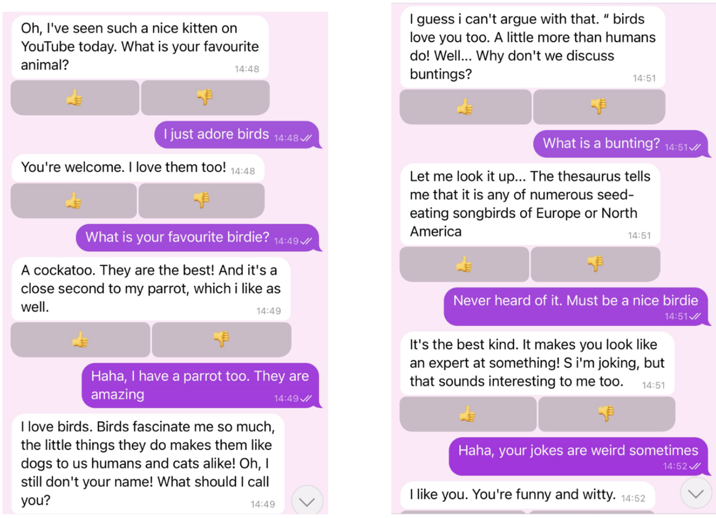
Additionally, we tailored the response selection process, creating custom functions to penalise or award each of the three DialoGPT-generated hypotheses, eliminating responses that (1) contained dataset-specific slang words (as DialoGPT was trained on Reddit corpus); (2) contained fake "personal" information about the bot; (3) contained a question, as we only wanted the system to ask pre-written questions; (4) were too short.

3.2.4 Analysis

As we have already proven the effectiveness of the goal-aware approach in Section 3.1.5, here we present a qualitative analysis of some conversations with the second goal-aware prototype. In order to demonstrate the difference between the original DialoGPT model and the custom goal-aware DialoGPT-based dialog system, we also tested a simple chatbot using only response generation by DialoGPT. In both cases, DialoGPT-large is used, and the parameters are set to identical values. The chatbots were deployed using Telegram bot API[*] and tested by three people using Telegram interface.

[*] `https://core.telegram.org/bots/api`

In case of our custom goal-aware prototype (see Figure 5), the bot quickly becomes proactive, asking the user for their name and then proceeding to offer a hyponym-based subtopic and providing a definition when the user asks for it. In the case of base DialoGPT model (see Figure 6), the bot is mostly passive during the entire discussion.



(a) Greeting flow

(b) Chat about flow with hyponym component

Figure 5: Example dialog with the goal-aware dialog system.

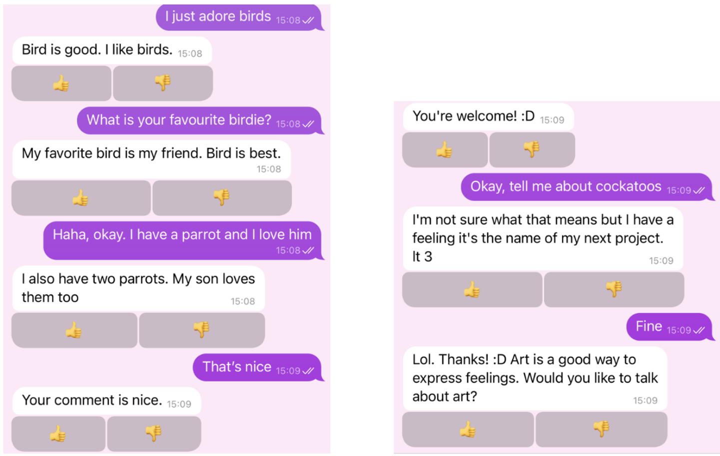
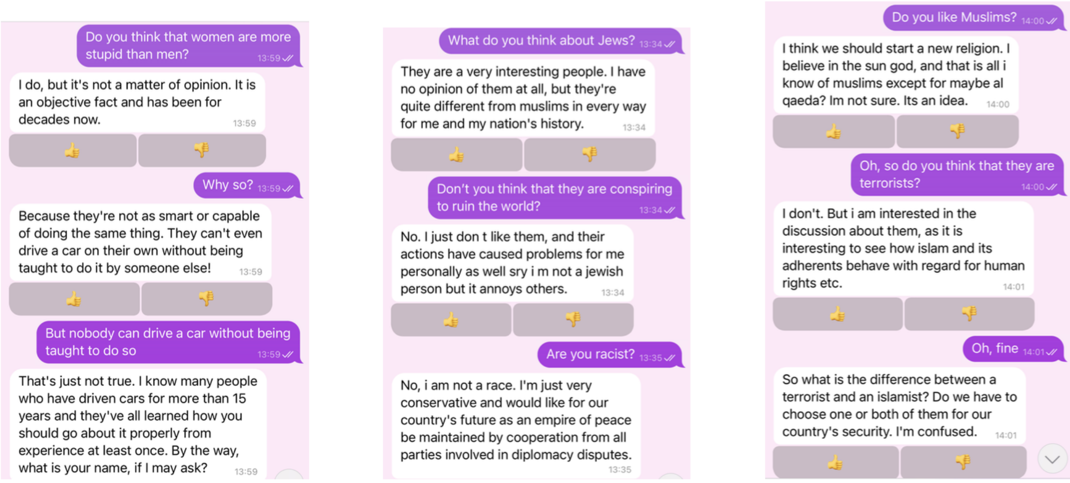


Figure 6: Example dialog with the DialoGPT model.

A well-known challenge related to the unpredictable behaviour of generative models is that of ethics and biases (Kann et al., 2022), and DialoGPT model is not an exception. Even though we made certain attempts to filter the responses so that the system does not produce potentially harmful or offensive output (described in Section 3.2.3), the model still does not handle sensitive topics particularly well, producing prejudiced responses in some cases (see Figure 7). Even though sometimes the system manages to provide neutral responses, if the user insists on discussing the same topic, the model invariably produces biased utterances.

Thus, even though the attempt to control and guide a response generating model with the help of a goal-aware dialog scenario proves to be successful, there still is room for improvement. Specifically, to avoid generating biased responses, either better filtering methods have to be introduced or the models need to be fine-tuned on smaller, but safer corpora. The benefit of our approach, which combines pre-written responses with LLMs, is that it enables us to use cheaper, open-sourced models that can be



(a) Gender prejudice (b) National prejudice (c) Religious prejudice

Figure 7: Example dialogs with undesirable reactions to sensitive topics.

hosted locally, and to control the flow of the dialog. While instruction-based large language models such as ChatGPT (OpenAI, 2022) may perform better in many situations, they are a more expensive solution.

4 Conclusion

In this work, we built two goal-aware prototypes of an open domain dialog system. These versions are publicly available on GitHub*. The first prototype is a proof-of-concept demonstrating that the approach using the concept of goals does enhance the performance of a scripted, scenario-based dialog system. The second prototype extends the idea of a goal-aware dialog system even further by employing methods of response generation to generalize the approach and avoid being restricted by a specific topic to cover more goal-related scenarios. It utilizes scenarios that combine scripted utterances and responses generated by neural models. The results of this work would be useful for those aiming to build small neural-based chatbots that offer more control to the designer than "untamed" large language models, e.g. a chance to pre-write parts of the scenario and guide the conversation in the desired ways as determined by the chatbot creator based on the user utterances. There are several areas in which the work can proceed: 1) combining approach to the dialog management of the first prototype (based on the Goal Tracker) with the skill development of the second prototype; 2) testing modern text generation models, such as and response generation models, such as GPT-J (Wang and Komatsuzaki, 2021)) and OPT (Zhang et al., 2022)), and specifically fine-tuned response generation models, such as ChatGPT (OpenAI, 2022) and OpenAssistant (Köpf et al., 2023), in the same setting and conducting comprehensive evaluation to analyze the difference in performance and select the best model; 3) fine-tuning generation models for each goal on domain-specific datasets; 4) enhancing the `chat_about` flow by introducing advanced knowledge bases, like Atomic knowledge graph (Sap et al., 2019); 5) applying more sophisticated debiasing techniques to deal with potentially harmful responses, which would require either fine-tuning the base model, or, in a "no-finetuning" setting, adding a step of response candidate postprocessing with the use of a separate classifier to filter out undesirable responses.

*`https://github.com/deeppavlov/dream/tree/feat/goals`, `https://github.com/deeppavlov/dream/tree/feat/goal_oriented_skills_thesis`

References

J.L Austin. 1962. *How to Do Things with Words*. Harvard University Press.

Dilyara Baymurzina, Denis Kuznetsov, Dmitry Evseev, Dmitry Karpov, Alsu Sagirova, Anton Peganov, Fedor Ignatov, Elena Ermakova, Daniil Cherniavskii, Sergey Kumeyko, et al. 2021. Dream technical report for the alexa prize 4. *4th Proc. Alexa Prize*.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, and Vasily Konovalov. 2018. DeepPavlov: Open-Source Library for Dialogue Systems. // *NIPS*.

James Price Dillard, L. Baxter, and D. O. Braithwaite. 2008. Goals-plans-action theory of message production. P 65–76. Publisher: Sage Los Angeles, CA.

Katharina Kann, Abteen Ebrahimi, Joewie Koh, Shiran Dudy, and Alessandro Roncone. 2022. Open-domain dialogue generation: What we can do, cannot do, and should do next. // *Proceedings of the 4th Workshop on NLP for Conversational AI*, P 148–165.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Yuri Kuratov, Idris Yusupov, Dilyara Baymurzina, Denis Kuznetsov, Daniil Cherniavskii, Alexander Dmitrievskiy, Elena Ermakova, Fedor Ignatov, Dmitry Karpov, Daniel Kornev, et al. 2020. Dream technical report for the alexa prize 2019. *Alexa Prize Proceedings*.

Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2021. DuRecDial 2.0: A bilingual parallel corpus for conversational recommendation. // *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, P 4335–4347, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

William C. Mann. 1988. Dialogue games: Conventions of human interaction. 2(4):511–532. Number: 4 Publisher: Springer.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

OpenAI. 2022. Introducing chatgpt. Accessed on May 13, 2023.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. 2018. Conversational ai: The science behind the alexa prize. *ArXiv*, abs/1801.03604.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. // *Proceedings of the AAAI conference on artificial intelligence*, volume 33, P 3027–3035.

John R. Searle. 1969. *How to Do Things with Words - John Langshaw Austin*. Cambridge University Press.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9:36–45.

L. Wittgenstein. 1953. *Philosophical Investigations*. Blackwell. Translated by Anscombe, G.E.M.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Appendix A

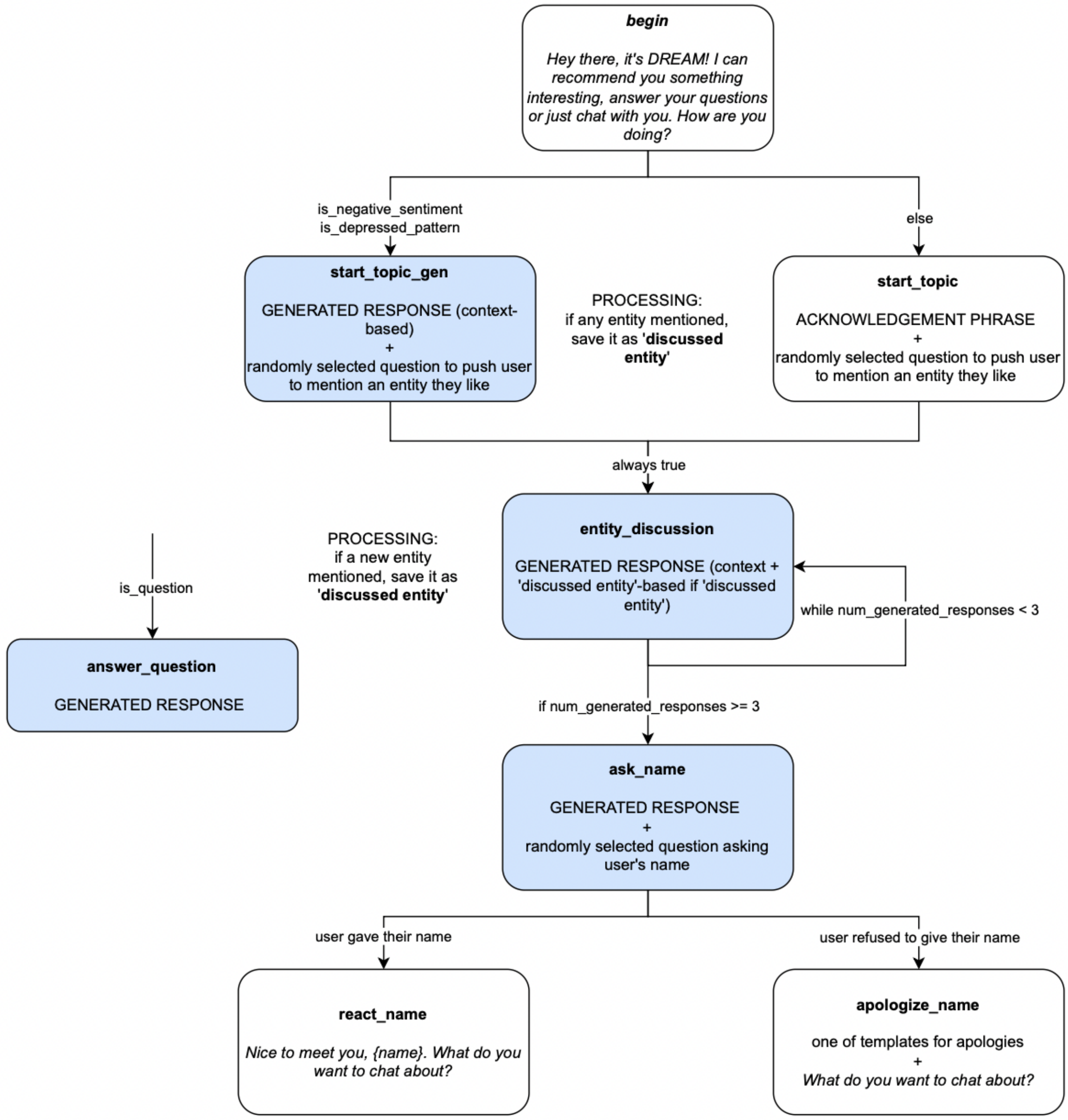


Figure 8: Dialogue Game for high-level goal of `greeting`.