Coordination within Conversational Agents with Multiple Sources

Vinícius Bitencourt Matos[1], **Ricardo Grava**[2], **Rodrigo Tavares**[2], **Marcos Menon José**[2], **Paulo Pirozelli**[3], **Anarosa A. F. Brandão**[2], **Sarajane M. Peres**[4], **Fabio G. Cozman**[2]

[1]Center for Artificial Intelligence (C4AI) — Universidade de São Paulo, Brazil

[2]Escola Politécnica — Universidade de São Paulo, Brazil

[3]Instituto de Estudos Avançados — Universidade de São Paulo, Brazil

[4]Escola de Artes, Ciências e Humanidades — Universidade de São Paulo, Brazil

vinicius.matos@alumni.usp.br

{rsgrava,rodrigotavares,marcos.jose,paulo.pirozelli.silva}@usp.br

{anarosa.brandao,sarajane,fgcozman}@usp.br

Abstract. *Conversational agents can now operate with language models, rules, ontologies and varied other sources to provide smooth dialogue. However, the coordination of multiple sources in conversational agents is a challenge. We present a mechanism to effectively orchestrate multiple sources in a conversational agent, by relying on a client-server approach with an associated prompt generation module that deals with heterogeneous domain-oriented modules. As a detailed use case, we describe the architecture of a chatbot specialised in topics related to the Brazilian coast, and we study the benefits of our approach.*

1. Introduction

The development of conversational agents engages one in many complex challenges, from the effort to organise and represent knowledge to the need to give fluidity and context to the dialogue. Moreover, for a conversational agent to be successful, it must contain modules that can manipulate various sources of information. To control all such modules and services, it is essential to have a loosely coupled but tightly integrated system architecture with an orchestration mechanism.

In this paper, we present a system architecture that can be used to effectively instantiate a conversational agent, where we take into account the recent explosive growth of Large Language Models (LLMs). Strategies entirely based on static rules do not seem appropriate given such LLMs, so we move to a more flexible scheme where instructions can be passed around in (suitably restricted) natural language. Our architecture adopts a client-server approach endowed with a prompt generation module to deal with heterogeneous resources. Our main contributions are related to the orchestration mechanism of the system. We feel that our solution can be effective with many other conversational agents, from shopping assistants to academic tutors.

We summarise relevant previous efforts in Section 2; there we describe existing architectures for conversational agents and the recent impact of LLMs over those agents.

We then present our solution in Section 3, where we describe the orchestration mechanism that we have developed. In Section 4, we present a detailed use case, where we apply our proposed techniques to a particular conversational agent. We first describe BLAB (BLue Amazon Brain), an agent that specialises in the Brazilian region of the Southern Atlantic. We go over the architecture and the modules of this agent, and then explain the orchestration of modules and the results we have obtained in our implementation.[1] Finally, we discuss our experience with our architecture in Section 5.

2. Background: Conversational Agents

Conversational agents, or dialogue systems, capture many aspects of artificial intelligence; here we adopt the conventions and terminology used by Jurafsky and Martin [10]. We thus take such agents to be either *task-oriented* ones, where the conversation aims at completing specific tasks, or *chatbots*, where extended conversations with broader goals must be allowed. A dialogue is a sequence of turns, each one by an agent engaged in the dialogue; at each turn an agent expresses a *speech act* that may offer an answer, a claim, an advice, a question, and so on [17]. Most conversational agents, and in particular most chatbots, are either rule-based or corpus-based systems.

Rule-based systems, such as ones based on Watson Assistant[2] or Rasa,[3] offer in fact a modern and usable version of the old expert system shells. The programmer must insert all relevant rules to detect the user's intention in each step so that the dialogue can be run towards the desired end. The programmer may not have advanced knowledge about artificial intelligence techniques, but rules and related machinery must cover all possible situations that are to be expected during the dialogue. Although restricted in promoting a fluid dialogue, such systems guarantee correctness and coherence.

Corpus-based systems may rely on responses that are generated by retrieval from a stored corpus, or on utterances that are generated by language models. Perhaps no conversational system has been more popular than ChatGPT, a chatbot based on a (very) Large Language Model [13]. Even though such systems are currently very effective and have the ability to act in open domain tasks, they do not provide guarantees of correctness and coherence in their responses [3].

The fact that conversational agents provide information may raise demands related to truthfulness and non-discriminatory behaviour. Not only may filters be needed, but also some guarantees that information is extracted from reliable sources. Such sources may be, for instance, annotated corpora or curated knowledge graphs. A conversational system must infuse some trust in its users so that interactions can be in fact useful.

In practice, conversational agents that combine all such strategies and goals can be found in the literature, as complex dialogue may require a multitude of strategies [11]. Hence, client-server systems, where modules interact through an orchestration engine, may be needed in agents that can be extended with tasks such as accreditation, argumentation and content filtering.

Several frameworks that combine language models with specialised tools have

[1] Our code base is available at https://github.com/C4AI/blab.

[2] https://www.ibm.com/products/watson-assistant

[3] https://rasa.com/docs/

been developed and published recently [7, 16, 18]. Their aim is to overcome the limitations of Large Language Models (LLMs): despite being able to generate text in various formats and styles following instructions in natural language, LLMs often fail when precise information is required, producing preposterous answers due to errors that stem, for instance, from their lack of access to structured and curated knowledge bases. LangChain [7] is a framework for developing applications that connect LLMs to data sources, allowing the models to query and interact with external resources. Similarly, Jarvis/HuggingGPT [18] is a framework that implements the integration between GPT and HuggingFace AI models to solve complex tasks. Toolformer [16] is a model that learns how to use external tools, when to call them and which arguments should be passed, and how to interpret their output to achieve a goal.

A conversational agent gets text from a user and returns an answer; in a sense, every such agent is an 'answerer'. However, there are techniques that are specifically developed to answer questions, and agents whose main goal is to answer specific questions are often called Q&A systems, or simply answerers. In this paper, we use the term *answerer* to refer to a module that returns answers to questions focusing on some relatively narrow theme (for instance, football games or economic forecasts).

Q&A systems have in fact been studied within Natural Language Processing for decades, with applications dating back to 1961 [9]. Since then, Q&A has been widely used to dynamically support human users. Current Q&A systems often resort to two blocks, a Retriever and a Reader [8, 12]. The Retriever gathers relevant passages from a given corpus that are then concatenated with the question. The Reader then generates the answer, usually by relying on an LLM. This is necessary as there is a limitation in the amount of knowledge that an LLM can store and update within its connections; the Retriever is used to collect information from large databases such as Wikipedia.

3. Coordination within a Conversational Agent

In this section we propose answers to a number of questions related to orchestration of modules within conversational agents. How can various modules be integrated? How can an LLM be used together with rule-based decisions and other language models? How can we harness the power of LLMs to help with orchestration?

We assume that a conversational agent of interest is built within a simple conceptual framework: the agent must have at least one user interface, perhaps a simple text-based or even a robot-like interface; a set of modules that contribute to the output, some of which are specialised answerers while others may be general purpose components; a controller that orchestrates the communication between all modules and user interfaces. In order to provide answers that are accurate about specific topics, we take that our agent depends on answers generated by specialised answerers rather than end-to-end all-purpose language models. In Section 4 we describe a concrete conversational agent where all such modules have been implemented.

We have pursued a solution that exploits the planning and natural language interpretation skills of LLMs. These skills are employed to make decisions and transform text in a dialogue that seamlessly integrates multiple answerers. In doing so, our work has similarities with the systems discussed in the last paragraph of the previous section. However, it should be noted that our objective here is, in a way, opposite to those efforts:

```
Given the following message history:

USER: «[message 1]»
BOT:  «[message 2]»
⋮
USER: «[message 2k−1]»

Rewrite the last message sent by the user by removing errors and
completing it with all the information necessary for an automatic
answerer to interpret it without having access to previous messages,
without null subjects and pronouns referring to external terms.
```

Listing 1. The Correction request in our implementation.

rather than using external tools to improve the accuracy of LLMs, we wish to employ LLMs as part of our orchestration mechanism, acting as a real-time interpreter between the user and various modules of a conversational agent.

Dialogue construction by such a strategy is not straightforward; it involves several important challenges that arise from the fact that answerers are built to generate answers to narrow and independent questions, not to act as conversational agents. The challenges are coordination ones.

3.1. Coordination Challenges

We highlight a few challenges that must be addressed to properly orchestrate answerers within a conversational agent; namely, sensitivity, state, selection and output challenges.

Sensitivity. It is not uncommon for user messages to contain grammar and spelling errors, as well as typical expressions of informal speech. If they are used as input to answerers, which were designed to handle properly written questions, incorrect answers are likely to be returned.

State. In a dialogue, it is not always possible to fully grasp the intended meaning of a message without considering the context in which it was sent. As most answerers can only deal with a single self-contained question, they may misinterpret isolated messages if their unprocessed contents are used as prompts.

Answerer selection. Given a question, it is necessary to decide which answerer should receive it. This is a nontrivial natural language processing task, in particular because the capabilities of each answerer must be taken into account — they may be specialised in different subtopics and may have been designed to handle specific types of questions.

Output. Specialised answerers typically generate short replies that directly answer posed questions but that are *not* effective in providing the conversational experience generally expected by the user engaged in a dialogue.

```
Given the following question-answering bots:

1:   «[description of bot 1]»
⋮
[n]: «[description of bot n]»


Which one will most likely answer the following message correctly?

«[corrected user message]»

If none of the bots can handle the requested topic or if the message is
 wrong, answer 0. Send ONLY the number, without text.
```

Listing 2. The Redirection request in our implementation. Note the use of the capitalised word ONLY.

```
Given the following message history:

USER: «[message 1]»
BOT:  «[message 2]»
⋮
BOT:  «[message 2k]»

Rewrite the last message sent from the bot and convert it into a
complete and concise sentence, regardless of whether it is correct.
```

Listing 3. The Completion request in our implementation.

3.2. Our Solution: Template-based Prompting To Orchestrate Dialogues

To overcome the challenges described in the previous subsection, we suggest that template-based prompts in natural language can be the orchestrating ‘glue’ in a conversational agent. That is, instead of designing domain specific protocols for the various communication endeavours, one can use LLMs themselves to support orchestration. In the remainder of this paper we assume that the agent relies on one particular LLM for orchestration; we refer to it as ‘the’ LLM (even though the agent may count on answerers that also rely on LLMs).

We take that for each question posed by the user, three requests to the LLM must be made, as a general protocol to guide the dialogue creation. We now describe the requests.

Correction. The first request deals with sensitivity and state challenges, and consists of pre-processing the user’s message and its context; we refer to it as the ‘Correction’ request.

When the user sends a message, an LLM is asked to rephrase it considering the full dialogue history. The result must be a single sentence that is both correct (in terms of spelling, punctuation, grammar, etc.) and self-contained (i.e. any references to previous messages — such as pronouns and null-subject sentences — must be replaced with the terms to which they refer). That is, this task tackles the first two challenges raised in the previous subsection.

Listing 1 shows a template prompt that works in our implementation with GPT-3.5, a modified version of the GPT-3 language model [4]. The construction of such a prompt is a nontrivial trial-and-error process.

Redirection. The second request deals with the challenge of choosing the answerer that will receive the question; we refer to it as the 'Redirection' request.

In short, with the corrected question in hand, the next command to the LLM is a request to decide which of the available answerers is the most suitable for handling it, or detect that the message is not supported.

The template prompt we use in our implementation is shown in Listing 2. Again, this prompt demanded a long trial-and-error process. In particular, it is interesting to note that, at least for GPT-3.5, we had to use the capitalised word ONLY to force the LLM to return a simple text without any additional explanation (without this word, GPT-3.5 provided directives containing superfluous explanatory text that would have required some additional parsing). The selected answerer is then called with the corrected and contextualised question.

Completion. After the selected answerer delivers its answer, it is necessary to generate a complete yet concise sentence, to avoid the short replies that are usually produced by question answering systems. We do so with a final 'Completion' request.

The Completion request envelopes the answerer output with additional instructions that avoid the last issue discussed in the previous subsection. Listing 3 contains a template prompt for this request, again obtained after substantial trial-and-error effort.

Note that in our implementation the Completion request (Listing 3) asks the LLM *not* to interfere in the contents of the answer, even if it can detect errors. This design decision was needed as the fact-checking abilities of current LLMs may not be satisfactory, in particular when we have the output of an already specialised answerer.

Figure 1 illustrates the overall communication flow. Tables 1 and 2 show an example of dialogue with two questions using the proposed solution ('←' and '→' denote 'from' and 'to', respectively, from the chatbot system's point of view).

4. A Detailed Use Case: the BLAB Chatbot

In this section, we describe a use case for the framework we proposed in Section 3. The first subsection briefly presents the *raison d'être* of BLAB, a large research effort to disseminate knowledge about the Brazilian maritime territory using artificial intelligence. The second subsection contains an overview of BLAB's service-oriented infrastructure and its components, and the last subsection thoroughly explains how the communication between the controller and its clients (users, answerers and LLMs) is orchestrated.

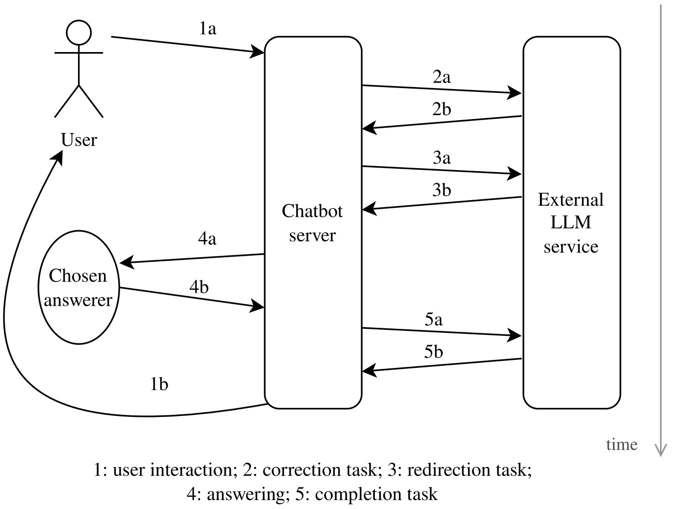

1a
User
2a
2b
3a
3b
Chatbot server
External LLM service
4a
Chosen answerer
4b
5a
5b
1b
time
1: user interaction; 2: correction task; 3: redirection task;
4: answering; 5: completion task

Figure 1. The proposed communication flow.

4.1. BLAB: Application Domain

Brazil's maritime territory is also known as the 'Blue Amazon'; the name makes reference to the region's size, wealth of natural resources, biodiversity, and ecosystem services. The Blue Amazon comprises Brazil's continental shelf and Exclusive Economic Zone — the area over which Brazil has sovereignty in terms of natural resources. It contains 90% of Brazil's oil reserves and 77% of its gas reserves [2]; carries 95% of Brazil's international trade, as well as other economic activities; is a vital source of food supply; and is a key player in climate regulation [1]. With respect to biodiversity, the Blue Amazon encompasses multiple distinct ecosystems, such as mangroves, estuaries and coral reefs; it is also a unique environment for fauna and flora [14]. Despite its importance to Brazil's economy and the South Atlantic environment, the Blue Amazon remains relatively unknown to most people, including those who live in coastal areas. Information about the region is usually restricted to academic papers, government reports and technical databases, with few tools available for a wider audience.

BLAB is a large effort to provide a platform of services with the aim of organising and sharing knowledge about this region [15]. This objective aligns with global efforts to disseminate technical and scientific knowledge, aiming to minimise the phenomenon of information asymmetry and, consequently, promote awareness of climate responsibility. Noting the current focus on the Ocean Decade[4] and on the 17 Sustainable Development Goals (with emphasis on Goal 14 — life in water)[5], both United Nations initiatives, BLAB represents an example of the application of AI for good[6].

BLAB was conceived as an implementation of the dialogue coordination described in Section 3. It encompasses a number of complex and interconnected services, allowing the seamless incorporation of additional components. By now, the operating services include a conversational agent, a news reporter, and a purposefully-developed wiki,

[4] https://oceandecade.org/
[5] https://sdgs.un.org/goals
[6] https://aiforgood.itu.int/

Description	From/To	Content
1a. Raw question	← User	‘*what is the blue amazon*’ (*)
2a. Correction request	→ LLM	(see Listing 1)
2b. Corrected question	← LLM	‘*What does the term “Blue Amazon” mean?*’
3a. Redirection request	→ LLM	(see Listing 2)
3b. Chosen bot number	← LLM	‘*1*’
4a. Corrected question	→ Ans. 1	(= *2b*)
4b. Raw answer	← Ans. 1	‘*This is the name given to the sea region that belongs to Brazil, from the beaches to the ocean. The name is a reference to the Amazon rainforest due to the similarity in the wealth of living resources, minerals, energy and biodiversity found in the two regions.*’
5a. Completion prompt	→ LLM	(see Listing 3)
5b. Completed message	← LLM	‘*The Blue Amazon is the region of the sea that belongs to Brazil, from the beaches to the ocean, and it is rich in living resources, minerals, energy and biodiversity, similar to the Amazon rainforest.*’
1b. Completed message	→ User	(= *5b*)

(*) The question has been intentionally written without capitalisation and punctuation.

Table 1. First example of the internal communication in a dialogue.

all of them organised in a portal. The conversational agent is the main piece of the BLAB architecture, which directly engages with users in natural language (Portuguese). The other two services, the reporter and the wiki, work independently and will not be covered in this text. The remaining part of this section will concentrate solely on the chatbot.

4.2. The BLAB Chatbot

The operation of the BLAB Chatbot depends on a service-oriented infrastructure (back-end) and on at least one user client (front-end). A broad picture is depicted in Figure 2.

The agent consists of:

- a controller, which receives, stores and delivers messages;
- the clients for answerers (in the grey area), which interface the conversational agent with external answering services that are specialised in one or more subjects;
- the interpreter client, which requests tasks to the LLM and parses the output text so as to make real-time decisions concerning the dialogue flow;
- the client for LLM, which interfaces the conversational agent with external LLM services;
- at least one user interface (e.g. the web client and the client for Robios Go robot/avatar), which consumes the API provided by the controller and allows the user to interact with the system.

Description	From/To	Content
1a. Raw question	← User	‘*what is the name of that ocean?*’ (*)
2a. Correction request *2b.* Corrected question	→ LLM ← LLM	(see Listing 1) ‘*What is the name of the ocean where the Blue Amazon is located?*’
3a. Redirection request *3b.* Chosen bot number	→ LLM ← LLM	(see Listing 2) ‘*2*’
4a. Corrected question *4b.* Raw answer	→ Ans. 2 ← Ans. 2	(= *2b*) ‘*Atlantic Ocean*’
5a. Completion prompt *5b.* Completed message	→ LLM ← LLM	(see Listing 3) ‘*The name of that ocean is Atlantic Ocean.*’
1b. Completed message	→ User	(= *5b*)

(*) The question has been intentionally written without capitalisation.

Table 2. Second example of the internal communication in a dialogue.

4.2.1. Coordination within the BLAB Chatbot

This chatbot operates through an implementation of the ideas proposed previously.

In our current implementation, the controller communicates with several clients: some of them are responsible for providing answers to user questions (DEEPAGÉ, Watson, Haystack[7], and Rasa[8])[9]; one client handles the integration of an LLM, which adjusts the messages from both the user and the answerers, and makes decisions regarding the suitability of answerers for the user messages; finally, the interpreter client ensures the formulation of appropriate prompts for the LLM, based on the interactive dynamics between the user and the chatbot. The LLM service we have used is OpenAI's GPT-3.5.[10]

For instance, in order to formulate a proper prompt so that GPT can suggest the most suitable answerer for a question, the interpreter client uses the following descriptions for this particular set of answerers:

1. a Watson Assistant bot developed by [6]: ‘A system that answers questions in Portuguese about the Brazilian coast, the Brazilian Exclusive Economic Zone and related topics. Ideal for questions that require long answers and cannot be answered briefly.’
2. DEEPAGÉ [5]: ‘A system that answers questions in Portuguese about the Brazilian coast, the Brazilian Exclusive Economic Zone and related topics. Ideal for questions that can be answered directly, with few words, such as names and dates.’

The former is a question answering system that combines the BM25 algorithm, a sparse retrieval technique, with PTT5, a pre-trained state-of-the-art language model, and can

[7] Haystack is an open-source NLP framework that uses transformer models. https://haystack.deepset.ai/

[8] Rasa is an open-source framework for building chat assistants. https://rasa.com/docs/

[9] The clients for Haystack and Rasa are integrated into the current implementation without specific answerers.

[10] More precisely, `gpt-3.5-turbo` from https://platform.openai.com/docs/models.

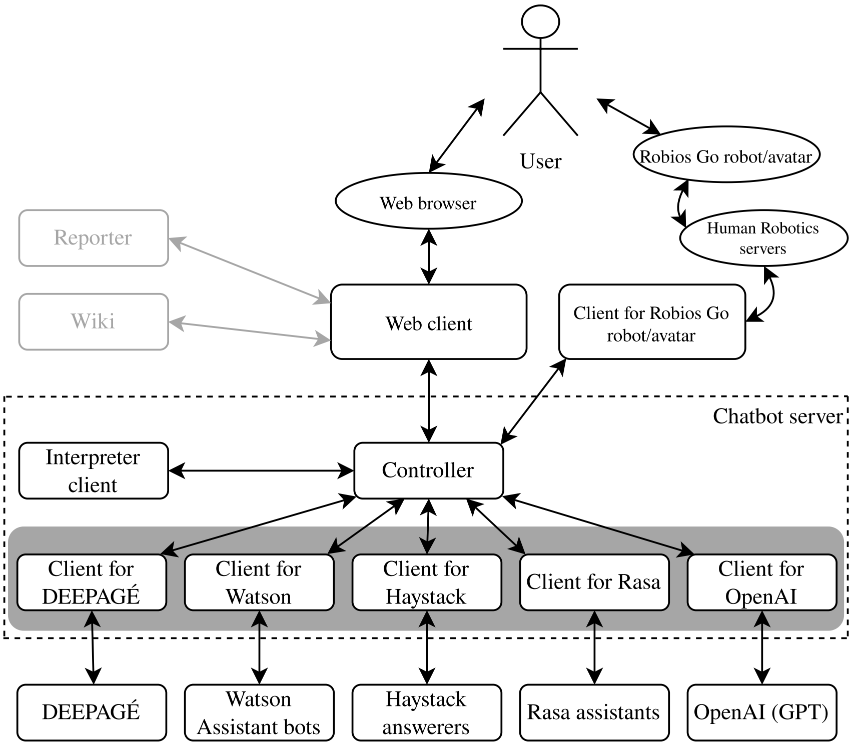


Figure 2. Overview of the BLAB Chatbot components.

answer questions in Portuguese about the Brazilian environment [5], whereas the latter is a bot created using Watson Assistant which is capable of answering questions about the various topics contained in it [the Blue Amazon], such as preservation and living, energy and mineral resources [6]. In the example given in Section 3, as expected, the first question (Table 1) is redirected to answerer 1, as it requires a long definition, whereas the second question (Table 2) is sent to answerer 2 because the answer is only a name.

Even though we have only tested the GPT-3.5 LLM, the architecture supports the inclusion of similar services that can perform Correction, Redirection and Completion (Section 3). These are the only tasks that our current implementation delegates to GPT, i.e. we do not use GPT to generate answers to user questions or to verify the accuracy of the obtained answers.

The Correction request can be of use even when a dialogue contains a single question-answering system. The example presented in Tables 1 and 2 shows that implicit references are replaced with the corresponding terms, which is required because the input expected by most answerers is a self-contained question, without the dialogue history. Moreover, the following example shows that the dialogue can benefit of the Correction request even there is only one answerer and no context is required. As described in [5], the DEEPAGÉ answerer yields the correct answer to the following question about a Brazilian archipelago:

> — *Quando Fernando de Noronha se tornou um Patrimônio Mundial da UNESCO?* (‘When did Fernando de Noronha become a UNESCO World Heritage Site?’)
> — *2001.*

However, if the word *'patrimônio'* (Brazilian Portuguese spelling) is replaced with *'patrimonio'* (misspelling) or even the variant *'património'* (European Portuguese spelling), the generated answer is incorrect:

— *1985.*

This illustrates the sensitivity issue that was mentioned in Section 3: incorrect answers can be produced if the question contains common errors (such as a missing diacritic) or even words written in another variety of the language. As the writing is standardised by the LLM through the Correction request, DEEPAGÉ's answer is correct when it is used through BLAB. This shows that the Correction request can be used to augment question-answering systems by allowing them to handle user questions that would otherwise be answered incorrectly.

Furthermore, the Completion request converts the blunt reply (containing only the year, without words) into a full sentence, improving the user experience in the dialogue:

> — *O arquipélago de Fernando de Noronha se tornou um patrimônio mundial da UNESCO em 2001.* ('The archipelago of Fernando de Noronha became a UNESCO World Heritage site in 2001.')

4.2.2. The User Interface for the BLAB Chatbot

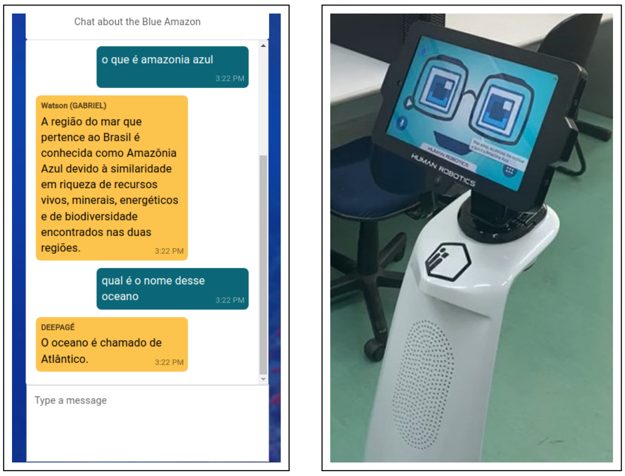
can in fact operate through two interface modules. First, a web-based interface embedded in a web-portal related to the Blue Amazon; second, a social-robot interface in a Robios Go[11] robot.

Web portal. The web-based interface is designed in the form of a dialogue window and is embedded in a web portal along with the other BLAB services. One of the main purposes of the portal is to share knowledge, information and news about the Blue Amazon, and thus it must be accessible by the widest possible audience. Hence, the development has been guided by the Web Content Accessibility Guideline's (WCAG) principles [19], by which the portal must be *perceivable* (content has to be presented in ways that are can be perceived by the user's senses), *operable* (it must be possible to operate the portal only with interactions that the user is capable of executing), *understandable* (the text has to be readable and easy to understand) and *robust* (the content must be interpretable by a wide range of users, thus assistive technologies must be supported).

Figure 3(a) shows such dialogue window with the example that we presented in Tables 1 and 2. This dialogue (in Portuguese) was obtained using the prompts shown in Section 3. In the first part, the chosen answerer was the one built with Watson Assistant [6], and the answer is formed by information organised in a long sentence; the second answer, generated by the language model DEEPAGÉ [5], is short and only names an entity.

Social robot interface. In order to bring corporality to the chatbot and promote a more immersive experience, we implemented a user interface based on a social robot. The robot

[11] https://www.humanrobotics.ai/



Chat about the Blue Amazon

o que é amazonia azul 3:22 PM

Watson (GABRIEL)
A região do mar que pertence ao Brasil é conhecida como Amazônia Azul devido à similaridade em riqueza de recursos vivos, minerais, energéticos e de biodiversidade encontrados nas duas regiões. 3:22 PM

qual é o nome desse oceano 3:22 PM

DEEPAGÉ
O oceano é chamado de Atlântico. 3:22 PM

(a) (b)

Figure 3. (a) Screenshot of the web interface. (b) Robios Go social robot. (Questions in the web interface have been intentionally written without capitalisation and punctuation, in order to demonstrate the Correction step; see Tables 1 and 2.)

used is the Robios Go. The robot, shown in Figure 3(b), has navigation capability, facial expressiveness and speech interaction. A conversational agent through a social robot provides a remarkable opportunity to establish meaningful connections with users, bringing emotional engagement, ease of use, and in some cases, accessibility. Such features potentially enhance engagement, thereby furthering the objective of knowledge sharing.

4.3. Evaluating the Orchestrator

A complex conversational agent such as BLAB has many modules; here we focus on the orchestrator, itself a complex piece with many interacting parts. We have tested a key element of the orchestrator; namely, the ability of the LLM to decide correctly which module to receive a particular question. The test is therefore related to the Answerer Selection challenge discussed in Section 3, and closely related to the Redirection request. Additional testing could be directed to the Sensitivity challenge, but we felt that the abilities of the GPT language model as related to text corrections, the abilities we relied upon in our implementation, have been documented elsewhere [13]. In addition, the evaluation of the Output challenge will require user testing with the complete system, a task we intend to close in the near future.

To test the Redirection request, we manually selected 50 question-answer pairs from the larger set of such pairs that have been used to train modules of BLAB: 15 pairs that should be directed to DEEPAGÉ; 20 pairs that should be directed to a Watson Assistant bot that was also mentioned in the previous section; and 15 pairs that address issues outside the scope of the BLAB. We labelled them according to the answerer module each pair was originally used for. Some examples of questions can be found in Table 3, together with our manually assigned labels.

Using our prompt, we asked GPT 3.5 to choose the most suitable answerer for each question (or detect when the question is out of scope). The results can be summarised as follows:

Question	Label
Qual é o maior rio da Bacia Amazônica? *What is the longest river in the Amazon Basin?*	2
O Parque Nacional de São Joaquim está localizado em qual bioma? *In which biome is located the São Joaquim National Park?*	2
Que tipo de vegetação é encontrada no bioma Cerrado? *Why is the Blue Amazon important?*	1
Qual é o impacto do aquecimento global na Amazônia Azul? *What is the impact of global warming on the Blue Amazon?*	1
Por que os Países Baixos sofreram da doença brasileira? *Why did the Netherlands suffer from the Brazilian disease?*	0
O clima é verde quando fica? *Is the weather green when it stays?*	0

Table 3. Sample of our testing dataset in Portuguese (English translation is shown for presentation purposes; the translation of malformed questions is necessarily difficult). There are three possible labels (our ground truth): '2' refers to DEEPAGÉ; '1' refers to the aforementioned Watson-based answerer; and '0' indicates a question to be out of scope (incorrectly built, about unsupported domains, etc.).

- 14 out of the 15 questions that should be sent to DEEPAGÉ have been correctly redirected;
- out of the 20 questions that were supposed to be answered by the Watson Assistant answerer, 5 have been manually discarded because they may have short answers, and the 13 out of the remaining 15 questions have been correctly redirected;
- 4 out of the 15 out-of-scope questions have received the correct output.

Thus, the success rate was $\frac{27}{30} = 90\%$ for well-formed questions about the domain supported by the answerers. However, the detection of out of scope questions was not as successful, since only a rate of $\frac{4}{15} = 27\%$ was reached. Further experiments are required to verify whether it is possible to improve the accuracy in the latter case without negatively impacting other cases. Also, we intend to evaluate larger sets of question-answer pairs about different domains.

5. Final Remarks

In this paper, we have proposed a novel strategy for those conversational agents that must integrate several question answering modules, each one with specific abilities. Our strategy is to exploit the resources of a support LLM so as to best orchestrate modules. By transforming the text in both ends, by taking into account the features of LLMs, and by selecting the best module at each turn, this strategy makes it possible for conversational agents to handle questions that would otherwise be misinterpreted or answered incorrectly. On the one hand, the use of LLMs within the orchestration mechanism frees one from dealing with many domain specific languages; on the other hand, the prompt engineering process is not at all straightforward and the templates we have developed should be useful for other agents.

By enhancing conversational fluency, particularly through a solution that appropriately handles conversation states and that provides complete and contextualised responses, we envision opportunities for engaging user interactions. Specifically, our approach can be adapted to applications that challenge the user (such as argumentation systems) or even gamified applications (such as closed-world role playing games) that rely on exploring the knowledge offered by multiple modules.

In the future, we would like to enhance our system with the use of LLMs to make small talk in a customised way, without letting it divert the dialogue away from the topics covered by specialised modules. Furthermore, we intend to build question-answering modules for BLAB using up-to-date technology and data, besides improving those that have already been deployed. Whilst we have already integrated BLAB with Rasa and Haystack, other answerers that use those frameworks are yet to be developed.

Acknowledgements

We would like to thank Gabriel Okamoto, Caio Deberaldini, Caio Asai, and Leticia Figueiredo for their previous contribution to this project. 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. R. M. Tavares was supported by the Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação (PIBITI) of CNPq; R. S. Grava was supported by the Programa Institucional de Bolsas de Iniciação Cientifica (PIBIC) of CNPq. P. Pirozelli was supported by the FAPESP grant 2019/26762-0. F. G. Cozman thanks the support of the National Council for Scientific and Technological Development of Brazil (CNPq grant #305753/2022-3). S. M. Peres would like to thank Human Robotics for the partnership and support concerning the use of the Robios platform.

References

[1] Abreu, C.T.: Brazilian Coastal and Marine Protected Areas Importance, Current Status and Recommendations. Ph.D. thesis, Division for Ocean Affairs and the Law of the Sea, Office of Legal Affairs, The United Nations New York (2015)

[2] Agência Nacional do Petróleo (ANP): Encarte de consolidação da produção 2021: Boletim da produção de petróleo e gás natural. https://www.gov.br/anp/pt-br/centrais-de-conteudo/publicacoes/boletins-anp/boletins/arquivos-bmppgn/2021/12-2021-boletim.pdf (2021), online; accessed on 31st May 2022

[3] Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., Do, Q.V., Xu, Y., Fung, P.: A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. arXiv:2302.04023 (2023)

[4] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. arXiv:2005.14165 (2020)

[5] Cação, F.N., José, M.M., Oliveira, A.S., Spindola, S., Costa, A.H.R., Cozman, F.G.: Deepagé: Answering questions in Portuguese about the Brazilian environment. In: Britto, A., Delgado, K.V. (eds.) Intelligent Systems - 10th Brazilian Conference, BRACIS 2021, Virtual Event, November 29 - December 3, 2021, Proceedings, Part

II. Lecture Notes in Computer Science, vol. 13074, pp. 419–433. Springer (2021). https://doi.org/10.1007/978-3-030-91699-2_29

[6] Carlos, G.O.: Desenvolvimento de um chatbot sobre a Amazônia Azul (2021), bachelor's thesis written in Portuguese. Translated title: "Development of a Blue Amazon chatbot".

[7] Chase, H.: LangChain. https://github.com/hwchase17/langchain (2022)

[8] Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to answer open-domain questions. In: Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1870–1879. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/P17-1171

[9] Green, B.F., Wolf, A.K., Chomsky, C.L., Laughery, K.: Baseball: an automatic question-answerer. In: IRE-AIEE-ACM '61 (Western) (1961)

[10] Jurafsky, D., Martin, J.H.: Speech and Language Processing. (draft), 3 edn. (2023)

[11] Khatri, C., Venkatesh, A., Hedayatnia, B., Gabriel, R., Ram, A., Prasad, R.: Alexa prize—state of the art in conversational AI. AI Magazine **39**(3), 40–55 (2018)

[12] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks. In: Advances in Neural Information Processing Systems. vol. 33, pp. 9459–9474. Curran Associates, Inc. (2020)

[13] OpenAI: Introducing ChatGPT. https://openai.com/blog/chatgpt/ (2022)

[14] Ortiz, F.: The Blue Amazon, Brazil's new natural resources frontier. https://www.ipsnews.net/2015/05/the-blue-amazon-brazils-new-natural-resources-frontier/ (May 2015), [Online; accessed on 11th July 2022]

[15] Pirozelli, P., Castro, A.B.R., de Oliveira, A.L.C., Oliveira, A.S., Cação, F.N., Silveira, I.C., Campos, J.G.M., Motheo, L.C., Figueiredo, L.F., Pellicer, L.F.A.O., José, M.A., José, M.M., de M. Ligabue, P., Grava, R.S., Tavares, R.M., Matos, V.B., Sym, Y.V., Costa, A.H.R., Brandão, A.A.F., Mauá, D.D., Cozman, F.G., Peres, S.M.: The BLue Amazon Brain (BLAB): A modular architecture of services about the Brazilian maritime territory. In: IJCAI Workshop: AI Modeling Oceans and Climate Change (AIMOCC 2022). pp. 1–11 (2022)

[16] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language models can teach themselves to use tools. arXiv:2302.04761 (2023)

[17] Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge, London (1969)

[18] Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: HuggingGPT: Solving AI tasks with ChatGPT and its friends in HuggingFace. arXiv preprint arXiv:2303.17580 (2023), https://arxiv.org/abs/2303.17580

[19] W3C World Wide Web Consortium: Web content accessibility guidelines 2.1, level a & level aa success criteria. https://www.w3.org/TR/WCAG21/ (Recommendation 05 June 2018)