Pre-Generative Conversational AI *or* How I Learned to Stop Worrying and Use LLMs

Staffan Larsson
Talkamatic AB and Gothenburg University
Gothenburg, Sweden
staffan@talkamatic.se, staffan.larsson@ling.gu.se

Abstract

Large Language Models (LLMs) such as ChatGPT have raised expectations on Conversational AI (CAI) applications, yet deployment is often hindered by controllability problems. This paper, to be accompanied by a demo, describes Pre-Generative Conversational AI (PGCAI) and its implementation in Talkamatic Dialog Studio, a tool suite for creating high-quality controllable conversational AI application without the need for coding, prompting or manual dialogue building. A key component is a controllable yet flexible and highly versatile dialogue manager, the Talkamatic Dialogue Manager (TDM). LLMs are used to generate dialogues which can be humanly curated to maximize quality. LLMs can also be used for NLU and as a fallback at runtime, but this is optional and not necessary.

1 Introduction

Generative AI in the form of Large Language Models such as ChatGPT is currently re-shaping the conversational AI landscape, and is generally taken to enable a multitude of practical Conversational AI applications in many different areas, including customer service, education, and more.

However, many companies and organizations are also hesitant when it comes to using an LLM-driven conversational agent to (for example) represent them on their website, or engage in one-to-one educational dialogue with children in schools. One reason for this is a host of well-known problems deriving from the overall problem of controlling the behaviour of LLMs as conversational partners. This may result in generating outputs that do not adhere to the desired agent behaviour (Kann et al., 2022). For example, LLMs have been known to "hallucinate" and produce false, harmful or otherwise unsuitable output. For question-answering conversational agents on websites, this can give a bad image of the company or organisation in question, which is typically something that one wants to avoid. Similarly, for educational applications it is crucial that the output of a conversational AI partner is not only factually correct and free from harmful or unsuitable elements, but also that it is pedagogically efficient and well designed to facilitate learning (Williams et al., 2023).

For many applications of LLMs, such problems can be handled by manually checking the output of the LLM before using it (e.g. publishing a text or sending an email). However, in conversational AI applications, this is typically not an option, as the LLM interacts directly in real time with users.

"Pre-Generative Conversational AI" aims to address precisely this problem. Instead of letting the user talk directly to a generative AI, with the risks that entails, we instead use generative AI to generate dialogues *before* they are published. At runtime, the dialogue can be handled without using LLMs at all, or using them only for limited tasks such as NLU. This makes it possible to control and design the dialogue so that it works exactly as desired. This also makes it possible to reduce the risk of hallucinations etc. to zero.

In essence, PGCAI enables using our normal preferred way of working with LLMs (generate-curate-publish) also for conversational AI.

2 Key components

PGCAI has to three key components: a dialogue generator, a dialogue editing tool, and a flexible but controllable dialogue manager. There are also two important additional components: the LLM control panel and the dialogue analytics tool.

2.1 Dialogue generator

The dialogue generator uses LLMs to create dialogues based on some content. Of course, there are many types of dialogue one could have about some content. Hence, the dialogue generator relies on distinguishing different dialogue types, or *genres* (Larsson, 2002; Ginzburg and Wong, 2024).

For example, given a text describing how to cook a pasta dish, one could have an educational dialogue where the Conversational AI partner assumes the user has read the recipe and checks that they have understood it correctly by asking a variety of questions, offering hints and encouragement if the user gives the wrong answer.

Another obvious type of dialogue to have about a recipe is an instructional dialogue where the CAI partner gives step-wise instructions for how to cook the pasta dish, answering any questions from the user that might come up.

A third type of dialogue (which could also be included in the previously mentioned types) could be a question-answering dialogue where the CAI can answer any questions about the recipe.

Given a database of recipes, a fourth type of dialogue could have the goal of helping the user decide which recipe to cook, given what ingredients are available at home or in a nearby shop. Such a dialogue could be limited to searching the recipe database but could also be extended to enable user and CAI to collaboratively explore and compare options before making a decision.

We refer to these types (or genres) of dialogue respectively as educational, instructional, question-answering and negotiative dialogue.

The task of the dialogue generator, then, is to take some content (a text, a database or something else) a specification of a dialogue genre, and produce a dialogue blueprint which can then be used by the dialogue manager to engage in a flexible dialogue. For each type of dialogue, it uses genre-specific prompts to produce dialogues of the type selected by the dialogue designer.

2.2 Dialogue curation tool

Since PGCAI does not require designing or implementing a dialogue (in the form of code or using a GUI), nor requires any prompt writing, we do not use the term "dialogue designer". Instead, the role of the human in building a dialogue application is to *curate* the dialogue, in the sense of taking an existing dialogue blueprint and adapting and perfecting it for the precise use it will be put to. To aid in this process, a dialogue curation tool is needed.

After a dialogue has been generated, it can immediately be tested by interacting with it. If the curator is unhappy with some aspect of the dialogue, they can go in and inspect and edit the dialogue blueprint. The precise structure of this blueprint will depend on the dialogue genre. For question-answering dialogue, the main component is a list of question-answer pairs. For education dialogue, it is a pedagogical interaction consisting mainly of questions of various kinds (right/wrong questions asking about information offered explicitly in the text, or requiring some inference on the part of the user, more open questions asking the user to reflect, and more). Other elements are also present, such as a list of potentially difficult words that the system can explain if needed. For other types of dialogue, other structures are available for curation. Importantly, these structures are quite simple and editing them does not require any deep technical understanding of conversational AI or even of human dialogue. However, genre-specific competence can often be useful, such as pedagogical skills in the case of educational dialogue.

2.3 Flexible dialogue management

LLMs are widely recognized as going considerably beyond the state of the art when it comes to natural language understanding (NLU). For this reason, we allow for using LLMs to take care of NLU even when not using them to generate responses to the user. Talkamatic Dialog Studio allows the dialogue designer to decide what NLU to use, offering LLMs as options but also non-LLM technologies including keyword-based NLU.

LLMs are also quite adept at handling many different kinds of dialogue in a flexible way, meaning that they often respond appropriately to less expected or less routine user behaviours. The success of PGCAI depends crucially on the ability of the system to achieve dialogue behaviour on par with or surpassing an LLM. Hence, we need to achieve a high level of flexibility in PGCAI, despite the fact that the dialogue blueprints are not generated at runtime. This poses considerable challenges for the dialogue manager. Talkamatic have developed the Talkamatic Dialogue Manager (Larsson and Berman, 2016) which supports a wide (and growing) variety of conversational behaviours across several dialogue genres, including the ones mentioned above.

2.4 LLM control panel

As mentioned above, it is possible to run dialogues without any LLM involvement, or using and LLM for NLU only. However, in some cases it may useful to access specific LLM features as a complement or fallback. For example, as part of educational dialogue, the CAI will sometimes ask users to provide a personal reflection on some topic, for example "What would you do in a Redwood forest?". Responses to such questions can be very diverse, but at the same time it is useful and encouraging if the system can react to the user answer in a way which shows it heard and understood, rather than just say "Thanks for sharing" or something similarly generic. In such cases, an LLM can be used in two different ways.

Firstly, the dialogue generator can provide a list of more detailed responses which are less generic but also not extremely specific, for example "That sounds very exciting", "Are you sure? That sounds a bit too wild for me!", "That seems like a calm and nice way to spend a day in the Redwood forest" and similar, but also including generic comments. At runtime, the LLM can then be asked to select from this list the most appropriate comment to make in reaction to the user's answer, also taking into account the dialogue so far. The output from the system will be adapted to the user's input, but with no risk of hallucinations or other errors.

Secondly, as a further option the LLM can be asked to generate a response to the user's answer, such as "Yes, driving a car through a Redwood tree can definitely be an interesting experience", "I agree, camping in the Redwood forest would probably be very cosy", etc.. While this will make the system seem even more understanding and clever, it of course again introduces the risk of hallucinations and bad output.

Still, both options are available to the dialogue curator, together with a host of other settings in an "LLM control panel". A simplified view of these settings is available in the form of an "LLM control slider" which decides an overall level of LLM involvement.

2.5 Dialogue analytics tool

Any conversational AI solution needs an analytics tool, to enable developers (or in the case of Talkamatic Studio, dialogue curators) to monitor how well the system is doing in terms of task success, user satisfaction, response time and so on. However, there are also certain analytics tools that are particularly useful for PGCAI. A dialogue curator may want to test a system with "friendly" users before launching it to the public. When doing this, the curator may allow the system to use LLMs to generate output to a relatively high degree. Taking the example of the question about what the user would do in a Redwood forest again, it may turn out that a large part of the user answers fall within a limited range of categories or variants. For each such category, one of the LLM-generated responses may be selected as a pre-stored response for answers in this category. When sufficient coverage has been achieved, the curator may switch off LLM generation and instead let the LLM map the user response to a category with a pre-stored response.

Of course, the usefulness of such "fine-tuning" of dialogues will be different depending on many factors, including how often the dialogue can be expected to be used.

3 Talkamatic Studio

Talkamatic Studio[1] is a comprehensive software service offering all the components needed for PGCAI. It offers a dialogue generator, a dialogue curation tool, a runtime frontend and backend using TDM for dialogue management, an LLM control panel, and a dialogue analytics tool. Screenshots of Talkamatic Studio are shown in the Appendix, Figures 1, 2, 3 and 4.

An overview comparing Talkamatic Studio to purely LLM-based conversational AI (such as ChatGPT) as well as the earlier generation of conversational AI prior to LLMs, such as Google DialogFlow[2] is shown in Table 1.

Regarding coverage of dialogue genres, TDM already handles all genres of dialogue mentioned above. At the time of writing (October 2024), dialogue generation and curation is available for question-answering and educational dialogue in the Freemium/Premium offer. Full-coverage versions of dialogue generation, dialogue curation, as well as a dialogue analytics tool will be rolled out in the near future.

4 Related work

4.1 "Guardrails" solutions

Of course, the control problem for LLMs is not new and a lot of work is being done to address it. The absolute majority of methods for dealing with this problem is of the "guardrails" type, referring to "set of predefined rules, limitations, and operational protocols that serve to govern the

[1] https://talkamatic.se

[2] It should be noted that DialogFlow recently also has been extended to incorporate elements of LLMs. Here we refer to the version of DialogFlow prior to these developments.

	Dialogue generated by LLM	Control and curation	Flexible dialogue across genres
CAI before LLMs (VoiceXML, Google DialogFlow etc.)	-	+	-
LLM-based CAI (ChatGPT etc.)	+	-	+
Talkamatic Studio	+	+	+

Table 1: Comparison of conversational AI solutions

behaviour and outputs of these advanced AI systems."[3]. In LLM-based Conversational AI, however, the user is still interacting with an LLM at runtime, and it is difficult or impossible to guarantee that guardrails always work. Ayyamperumal and Ge (2024) discuss various guardrail approaches such as layered protection models, system prompts, Retrieval-Augmented Generation (RAG) architectures and bias mitigation, and observe that "[c]rucial challenges remain in implementing these guardrails." Xu et al. (2024) show that hallucination is not just a temporary glitch, but is in fact inevitable in LLMs.

We believe that in many applications, including using Conversational AI agents for education in schools and to represent companies and organisations, there will be a strong preference for zero risk solutions, i.e. solutions that can *guarantee* there will be no bad output from the system.

4.2 Other controllable Conversational AI solutions

Before LLMs, the vast majority all commercial conversational AI solutions relied on controllable rule-based dialogue management. Prominent examples include Apple's Siri and Google DialogFlow, where dialogue designers could build dialogues in a low-code environment. Applications were of varying quality, depending to a large extent on the skills of the dialogue designer. With respect to dialogue management, the focus was on form-filling dialogue where the system collects a number of parameter values before searching a database or executing a requested action. The flexibility of dialogues was limited (Larsson, 2015, 2017).

There are a couple of other solutions on the market that combine the use of LLMs with fully controllable conversational AI. Rasa (Bocklisch et al., 2024) uses LLMs for NLU, but dialogue management is handled by a rule-based dialogue manager using Rasa's "business logic", not dissimilar to the dialogue plans used in TDM (Larsson and Berman, 2016). Dialogues are manually built by a dialogue designer using the Rasa Studio tool. Dialogue management is fairly versatile but appears to focus on form-filling dialogue.

OpenStream AI (Cohen and Galescu, 2023) employ symbolic planning and plan-recognition algorithms, based on work dating back to the 1980's, to provide explainable and controllable dialogue. In principle, this approach allows for flexible dialogue across several dialogue genres, but we have not been able to find information about how this translates to practical capabilities in applications built using the OpenStream platform. We have also not been able to find information about how dialogues are authored or designed, but we see no indication that LLMs are being employed.

We see it as a positive sign and a validation of our approach that other commercial actors are working along similar lines as pursued in Talkamatic Studio, specifically concerning the combination of LLMs and controllable dialogue management.

5 Conclusion

We have presented Pre-Generative Conversational AI and its implementation in Talkamatic Studio. This approach and implementation addresses a central problem with using LLMs for Conversational AI — the lack of control. To the best of our knowledge, Talkamatic Studio is the only solution that combines dialogues generated by LLMs, control (including complete control with no LLM output generation at runtime), curation (putting a human in the loop), and flexible dialogue across several dialogue genres, going beyond form-filling dialogue.

Future work includes comparative evaluation (along the lines of Bocklisch et al. (2024)) and extending the scope of PGCAI to more dialogue genres, thereby enabling its application in more

[3] https://attri.ai/blog/a-comprehensive-guide-everything-you-need-to-know-about-llms-guardrails

domains.

References

Suriya Ganesh Ayyamperumal and Limin Ge. 2024. Current state of llm risks and ai guardrails. *arXiv preprint arXiv:2406.12934*.

Tom Bocklisch, Thomas Werkmeister, Daksh Varshneya, and Alan Nichol. 2024. Task-oriented dialogue with in-context learning. *arXiv preprint arXiv:2402.12234*.

Philip R. Cohen and Lucian Galescu. 2023. An explainable collaborative dialogue system using a theory of mind.

Jonathan Ginzburg and Kwong-Cheong Wong. 2024. Language games and their types. *Linguistics and Philosophy*, 47(1):149–189.

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. In *Proceedings of the 4th Workshop on NLP for Conversational AI*. Association for Computational Linguistics.

Staffan Larsson. 2002. *Issue-Based Dialogue Management*. Ph.D. thesis, Department of Linguistics, Goteborg University.

Staffan Larsson. 2015. The state of the art in dealing with user answers. In *Proceedings of the 19th Workshop on the Semantics and Pragmatics of Dialogue (goDIAL)*.

Staffan Larsson. 2017. User-initiated sub-dialogues in state-of-the-art dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 17–22.

Staffan Larsson and Alexander Berman. 2016. Domain-specific and general syntax and semantics in the talkamatic dialogue manager. *Empirical Issues in Syntax and Semantics*, 11:91–110.

Tom Williams, Cynthia Matuszek, Kristiina Jokinen, Raj Korpan, James Pustejovsky, and Brian Scassellati. 2023. Voice in the machine: Ethical considerations for language-capable robots. *Communications of the ACM*, 66(8):20–23.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

A Talkamatic Studio workflow and screenshots

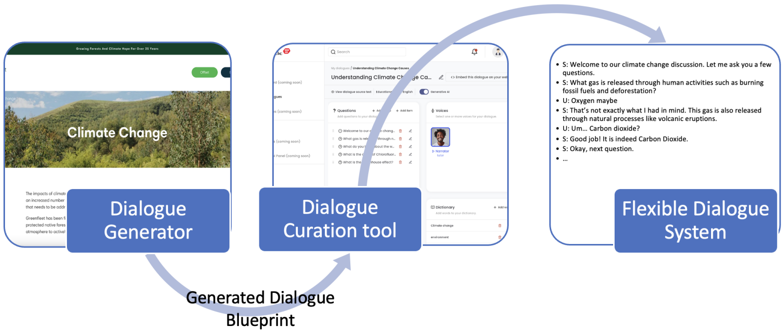

Figure 1: Schematic overview of workflow

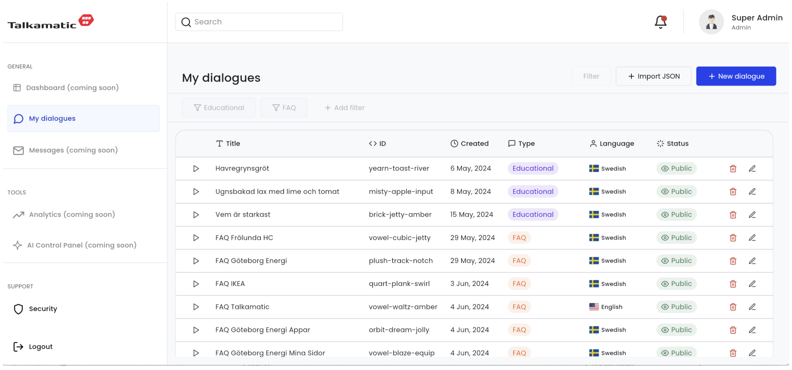

Figure 2: "My dialogues" view, showing a list of Educational and FAQ (question-answering) dialogues.

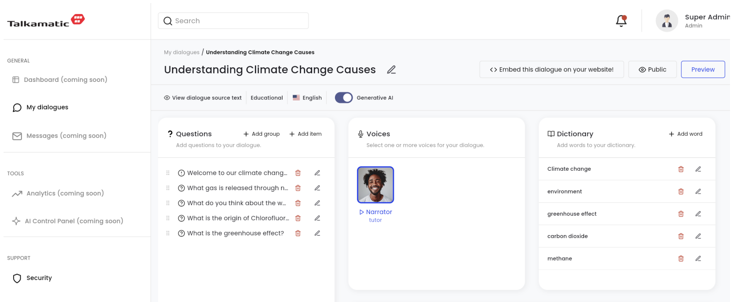

Figure 3: "Dialogue curation" view, where the curator can edit dialogues generated by an LLM.

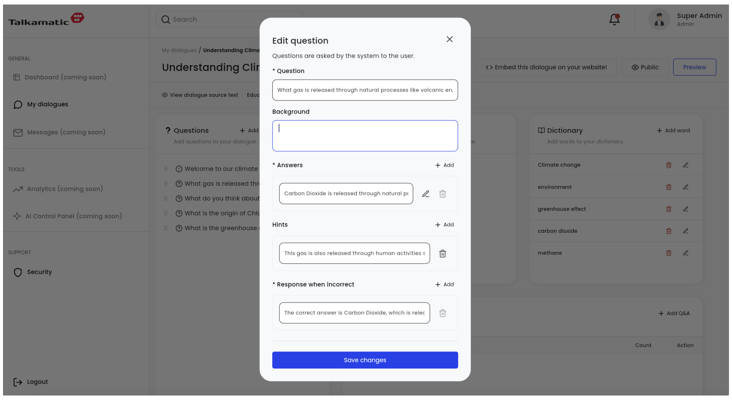

Figure 4: Editing an educational dialogue question.