A Goal-Oriented Chat-Like System for Evaluation of Large Language Models

Guilherme S. Teodoro Junior[1], **Sarajane M. Peres**[1], **Marcelo Fantinato**[1], **Anarosa A. F. Brandão**[1], **Fabio G. Cozman**[1]

[1]Universidade de São Paulo, Brazil

{teodoro538,sarajane,m.fantinato,anarosa.brandao,fgcozman}@usp.br

Abstract. *Large language models have changed the way various applications are developed. Interactions with large language models have reached a new level of complexity and now act as real problem solvers. However, despite their apparent competence, it is still necessary to accredit them with respect to the tasks they are assigned. In this paper, we discuss a systemic approach to accredit large language models through their integration with a goal-oriented chat-like system. An experiment involving prompt engineering for two models from the GPT family illustrates our evaluation scheme when applied to a real-world chatbot use case; our evaluation scheme reveals, that the resulting chatbots perform well but are not yet ready for real-world dialogues under specific requirements.*

1. Introduction

Large language models (LLMs) have the ability to solve natural language processing tasks, often without additional optimization processes for specific tasks. These models are trained with huge corpora and can generate text in different languages, in a plethora of scenarios. This versatility opens the door for specialized systems that have LLMs as a core of intelligent processing. Indeed, significant effort has been devoted to specialize the tasks and domains in which language models are expected to operate [Bommasani et al. 2022]. In this process, language models have become an integral part of information systems that support the automation of highly complex tasks. To attain real practical value, language models must be instructed with respect to the task of interest [Brown et al. 2020]. Prompt engineering employs a range of strategies so as to construct prompts for language models. These prompts, varying from simple instructions to role-playing commands [Chang et al. 2024a], optimize and direct responses generated by language models.

In this context, a key challenge is understanding and measuring how well a model is suited for a given task. LLMs are presented alongside a series of quantitative evaluations run on benchmark datasets [Chang et al. 2024b]. The performance on these datasets may not reflect the models' performance in any other given task, when subjected to fine-tuning procedures and when coupled with prompt engineering. In general, even benchmark datasets for complex tasks test the model's success with single requests. Real applications of these models are interactive, implying a higher complexity in behavior. This challenge gives rise to the problem of accrediting a model for use in specific tasks.

This paper addresses the evaluation of LLMs integrated into goal oriented chat-like information systems, where the LLM, prompt engineering strategies and imperative programming collaboratively handle specific tasks. By focusing well-defined goals, one

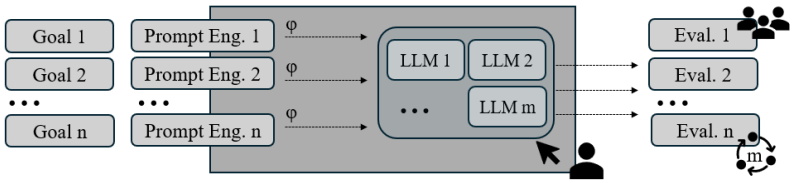

Figure 1. Goal-oriented evaluation system with LLMs as decision kernels: well-defined goals are established; for each goal, prompt engineering strategies are designed; users must interact with the system to elicit responses from each LLM; outputs reflecting the behavior of LLMs are provided for a goal-oriented and interactive evaluation process.

can run an interactive and goal-oriented evaluation of different models and achieve some degree of generalization of results. This paper thus presents a goal-oriented chat-like system, where the LLM model, supported by prompts and programming logic, is evaluated on its ability to guide the system's operations towards achieving its objectives. From the perspective of evaluation, a information system itself (Figure 1) serves as a means to tie goals and behaviors to the language models, testing them for real-world applications. In order to apply the broad strategy carried by a goal-oriented chat-like system, we have evaluated elements of a chatbot in a specific scenario. Several goals can be established for a chatbot; in our case we chose three general goals: engaging in a conversation with a target audience using suitable language; keeping them interested; staying within the specified domain. To achieve this, the chatbot must adopt the right persona, be able to talk about the chosen domain, and create an engaging conversational flow.

This paper is structured as follows. Section 2 introduces basic concepts of LLMs, their evaluation, and prompt engineering. Section 3 discusses related work. Section 4 describes our evaluation system, while Section 5 presents an instance of the system and the evaluation of two LLMs under the proposed domain. Section 6 concludes the paper.

2. Background

2.1. Large Language Models

Language models can be statistical or neural network-based and are designed to understand and generate human language. The basic concept behind their development involves processing a sequence of words or tokens to predict the probability of the next word or token in the sequence, or alternatively, a missing word or token in the sequence [Chang et al. 2024b, Zhao et al. 2023]. Language models have adopted increasingly larger neural network architectures and have been trained on vast amounts of data from a plethora of domains. This has led to the development of LLMs that serve as base for natural language processing and that can be adapted for various tasks [Zhao et al. 2023]. The success of LLMs is linked to the Transformer architecture and self-attention mechanisms [Vaswani et al. 2017]. The Transformer architecture takes, as input, word embeddings combined with positional embeddings that carry sequence information. Transformers use the self-attention mechanism, which is based on similarity scores between the embedding vectors input into the architecture. Successive calculations of these scores implement the extraction of a hierarchy of concepts, create a representation of contexts for the words (or tokens), useful to improve the performance of the models, and allow computational parallelism, providing efficiency to processing [Jurafsky 2024].

2.2. LLM Evaluation

How to measure the effectiveness of an LLM remains an open question. Due to the sizes and complexities of LLM architectures and the volume of data involved, traditional techniques (e.g. cross-validation, confusion matrix measures) are not satisfactory. Because LLMs are inherently used in tasks that involve language generation and comprehension, the evaluation must primarily consider linguistic aspects. Thus, the evaluation of LLMs has been aimed at analyzing their performance in tasks such as question answering and text summarization, and studying their abilities to, for example, avoid sensitive topics, handle unexpected inputs, and minimize hallucinations [Chang et al. 2024b]. The evaluation of LLMs has adopted three main strategies: quantitative, qualitative, and red-teaming evaluations. Quantitative evaluation involves the application of traditional or trained indicators that address a range of evaluation objectives, whether task-agnostic or not [Sai et al. 2022]. Qualitative evaluation entails allocating individuals to assess the correctness or appropriateness of LLM responses in specific usage contexts. Red-teaming evaluation involves intentionally inducing the model to make errors that compromise its safety mechanisms and reveal its vulnerabilities [Chowdhury et al. 2024]. However, it is well-known that these evaluation practices are not sufficient. Due to the success of LLMs in communicating with humans, their accreditation should focus on a user-centered evaluation. It should pay attention to the link between the LLM's behavior and the system's non-functional needs, ensuring user comfort and satisfaction in their interactions, and considering the broader impacts of using this system in society [Floridi and Cowls 2022].

2.3. Prompt Engineering

LLMs can be used in at least two schemes: the *pre-train and fine-tune paradigm* and the *pre-train, prompt, and predict* paradigm [Liu et al. 2023]. According to Liu et al. [Liu et al. 2023], under the latter paradigm, pre-trained LLMs are no longer adapted to solve downstream tasks but are instead invoked via a textual input (prompt) that reformulates the downstream task so as to make it more similar to the task for which the LLM was originally trained. Prompts can be as simple as a well-formulated question, or as complex as a composition of guidelines that constitute intermediate reasoning steps to guide the model's responses, a strategy known as "chain of thought". This paradigm reinforces the role of LLMs, as their applicability can be extended given a suitable set of prompts. In the *pre-train, prompt, and predict* paradigm, a new discipline called prompt engineering has emerged: "prompt engineering refers to the systematic design and optimization of input prompts to guide the responses of LLMs, ensuring accuracy, relevance, and coherence in the generated output" [Chen et al. 2024]. The use of prompt engineering methods amplifies the value of LLMs; however, it introduces a new layer of information into solution construction that must be assessed. Nevertheless, the evaluation of prompting methods is intertwined with the evaluation of the performance of the invoked LLMs, as it depends on the responses generated by the models to determine their effectiveness.

3. Related work

This section presents the key papers that influenced the design of the our system and prompt engineering solutions. Methods for evaluating LLMs have been discussed from the perspective of "what, where, and how" they must be evaluated [Sai et al. 2022]. An extensive evaluation of LLMs is presented by Liang et al. [Liang et al. 2023],

which, although using different types of metrics and proposing a holistic evaluation, still translates the quality of LLMs into particular and unrelated statistics. Evaluations beyond measures that reduce the conclusions to statistics are advocated by Lee et al [van der Lee et al. 2019]. The author argues that quality evaluation using such measures is controversial, and suggests that evaluations conducted by humans are a necessary alternative. Furthermore, live dialogue (the interactive task implemented in chatbot systems) evaluation is not covered by Liang et al. [Liang et al. 2023]. Regarding evaluations of chatbot systems, we report on the two initiatives [Sedoc et al. 2019, Lee et al. 2020]. The authors made efforts to create a framework that would contribute to advancing the standardization and systematization of evaluations of chatbot systems embedded with generative AI. In their protocol, an A/B paired testing is proposed and applied to ten systems.

Our evaluation system is based on two prompt engineering strategies: step-by-step thinking prompts [Kojima et al. 2023] and system-level prompt structures [Wu et al. 2022]. Takeshi et al. [Kojima et al. 2023] explored zero-shot chain of thought and concluded that by simply adding the phrase "let's think step by step" to guide the actions, the performance of a LLM can be significantly increased in terms of answer precision and logical coherence — a similar strategy was used in the *creator prompts* (Figure 2). On the other hand, the idea to implement the *inspectors prompts* (Figure 2) is inspired by proposals that use prompts as a way to build a control flow in the system in which the LLM is embedded. Wu et al. [Wu et al. 2022] used this strategy by chaining LLM steps so that the output of one step is the input of another. According to the authors, this approach also opened up possibilities for conducting unit tests regarding sub-components of a chain. The same possibility is present in the system proposed in this paper.

4. Goal-oriented Chat-like System for LLM Evaluation

This paper discuss how to accredit LLMs for specific tasks. An alternative to address this issue is proposed as a goal-oriented chat-like system (following the idea shown in Figure 1), in which different LLMs are compared based on their ability to drive the system towards its goals. For evaluation purposes, the same system and the same prompt engineering strategies must be used with each of the evaluated LLMs. We now examine how to build such a system from three perspectives: design, implementation and evaluation.

Design: We take that an information system that embodies the proposed evaluation strategy should be developed in a chat-like fashion: a user initiates an interaction with the system, and the interaction flow proceeds in simple stages. Each stage is formed by a user message followed by an LLM response. The LLM's response can be solely reactive to the user message, or it can incorporate a decision-making process that influences the dialogue path and user behavior. The set of goals for the information system must be designed for each instance established (see Section 5 for an example). Each goal must be associated with prompts designed to guide the LLMs under evaluation in achieving the goal.

In the current information system we have implemented to make our ideas concrete, we defined three general goals: engaging in a conversation using appropriate language to a target audience, keeping the conversation interesting and within a specific domain. The evaluation should determine whether, given appropriate prompt engineering, an LLM can (or cannot) ensure desired behaviors while a user interacts with the system. For this purpose, the general goals were broken down into the following small ones:

1. *Introduction and dialogue contextualization*: Given that the use of the system presupposes a dialogue with a user, it is expected that the model assumes a persona, and is capable of presenting it to the user, making them understand the purpose of that persona's existence within that interaction. This enables the user to understand the interaction. The LLM must also express an interest in getting to know the user so that it is possible to maintain a natural flow during the interaction.
2. *Persuasion and engagement*: Persuasion plays a crucial role in interactions in which one of the agents may be inclined to abandon it. For the LLM to be evaluated, the interaction needs to occur and last long enough to allow the production of evaluation data. Thus, the LLM must generate messages that can persuade the user to interact and to remain engaged in the interaction. At different moments of the interaction, the LLM should generate invitations and motivational messages.
3. *Restriction of the interaction scope*: The dialogue established in the interaction must be related to a specific topic. Therefore, the LLM must be able to understand whether a user's message is within or outside this topic. If the user, intentionally or not, diverges from the topic, the LLM must be able to bring them back.
4. *Appropriate language*: Establishing a persona and the objective of the dialogue generates an expectation regarding the type of discourse used by the LLM. The language must be correct concerning the topic under discussion and appropriate for the age group, education level, and other characteristics of the target audience.
5. *Elaboration of multiple-choice questions*: As the interaction progresses, the user's motivation may decrease. To assist the LLM in keeping the user engaged, the system prompts the LLM to create multiple-choice questions about the topic under discussion. Two types of questions are addressed: the first type has no incorrect answers but rather more or less appropriate responses within the context of the dialogue; the second type assumes there is one correct answer and should pertain to the topic discussed at that moment of the dialogue. The LLM must also react to the user's chosen answer by commenting on or correcting the choice made.
6. *Topic analysis*: To conclude the interaction with the user, the LLM produces an evaluation of the interaction by performing a topic analysis on the exchanged messages, and organizes a summary of topics extracted in this analysis.

Implementation: The implementation of the system requires a tightly coupled architecture (Figure 2) to determine the information system logic, and demands prompt engineering to invoke the LLMs. The interaction stages use two types of prompts[1,2]: **creator prompts** and **inspector prompt**. The former (**creator prompts**) are responsible for invoking the LLM to generate a message for the user. Each "creator" prompt comprises three elements. The first element is a memory of the last stage of the interaction (user and LLM messages). The second element is the declaration of the LLM persona, its goal, and the restrictions. The third element is the step-by-step declaration of what the model needs to do to generate an utterance for the speech, using a technique similar to the zero-shot chain of thought as the primary method. The latter (**inspector prompts**) are responsible for invoking the LLM to check the user's message intention and to verify if the user's

[1]The prompts were implemented for GPT models using the OpenAI Chat Completion API, and consist of three roles: *user*, the user's messages; *assistant*, the messages generated by the model; *system*, which defines the model's behavior. See `https://platform.openai.com/docs/overview`.

[2]For an evaluation focused on LLMs, we chose to keep the prompt engineering as simple as possible.

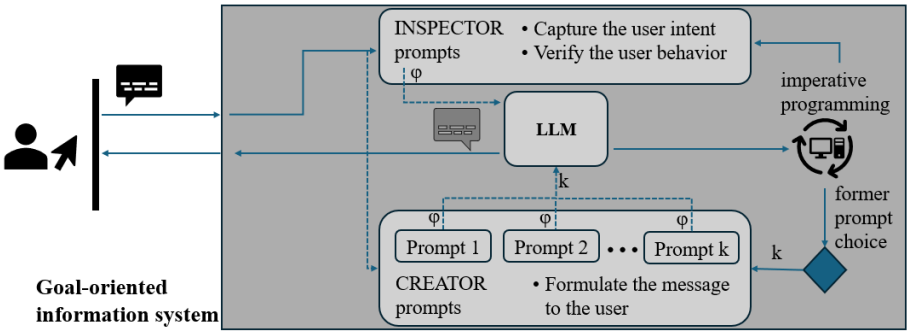

Figure 2. System architecture: imperative programming working with prompt engineering to properly invoke LLMs during a dialogue interaction.

behavior meets the requirements of the interaction at that stage. The LLM's response to a "inspector" prompt is processed by imperative programming so that an appropriate "creator" prompt is triggered[3]. The prompts systemic function is to keep the interaction restricted to the system's goals. In some phases of the dialog, there are multiple inspectors verifying different things. Each "inspector" prompt comprises two elements: the first element contains a memory that holds either the user message or both the user and LLM messages. The second element is the instruction for the LLM to analyze the utterances used in that dialog stage. Here, the use of few-shots technique is employed.

Each interaction stage may involve one or more prompts of each type. The dialogue unfolds through interconnected stages that culminate in a conclusion. Its progression is linear, passing through specific steps that are essential to complete the interaction. However, this linear progression does not imply that the dialogue always follows the exact sequence of steps, nor that each step consistently contains the same number of stages.

The prompt engineering strategies used in this implementation keep the LLM behavior on track, aiming to balance the model's creativity with the need to keep the conversation coherent by defining the necessary steps to generate a response. For example, although we expect an LLM to maintain an engaging dialogue, its primary focus must be on maintaining a coherent dialogue flow on a specific topic rather than engaging in aimless conversation; hence, the LLM must be controlled to reach this goal.

Evaluation: Defining evaluation strategies to accredit the LLMs of interest, with some degree of generalization, is the last challenge in the evaluation pipeline proposed here. Actually, the challenge is to construct an evaluation instrument that allows for verifying if the goals established for the system have been achieved, given the constructed prompt engineering and each LLM under evaluation.[4] Specifically, for the proposed system, it should be possible to verify if the LLM adheres to the steps proposed in the applied prompt engineering. The proposal for this phase of the research is to combine "exploratory testing"

[3]When the prompt inspector calls a prompt creator, the user's message is sent to the LLM again. Thus, each message is analyzed multiple times by the LLM, each time with the specified objective in the prompt.

[4]Our proposal is that different instances can be created. The idea is that from each instance, levels of generalization of LLM evaluation are possible. However, implementation decisions within the proposition of a specific instance can be conducted in different ways. Thus, determining the optimal prompt engineering design and assessing whether the information system logic and the prompt engineering are correctly coupled require ablation experiments that are beyond the scope of this paper.

[Pfahl et al. 2014] and an assisted evaluation: exploratory testing because human testers will use the system as end-users, learning through the testing process itself and exploring it deliberately and skillfully. Without relying on predefined scripts, they employ creativity and intuition to validate functionalities (whether the system's established goals have been achieved); assisted because, while using the system, they are guided by questions to be answered in a additional evaluation instrument. The assistance is important because this evaluation aims to identify specific behaviors of the LLM. This type of evaluation precedes an unassisted evaluation, in which users interact with the system freely and without specific guidance to study higher-level interaction requirements (interaction difficulty or satisfaction during interaction). Unassisted evaluation is outside the scope of this paper.

5. Instanciating the Goal-Oriented Chat-Like System

Instantiating the goal-oriented system described earlier means dealing with a concrete context in which the objectives and goals of the system align with objectives or desires of the relevant stakeholders. In this section we describe an instance of our system, named *Blabinha*, that deals with a gamified chatbot-like application. We present the resulting instance of the goal-oriented system and the results of exploratory testing and assisted evaluation, employing two versions of OpenAI's GPT model.

5.1. Blabinha: Blue Amazon' Superhero Challenge

We look for a dialogue where the user is invited to take on the challenge of creating a superhero to protect the Blue Amazon.[5] The path to achieving this goal (creating a superhero) involves the user gaining knowledge about the topic (the Blue Amazon) through dialogue with the system. The more knowledge the user acquires, the more powerful the superhero will be. The LLM holds the knowledge, and the user is a 10-year-old child. Figure 3 shows how the dialogue flow is organized. Broadly, the system is divided into three main phases: presentations and challenge invitation; dialogue loop about the Blue Amazon; and conversation analysis, scoring, and hero image generation.[6] The numbers in Figure 3 indicate at which part of the dialogue the behavior of the LLMs is oriented towards achieving each system goal, and the associated strategies are as follows:

1. For the *introduction*, prompts are designed for the LLM to introduce itself, discover the user's name, or give up on finding it. In *contextualization*, the LLM explains the "Blue Amazon" concept and invites participation in the challenge.
2. *Persuasion* and *engagement* are goals pursued throughout the interaction with the user. Persuasion is necessary when the LLM seeks to: understand the user's name; insist with the user when they do not accept to participate in the challenge; or insist that the user does not exit the dialogue when she expresses a desire to end the conversation. Engagement is promoted through positive reactions to the user's behavior, through the presentation of small challenges with multiple-choice questions (item 5), and through the final evaluation, which although left to the end of the interaction, it has the potential to retain the user for future interactions.

[5]Blue Amazon is a term used by the Brazilian Navy to refer to the Brazilian coastline and the exclusive economic zone of Brazil's maritime space.

[6]In the current version of the system, the LLM is not responsible for generating the score, and the superhero image generation module is not under evaluation.

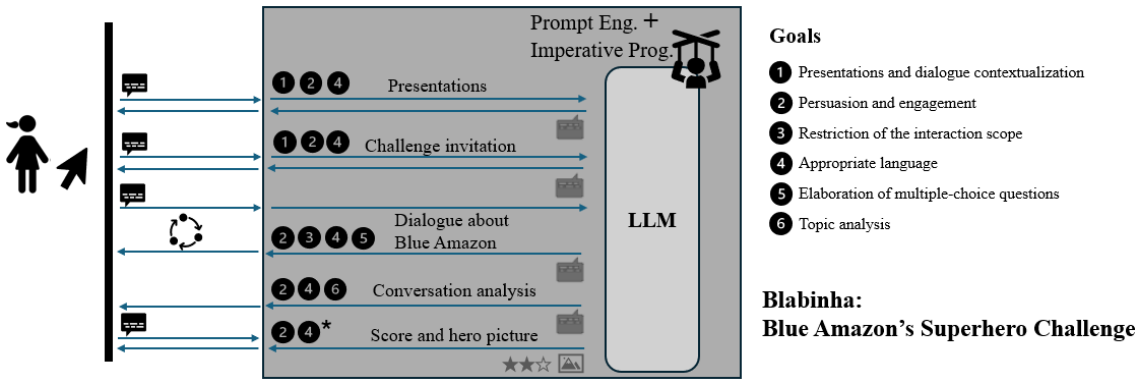

Figure 3. Dialogue flow for building the Blue Amazon's Superhero: the LLM generates the dialogue utterances and pursues the system's goals indicated by the numbered circles. Prompt engineering guides the LLM, supported by imperative programming to implement the system's logic.

```
Prompt CREATOR
[...{"role": "system", "content": "You are a robot named NAME and you are talking to a child" },
{"role": "system", "content": "Follow only the steps to generate the text:" +
    "Step 1 - If the person already knows about the Blue Amazon, congratulate them; if not," +
    "say something reassuring. Step 2 - Explain briefly that the challenge involves" +
    "creating a superhero and that to do so, they will need to learn about the Blue Amazon" +
    "Step 3 - INVITE them to participate in the challenge"},]
Prompt INSPECTOR
[...{"role": "system", "content": "You are a robot named NAME and you are talkin to a child" },
{"role": "system", "content": "Check if the person really wants to proceed to create the hero." +
    " Answer TRUE if the person say yes and FALSE if no."},]
```

Figure 4. Examples of creator and inspector prompts. For simplicity, only the role "system" is shown. The messages are originally in Portuguese and have been translated for the sake of better understanding.

3. *Restricting the dialogue to the topic "Blue Amazon"* is a goal for the LLM in the dialogue loop phase. Thus, prompt engineering guides the model by opening the context of the Blue Amazon to subjects related to the "sea AND Brazil". If the LLM recognizes the user straying from the topic, it is instructed to explain the digression and offer tips on subjects related to the Blue Amazon.
4. The model is informed via prompts that the system user is a child. Although it is then expected that simple and *appropriate language* be used, no further instruction about this is given to the model.
5. The *elaboration of multiple-choice questions* is used as a way to bring engagement. Preference option questions are presented when a particular subtopic (governance of the Blue Amazon) is mentioned during the dialogue. Multiple-choice questions with correct and incorrect alternatives, about the subtopic being discussed at the moment, are randomly triggered during the dialogue. In both cases, the LLM reacts to the option chosen by the user.
6. *Topic analysis* is carried out by a prompt that instructs the LLM to check whether "environment", "governance", or "resources" were addressed in the dialogue.

To exemplify prompts of the "creator" and "inspector" types, Figure 4 shows a "creator" prompt with an engagement reaction used to invite to the challenge; an "inspector" prompt that checks if the user accepted the invitation.

A typical interface for conversational agents was made available as a service. In this way, an interactive mode of engaging with the LLM during testing is provided. Two service instances were provided using two LLMs: *gpt-3.5-turbo* and *gpt-4-turbo*. During interaction, the tester can create multiple chat sessions in parallel. The state of the sessions is recorded so that an interaction can be interrupted and resumed at a later time. All sessions are persisted via a logging system and can be analyzed offline.

5.2. Evaluation setup and results

A group of 26 testers (from undergraduates to postdoctoral researchers), members of our broad research group but not involved in implementing the system, were invited in the evaluation process.[7] The group mainly consisted of researchers from computer science and engineering, with some from education and oceanography. Each tester used only one system instance (12 tested version *gpt-3.5-turbo* and 14 tested version *gpt-4-turbo*) and, although aware of the two instances, they did not know the differences between them, only knowing that they were using a system where a GPT family LLM was being used.

The evaluation process involved testers chatting with the system (interactive mode of use). They were asked to create multiple chat instances to check if the model's behavior was appropriate in different scenarios (exploratory testing). The interaction and evaluation were guided by a protocol with instructions for using the system and questions that directed reflection on the LLM's behavior and the expected system's goals (assisted evaluation). The protocol had 56 questions on various aspects for each system's goal. Testers had to create at least one chat where the conversation followed the expected flow (e.g. mention name, accept challenge, ask specific questions about the Blue Amazon). In other chats, testers could try different flows (e.g. refuse challenge, change topic). Testers were told not to do prompt attacks. The evaluation discussions presented herein cover 15 aspects related to the six goals established in the system. For the analysis of each aspect, only one question from the protocol was considered. The remaining questions addressed other aspects of the same goals, but due to space constraints, only a subset of aspects is discussed here. The discussions are divided into quantitative and qualitative perspectives.

Quantitative evaluation: Table 1 lists the results from the testers' responses in the evaluation form, considering the success in meeting the target/evaluation aspects across all the chats they created. The questions considered are 5-point Likert scales or multiple-choice questions. In the multiple-choice questions, the options range from "failure in all chats" to "success in all chats". For the 5-point Likert scale, we considered "success case" when the highest point on the scale was chosen. The results presented in Table 1 allow for comparative analyses between the LLMs tested and their accreditation considering an acceptance threshold. The relative values shown in the table refer to the success rate reported by the testers. For this success rate and for each evaluated aspect, confidence intervals for proportions were calculated using the Wilson method [Kvam and Vidakovic 2007][8].

For comparing LLMs based on their success rates, a proportions z-test [Altman et al. 2013], with a Bonferroni correction [Altman et al. 2013], was conducted. Bold Highlights in Table 1 indicate the statistically significant comparative success rate

[7]The system's code, evaluation form, anonymized logged chats examples, and anonymized testers' form responses are available at: `https://github.com/C4AI/Blabinha`.

[8]The Wilson method was chosen for handling small samples (small number of testers).

Table 1. Number of testers (absolute values; relative values) who reported success in meeting the target evaluation aspect in all the chats they created: 6 goals, 12 testers for GPT 3.5 and 14 testers for GPT 4.

		GPT models			GPT models	
	Aspect	*3.5*	*4*	**Aspect**	*3.5*	*4*
1	Introductions	7; 58.3	8; 57.1	Correcteness*	11; 91.7	13; 92.9
2	Challenge acceptance	**7; 58.3**	6; 42.9	Keep the interaction	7; 58.3	9; 64.3
3	Scope restriction (Blue Amazon)	**8; 66.7**	5; 35.7	Scope restriction (Superhero)	2; 16.7	**5; 35.7**
4	Challenge rules	3; 25.0	**7; 50.0**	Domain explanation	7; 58.3	10; 71.4
4	Domain concepts	**8; 66.7**	6; 42.9	Simple vocabulary	4; 33.3	5; 35.7
5	Subject covered	5; 41.7	9; 64.3	Formulation quality*	6; 50.0	**13; 92.9**
5	One correct	2; 16.7	**12; 85.7**			
6	Topic analysis (bonus)	4; 33.3	**9; 64.3**	Topic analysis (dialog)	4; 33.3	**7; 50.0**
Average width of confidence intervals: 46 points % for GPT 3.5; and 43 points % for GPT 4						

according to this test. Due to the subjective nature of human analyses, uncertainties were investigated through the width and overlap of confidence intervals. The intervals' width revealed uncertainty, but for the "One correct" aspect, there was no overlap, and for the "Formulation quality" aspect, the overlap was extremely small (0.06 percentage points). This indicates a high level of confidence in the evaluation of both aspects.

Success rates above a threshold can indicate a means of accrediting the LLMs, although each evaluation aspect may be more or less important and may imply specialized thresholds for each system goal. For simplicity, a success rate requirement of at least 90% is assumed for all aspects. According to this criterion, aspects where at least one LLM is accredited are annotated with an asterisk (*) in Table 1. Following this analysis, the tested LLMs would not be accredited under any goal, as they do not achieve the expected quality in any complete set of aspects associated with the goals.

Qualitative evaluation: Based on the testers' interactions with the system, insights about the competence of the LLMs in guiding the dialogue were gathered, for instance:

1. Some testers have reported that both models can have difficulty comprehending the tester's name at first glance, especially with names that are uncommon in the language of the prompts. To address this issue, testers should explicitly state their name by beginning with "*My first name is ...*".
2. Regarding persuasion and engagegement goals testers reported instances where the models did not fully comprehend responses related to agreeing or disagreeing to continue the interaction, mainly when the response was not accompanied by context and consisted of brief phrases or single words such as "*Ok*" and "*Yes*".
3. When discussing restricting the scope of the dialogue, testers attempted to explicitly discuss a topic that does not revolve around the Blue Amazon. The models performed well in stating that there was an attempt to diverge from the theme. When the user's utterance touched on topics indirectly related to the Blue Amazon, such as the Navy and oil, the models erroneously perceived it as divergent.
4. At the beginning of the interaction, when the model introduces the dialogue participants, testers noted that the *gpt-3.5-turbo* model's rigid explanation style

could be challenging for children, while the *gpt-4-turbo* model's informal language was more child-friendly. The challenge rules and the Blue Amazon concept were clearer with the *gpt-4-turbo* model's discourse than with the *gpt-3.5-turbo* model's. Overall, the *gpt-4-turbo* model presented an easy-to-grasp discourse.

5. Most of the *gpt-4-turbo* testers stated that all processes involving multiple-choice questions were appropriate. The questions had one correct answer, were about the Blue Amazon, most related to the last topic of the dialogue, and if the tester chose an incorrect answer, the model provided the correct one. Testers of the *gpt-3.5-turbo* model reported that the questions sometimes had multiple correct answers or no correct answer at all, and the model corrected the tester by providing an answer not included in the previously provided choices.

6. Conclusions

We proposed a strategy for evaluation of LLMs placed within conversational agents applied to real-world scenarios. A goal-oriented chat-like system was introduced in somewhat abstract terms, and then instantiated (*Blabinha* system), to evaluate the capability of LLMs to run a dialogue within specific expectations. The key feature of the proposed system and evaluation is the support for observing and accrediting the LLM during a dialogue (interactive task). Such evaluations are still rarely reported in the literature. Our experiment shows that our strategy is appropriate in practice, primarily by establishing measurable parameters for the accreditation of models. However, we identified two aspects for further investigation to enhance the process robustness: an ablation procedure to determine the extent to which prompt engineering procedures contribute to inadequate LLM behavior; the specialization of testers in evaluating individual aspects to improve the conditions for achieving statistical significance in the results and to minimize the effects of subjectivity, criteria drift [Shankar et al. 2024] and potential evaluator fatigue.

Acknowledgements

The authors thank the Center for Artificial Intelligence (C4AI-USP) and the support from the São Paulo Research Foundation (FAPESP grant #2019/07665-4) and from the IBM Corporation. M. Fantinato and F. G. Cozman thanks the National Council for Scientific and Technological Development of Brazil (resp. CNPq grants #312630/2021-2 and #305753/2022-3). The authors thank everyone who evaluated the Blabinha system.

References

Altman, D., Machin, D., Bryant, T., and Gardner, M. (2013). *Statistics with Confidence: Confidence Intervals and Statistical Guidelines*. Wiley.

Bommasani, R. et al. (2022). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Brown, T. et al. (2020). Language models are few-shot learners. In *Advances in Neural Inf. Processing Syst.*, volume 33, pages 1877–1901.

Chang, K., Xu, S., Wang, C., Luo, Y., Xiao, T., and Zhu, J. (2024a). Efficient prompting methods for large language models: A survey. *arXiv preprint arXiv:2404.01077*.

Chang, Y. et al. (2024b). A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Chen, B., Zhang, Z., Langrené, N., and Zhu, S. (2024). Unleashing the potential of prompt engineering: a comprehensive review. *arXiv preprint arXiv:2310.14735*.

Chowdhury, A. G. et al. (2024). Breaking down the defenses: A comparative survey of attacks on large language models. *arXiv preprint arXiv:2403.04786*.

Floridi, L. and Cowls, J. (2022). A unified framework of five principles for ai in society. *Machine learning and the city: Applications in architecture and urban design*, pages 535–545.

Jurafsky, D, M.-J. H. (2024). *Speech and Language Processing*. 3rd (draft) edition.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2023). Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Kvam, P. H. and Vidakovic, B. (2007). *Nonparametric Statistics with Applications to Science and Engineering*. Wiley-Interscience, USA.

Lee, S., Lim, H., and Sedoc, J. (2020). An evaluation protocol for generative conversational systems. *arXiv preprint arXiv:2010.12741*.

Liang, P. et al. (2023). Holistic evaluation of language models. *Transactions on Machine Learning Research*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Pfahl, D., Yin, H., Mäntylä, M. V., and Münch, J. (2014). How is exploratory testing used? a state-of-the-practice survey. In *Proc. of the 8th ACM/IEEE Int. Symp. on Empirical Software Eng. and Meas.*, New York, NY, USA. ACM.

Sai, A. B., Mohankumar, A. K., and Khapra, M. M. (2022). A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2).

Sedoc, J., Ippolito, D., Kirubarajan, A., Thirani, J., Ungar, L., and Callison-Burch, C. (2019). ChatEval: A tool for chatbot evaluation. In *Proc. of the 2019 Conf. of the North American Chapter of the ACL (Demonstrations)*, pages 60–65. ACL.

Shankar, S., Zamfirescu-Pereira, J. D., Hartmann, B., Parameswaran, A. G., and Arawjo, I. (2024). Who validates the validators? aligning LLM-assisted evaluation of LLM outputs with human preferences. *arXiv preprint arXiv:2404.12272*.

van der Lee, C., Gatt, A., van Miltenburg, E., Wubben, S., and Krahmer, E. (2019). Best practices for the human evaluation of automatically generated text. In *Proc. of the 12th Int. Conf. on Nat. Lang. Gener.*, pages 355–368. ACL.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In *Adv. in Neural Inf. Proces. Syst.*, volume 30.

Wu, T., Terry, M., and Cai, C. J. (2022). AI Chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proc. of the 2022 Conf. on Hum. Factors in Comp. Syst.* ACM.

Zhao, W. X. et al. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.