HERO-GPT: Zero-Shot Conversational Assistance in Industrial Domains Exploiting Large Language Models

Luca Strano[1], Claudia Bonanno[1], Francesco Ragusa[1,2], Giovanni M. Farinella[1,2,3] and Antonino Furnari[1,2]

[1]*FPV@IPLAB, DMI - University of Catania, Italy*

[2]*Next Vision s.r.l. - Spinoff of the University of Catania, Italy*

[3]*Cognitive Robotics and Social Sensing Laboratory, ICAR-CNR, Palermo, Italy*

Keywords: Virtual Assistants, Visual Question Answering, Large Language Models.

Abstract: We introduce HERO-GPT, a Multi-Modal Virtual Assistant built on a Multi-Agent System designed to swiftly adapt to any procedural context minimizing the need for training on context-specific data. In contrast to traditional approaches to conversational agents, HERO-GPT utilizes a series of dynamically interchangeable documents instead of datasets, hand-written rules, or conversational examples, to provide information on the given scenario. This paper presents the system's capability to adapt to an industrial domain scenario through the integration of a GPT-based Large Language Model and an object detector to support Visual Question Answering. HERO-GPT is capable of offering conversational guidance on various aspects of industrial contexts, including information on Personal Protective Equipment (PPE), machinery, procedures, and best practices. Experiments performed in an industrial laboratory with real users demonstrate HERO-GPT's effectiveness. Results indicate that users clearly prefer the proposed virtual assistant over traditional supporting materials such as paper-based manuals in the considered scenario. Moreover, the performance of the proposed system are shown to be comparable or superior to those of traditional approaches, while requiring little domain-specific data for the setup of the system.

1 INTRODUCTION

AI assistants capable of communicating with humans through the use of Natural Language experienced a surge in popularity during the last decade, revolutionizing the way we engage with technology. Prominent examples include ChatGPT[1], developed by OpenAI, which excels in generating human-like responses across a wide range of topics, Amazon's Alexa[2], a household name virtual assistant embedded in smart devices, as well as Google's Assistant[3] and Apple's Siri[4], both employed in smartphones and other smart devices to provide information, manage schedules and execute tasks, bringing voice-activated assistance to millions of users globally. A virtual assistant able to give assistance to users which have to accomplish specific tasks becomes particularly beneficial in industrial contexts, especially when the users are unfamiliar or only partly familiar with their surroundings (e.g., novel workers). If a worker seeks information about a particular piece of equipment or a specific step of a procedure to be performed, the intelligent assistant should provide a relevant response, allowing the user to seamlessly proceed with their work.

The current approach to the development of a conversational assistant in a given domain involves defining a comprehensive list of potential intents (user's goals), entities (mentioned objects), responses, and conversational paths tailored to a specific context to effectively assist the user with their queries (Bonanno et al., 2023). The majority of well-established frameworks employed in the development of virtual assistants operate based on a similar principle, including notable examples such as the open-source frame-

[1] https://openai.com/chatgpt

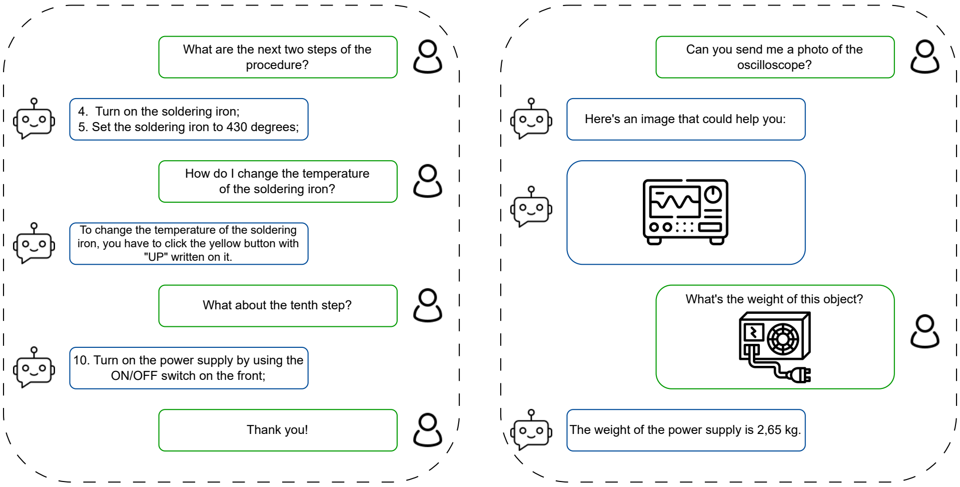
[2] https://developer.amazon.com/alexa

[3] https://developers.google.com/assistant

[4] https://www.apple.com/siri/

Strano, L., Bonanno, C., Ragusa, F., Farinella, G. and Furnari, A.
HERO-GPT: Zero-Shot Conversational Assistance in Industrial Domains Exploiting Large Language Models.
DOI: 10.5220/0012688900003720

In *Proceedings of the 4th International Conference on Image Processing and Vision Engineering (IMPROVE 2024)*, pages 74-82
ISBN: 978-989-758-693-4; ISSN: 2795-4943



What are the next two steps of the procedure?

4. Turn on the soldering iron;
5. Set the soldering iron to 430 degrees;

How do I change the temperature of the soldering iron?

To change the temperature of the soldering iron, you have to click the yellow button with "UP" written on it.

What about the tenth step?

10. Turn on the power supply by using the ON/OFF switch on the front;

Thank you!

Can you send me a photo of the oscilloscope?

Here's an image that could help you:

What's the weight of this object?

The weight of the power supply is 2,65 kg.

Figure 1: Examples of interactions between users and HERO-GPT. Left: users can ask information on procedures and objects through textual interactions. Right: the system also allows for multi-modal interactions, giving information about objects recognized from visual observations and providing images as responses.

work RASA[5] and Amazon's Lex[6]. Following this paradigm, to train a conversational assistant, it is necessary to acquire a domain-specific dataset encompassing examples of intents (for example, to obtain the next step in a procedure, the user may use different expressions such as "go on", "what I should do next?", etc.), entities (the objects relevant in a given industrial context may not be relevant in a different one), domain-specific information (e.g., best practices or instructions on the use of equipment), as well as conversation examples. The collection of such kind of datasets requires domain-specific expert knowledge and is increasingly demanding as the number of possible intents, entities, and responses grows. Furthermore, current approaches prove nearly impossible to generalize to different contexts, as the required data given to the system is intricately linked to the environment considered during the design of the conversational agent, hence requiring a full re-design of the system when a new context is considered.

To tackle the problems tied with current approaches, we present HERO-GPT, a Multi-Modal Virtual Assistant based on a Multi-Agent System[7] able of swiftly adapting to any given context without the need of specific training or a dataset of context-specific intents, entities, responses or conversation examples. Rather than relying on such datasets, HERO-GPT is fed with a series of documents providing the necessary information on the scenario at hand (e.g., a series of digital documents pertaining to the maintenance process of a specific machine). Through the analysis of such documents and the integration of a GPT-based Large Language Model, our system is able to offer conversational guidance to users across several aspects of the considered context. Also, our system exploits an object detector to provide Multi-Modal conversational abilities and give information on objects of interest from visual observations (e.g., "Which PPE should I use with this object?") avoiding language ambiguity, a useful feature in hands-free agents embedded in wearable systems. Figure 1 illustrates the functionalities and interaction flow of HERO-GPT. The performance of the proposed system is evaluated through a user study in an example industrial laboratory where users are tasked to complete given procedures through the help of the conversational agent. Comparisons with traditional approaches (i.e., paper-based manuals) and a conventional implementation of a conversational agent through the manual definition of entities, intents and responses show the potential of the proposed approach, with HERO-GPT being preferred over traditional approaches and performing on-par with conventional implementations requiring a fraction of domain-specific data.

In sum, the contributions of this work are as follows: 1) we propose HERO-GPT, a generic conversational agent able to easily adapt to new contexts through the integration of digital documents describing best practices and technical information on the scenario at hand; 2) we compare the proposed system with traditional supporting materials (paper manuals) and conventional conversational agent implementations in an industrial scenario, highlighting the potential of HERO-GPT.

[5] https://rasa.com

[6] https://aws.amazon.com/lex/

[7] By "Multi-Agent System" we intend a system composed of a multitude of autonomous Language Models capable of interacting with each other, as described here: https://python.langchain.com/docs/modules/agents/.

2 RELATED WORK

Intent Recognition and Conversational Assistants. Intent Recognition refers to the ability of a conversational assistant or an AI system to comprehend the purpose or objective underlying a user's request. Approaches for intent classification include the deployment of a simple CNN on top of a pre-trained word embedding model (Kim, 2014), a joint CNN-RNN framework to facilitate long-term dependency capturing (Hassan and Mahmood, 2018) and a BERT-based model for both intent classification and slot filling (Chen et al., 2019). Intent recognition serves as the foundational building block for every conversational assistant. Indeed, the accurate prediction of the user's underlying intent behind their queries is necessary for guiding every subsequent action the system might undertake. The authors of (Huang et al., 2018) built a crowd-sourced system with automation capabilities such as automated voting for optimal responses, whereas (Cui et al., 2019) proposed a multimodal dialogue system that leverages visual features and the user's preferences expressed during dialogue to assists them in the fashion domain. The authors of (Sreeharsha et al., 2022) built a voice-enabled chatbot on top of the Amazon's Lex service for hotel reservation purposes. The conventional development of conversational assistance typically demands training data tailored to a specific context, which is demanding to acquire and label. Adapting an existing system to a new context generally requires the collection and labeling of new data, an exhaustive training session or a complete re-design of the systems. In contrast, the HERO-GPT framework proposed in this paper does not require training or fine-tuning on utterances gathered specifically for context-specific intents (except for general-purpose intents such as greeting the assistant), which allows for a seamless adaptation to varying contexts by dynamically updating the system's Knowledge Base.

Language Models. Language Models are probabilistic systems capable of predicting the next most suitable token in a sequence, based on the contextual information present within a given text. The latest significant innovations in language models revolve around the concept of attention and exploit the Transformer architecture (Vaswani et al., 2017). BERT (Devlin et al., 2018), LLaMA-2 (Touvron et al., 2023) open foundation models and Google's T5 (Raffel et al., 2020) are examples of such models. Recently, the GPT-3 (Brown et al., 2020) architecture underwent a fine-tuning phase to enhance its alignment with user intent, resulting in improved performances. Notably, this fine-tuning process also led to a significant reduction in the number of model parameters, leading to the InstructGPT model (Ouyang et al., 2022). Our HERO-GPT framework leverages the advanced general purpose language understanding capability of Large Language Models (LLMs) integrating them into a Multi-Agent environment.

Visual Question Answering. Visual Question Answering (VQA) involves the integration of Machine Learning, Computer Vision and Natural Language Processing to comprehend and respond to questions related to visual queries. VQA bridges the gap between visual content and human-like interaction, making it an essential component for modern AI Assistants. Previous approaches include the use of reinforcement learning in a cooperative environment (Das et al., 2017b) and the selection of specific image regions containing answers to the text-based queries (Shih et al., 2016). Research pertaining VQA moved to consider the conversational history other than the user query, leading to the Visual Dialog task (Das et al., 2017a). Recently, VQA and Visual Dialog have been addressed by using novel concepts such as recursive attention for pronoun resolution (Niu et al., 2019) and the deployment of a large-scale variant of a Transformer model (Tan and Bansal, 2019). HERO-GPT offers similar functionalities to VQA, relying on an object detector to extract visual cues from an image provided by the user.

3 APPROACH

This section discusses the details of the proposed system. Figure 2 illustrates a detailed working scheme of the HERO-GPT's Multi-Agent framework, which is comprised of five main modules: 1) Router module, 2) GPTManager module, 3) ObjectDetector Module, 4) ImageManager Module and 5) ProcedureManager Module. Some of these main modules are supported by multiple LLM-based entities to accomplish different Natural Language Understanding sub-tasks.[8] The main modules also rely on secondary components, highlighted with dashed boxes in Figure 2. The system also relies on a Knowledge Base containing documents (e.g., pdf documents of equipment manuals, procedure explanations, or best practices) and images related to the target environments. These documents

[8] Please see the supplementary material available at https://iplab.dmi.unict.it/download/hero_gpt_supplementary.pdf for examples of the prompts used by the different modules.

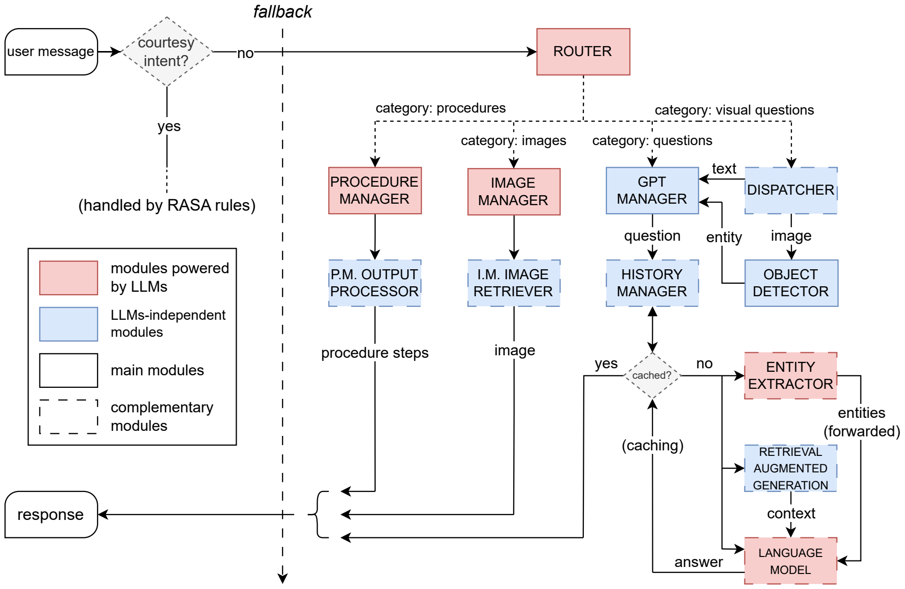


Figure 2: High level overview of HERO-GPT's Multi-Agent system. Red boxes represent LLM-powered modules, whereas blue boxes delineate LLM-independent modules. Solid contour represents main modules, while dashed lines depict complementary modules. The diagram illustrates the five principal modules within the system: 1) The Router Module has the role of choosing the appropriate path to fulfill the request; 2) The GPTManager Module is responsible for managing relationships between complementary modules tasked with Natural Language output generation (bottom right modules on the figure); 3) The ObjectDetector Module is designed to identify entities within images submitted by users; 4) The ImageManager Module retrieves images from the Knowledge Base depending on user input; 5) The ProcedureManager Module has the role of retrieving the desired steps of the initiated procedure. See text for additional details.

undergo a pre-processing stage in which they are broken into smaller units and indexed with vector-based representations obtained through the use of an embedding model. HERO-GPT is built on top of the RASA framework in a way that allows context independence by leveraging the built-in fallback intent. The LangChain framework[9] is employed for every LLM related operation. HERO-GPT is deployed as a Telegram Bot using RASA's channel connector feature. Every module and secondary component is described in greater detail in the next sections.

3.1 Router Module

A set of courtesy intents is defined to familiarize users with the functionalities of the assistant. Courtesy intents, namely "user_greet", "user_start", "user_deny", "user_bot_challenge" and "user_send_image", are standard and remain consistent across different contexts. To recognize these intents, a brief training phase with the standard RASA Natural Language Processing pipeline is required. User utterances categorized with such intents are handled with RASA's rule-based system (e.g., the assistant will greet the user when the "user_greet" intent is recognized). Any other inquiry that doesn't align with these predefined intents is sent to the Router Module. The Router Module leverages the general-purpose language understanding ability of Large Language Models to avoid the need of context-specific intent classification, accurately forwarding the user's query to the appropriate module based on the identified intent category. Intent categories encompass: 1) Procedures (e.g., "What's the next step?"); 2) Images (e.g., "Can you send me an image of the oscilloscope?"); 3) Questions (e.g, "How do I turn on the soldering iron?"); 4) Visual Questions (e.g., "What's this object needed for?). Associated queries are appropriately forwarded to the ProcedureManager, ImageManager, GPTManager and Dispatcher modules respectively.

3.2 ProcedureManager Module

HERO-GPT is capable of outputting specific steps of a selected procedure, contained in the Knowledge Base. A procedure is defined as a sequence of steps required to achieve a particular objective. This concept is expandable across various contexts. For instance, in a culinary setting, a procedure might refer to a cooking recipe; in an industrial setting, a procedure might refer to the repair procedure of a high voltage board. Once the user initiates a procedure in

[9] https://www.langchain.com

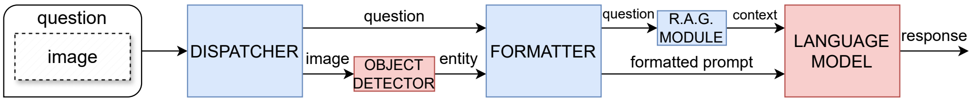


Figure 3: Architecture of the Image-to-Prompt system. The Dispatcher Module divides user input into question and image. The former is forwarded through the Formatter Module, which is part of the GPTManager Module, into the Retrieval Augmented Generation (R.A.G.) Module to retrieve context, while the latter is forwarded to the Object Detector Module, which outputs the class of the closest object to the center as an entity. Finally, context and formatted prompt (e.g. "question about (entity): (question)") get integrated and transmitted to a Language Model, which generates the response.

Natural Language, they have the option to request the previous or next steps, as well as specify a particular step (refer to Figure 1-left). Procedures are sourced from documents inside the Knowledge Base that are marked with the "procedure" keyword. To address the user query, the LLM instance is tasked with generating a JSON object containing command (next, previous or specific) and steps number. For instance, if the user asks "What are the next four steps?", the Language Model should return a JSON object containing the "next" command and the integer 4. This JSON object is subsequently processed by the Procedure-Manager complementary module (named P.M. Output Processor in Figure 2), which reads the desired steps from the procedure loaded in memory and outputs them to the user.

3.3 ImageManager Module

HERO-GPT possesses the capability of forwarding images sourced from the Knowledge Base upon user request. This functionality proves to be especially useful when users are unfamiliar with their environment; indeed, visual information often grants better assistance compared to Natural Language responses. When a user requests a visual output, the LLM instance is prompted to select the most relevant image based on the user's query. Image search relies on filenames for retrieval. Lastly, the ImageRetriever Module retrieves the selected image from the Knowledge Base and forwards it to the user.

3.4 GPTManager Module

The GPTManager Module coordinates the generation of Natural Language responses to user's queries. HERO-GPT's responses are generated through the use of Retrieval Augmented Generation (RAG) (Lewis et al., 2020), which retrieves the essential context required to correctly answer the user's query from the Knowledge Base. To reduce the number of calls to the LLM instance, the GPTManager Module forwards the received question to the HistoryManager, which caches questions and related previously generated responses. If the question is sufficiently similar to an already cached question, the related answer is directly returned to the user. If the question is not cached, it is forwarded to the EntityExtractor, Retrieval Augmented Generation and Language Model Modules. The EntityExtractor Module uses an appropriate prompt to extract key entities from the user input. The RAG Module computes a similarity measure to retrieve the k most similar documents chunks to the query. Subsequently, a prompt is dynamically constructed by incorporating the retrieved document chunks along with the user's query. Lastly, the formatted prompt is forwarded to the LLM instance to generate contextually relevant responses. The user input, along with the extracted entities, associated response and other relevant information is forwarded to the HistoryManager Module, which caches the response and outputs it to the user.

3.5 ObjectDetector Module

When the Router Module detects a visual question (i.e., a question complemented with an image), the whole bundle is sent to the Dispatcher Module, which forwards the textual part to the GPTManager and makes use of the ObjectDetector Module to extract the appropriate entity (i.e., the object's identity) from the image. The Object Detector deployed for this Module consists of a two-stage Object Detector Faster R-CNN (Ren et al., 2015). The ObjectDetector Module extracts the class of the closest object to the center of the input image (the one the user is likely looking at) as an entity and forwards it to the GPTManager Module. Subsequently, the GPTManager Module constructs a prompt that incorporates the received entity along with every other necessary contextual information (see Figure 3). It is noteworthy that, while the object detector may need to be trained on domain-specific images and object classes, given the modular nature of the system, the described module could be implemented with an Open Vocabulary Object Detector or a vision-capable LLM, such as GPT-4V[10].

[10] https://openai.com/research/gpt-4v-system-card

4 EXPERIMENTS AND RESULTS

To evaluate the performances of our system, we conducted a user study with a group of 12 volunteers who were asked to carry out two procedures consisting of about 10 steps each in a mock-up industrial laboratory. The two procedures are randomly assigned to volunteers from a set of four procedures involving activities such as repairing a low voltage board and testing the high voltage one. We performed two sets of tests. The first one aims to assess the usefulness of HERO-GPT when compared to traditional supporting materials, such as paper-based manuals. For these tests, one of the two assigned procedures was performed with the support of HERO-GPT, whereas the other one was performed with the support of a classic paper instruction manual. After testing the system, the participants were asked to fill two questionnaires: the first report to be filled was focused on assessing user's satisfaction degree of the assistant itself, whereas the second one sought feedback on whether the assistant was deemed superior and more user-friendly compared to the classic paper instruction manual. The second set of tests aimed to assess the degree of satisfaction of the user with respect to a Baseline Model implemented following the traditional protocol based on manual definition of intents, entities, and standard answers (see section 4.2). We adopt the same protocol for this tests, asking subjects to perform one of the two activities supported by paper manuals and the other one supported by the Baseline Model.

4.1 Mock-Up Industrial Laboratory

During the testing phase, the context provided to both systems revolves around a mock-up laboratory scenario. The considered laboratory is inspired by a real industrial laboratory (Ragusa et al., 2023), housing various instruments essential for executing a set of procedures.[11] The laboratory is comprised of the following components: 1) three pieces of equipment, namely the oscilloscope, soldering iron, and programmable power supply; 2) two Personal Protective Equipment (PPE) items, gloves and a helmet; 3) two boards, one operating at low voltage and the other at high voltage; 4) a set of tools required to carry out the procedures (e.g., a screwdriver, pliers, electrical screwdriver) and 5) a total of four procedures focusing on the repair and testing process of the laboratory's boards, two for each kind of board. The Knowledge Base of the assistant comprised the following material: 1) a document enumerating and describing the objects within the laboratory; 2) instruction manuals of the oscilloscope, soldering iron and power supply; 3) four procedures encompassing the repair process of low and high voltage boards, as well as the testing procedures for both boards; 4) images for each object present in the laboratory. All of the tests were conducted inside the laboratory.

4.2 Baseline Model

The Baseline Model (Bonanno et al., 2023) considered for the testing phase does not employ Language Models in any of its modules. It was developed through the use of conventional methodologies, defining a dataset of utterances labelled with intent and entities. The Baseline system has equivalent capabilities to the proposed system and is entirely built on top of the RASA framework. The HERO-GPT framework and the Baseline Model share the same object detection model based on the two-stage Object Detector Faster R-CNN.

4.3 Questionnaires

Participants were presented with a total of 21 questions distributed across two questionnaires. Some of these questions were assigned a "satisfaction score" on a scale from 1 to 5, while others had multiple-choice responses. Table 1 and Table 2 present the list of questions included in the two questionnaires, focused on user satisfaction and system-manual comparison respectively. Question 1.13 and 1.14 were not administered during the testing of the Baseline Model, which was tested in a preliminary stage of this research. Note that Question 1.13 regards the Object Detection Module, which was shared for both systems, so we expected the same distribution on participants' satisfaction on both of the proposed assistants, whereas Question 1.14 reflected the preference of the participants between HERO-GPT and the Baseline Model.

4.4 Implementation Details

During the testing phase, GPT-4[12] served as the LLM for the GPTManager Module, while gpt-3.5-turbo-instruct[13] was used for all other modules. Documents were stored in chunks of 400 tokens with an overlap of 40 tokens. The FAISS library (Johnson et al.,

[11] For more information on the laboratory, please refer to the supplementary material: https://iplab.dmi.unict.it/download/hero_gpt_supplementary.pdf

[12] https://openai.com/gpt-4

[13] https://platform.openai.com/docs/models/gpt-3-5

Table 1: Questionnaire 1.

ID	Question
1.1	How satisfied are you overall with the experience in a range from 1 to 5? 1-definitely not satisfied, 5-definitely satisfied
1.2	How natural did you find the interaction with the app in a range from 1 to 5? 1-definitely not natural, 5-definitely natural
1.3	How often did you use the photo sending feature to communicate with the bot? a-never, b-once, c-more than once
1.4	How natural did you find this feature (if you didn't use this feature, you can skip this question) in a range from 1 to 5? 1-definitely not natural, 5-definitely natural
1.5	How helpful do you think the technology demonstrated in this application prototype can be in a range from 1 to 5? 1-definitely not helpful, 5-definitely helpful
1.6	Do you think the technology demonstrated in this prototype can be used in other contexts besides the industrial context? a-yes, b-no
1.7	How often did the system correctly recognize the intent of your questions in a range from 1 to 5? 1-never, 5-each time
1.8	How useful do you think the information received from the application is in a range from 1 to 5? 1-definitely not useful, 5-definitely useful
1.9	How clear do you think the information received from the application is in a range from 1 to 5? 1-definitely not clear, 5-definitely clear
1.10	How satisfied are you with the system response time in a range from 1 to 5? 1-definitely not satisfied, 5-definitely satisfied
1.11	How useful do you think it is for the application to be available on the phone rather than another device (wearable devices, tablets, fixed screens) in a range from 1 to 5? 1-I'd prefer a different device, 5-I prefer a mobile device
1.12	Would you prefer a version with voice dictation? a-yes, b-no
1.13	How often did the system correctly recognize the object in a photo you submitted in a range from 1 to 5? 1-never, 5-every time
1.14	Which version did you prefer the most? a-the previous version, b-today's version, c-no preference.

Table 2: Questionnaire 2.

ID	Question
2.1	Which experience satisfied you the most in a range from 1 to 5? 1-definitely the paper-based manual, 5-definitely the application
2.2	How convenient did you find the use of the paper-based manual in a range from 1 to 5? 1-definitely not convenient, 5-definitely convenient
2.3	How much do you think the technology demonstrated in this application prototype could support you, compared to the use of the paper-based manual in a range from 1 to 5? 1-I found the manual more supportive, 5-I found the application more supportive
2.4	Which tool allowed you to complete the instructions more quickly in a range from 1 to 5? 1-I found the manual as the quickest tool, 5-I found the application as the quickest tool
2.5	How useful do you think the information received from the application is compared to the information obtained through the paper-based manual in a range from 1 to 5? 1-I found the manual instructions more useful, 5-I found the application instructions more useful
2.6	Which tool provided clearer instructions in a range from 1 to 5? 1-I found the manual instructions clearer, 5-I found the application instructions clearer
2.7	Which experience did you prefer overall? a-the use of the application, b-the use of the paper-based manual

2019) was employed as an efficient similarity search approach. Context was provided to the LLM by forwarding a maximum of 3 chunks with an L2 Score lower than 0.42 which resulted most similar to the user's query. For each embedding, the OpenAI text-embedding-ada-002[14] model was applied. To compare the user's query with interactions within the conversation history, cosine similarity was used. The minimum similarity threshold was set at 0.94 (L1 score). The Object Detector Module implemented in HERO-GPT consists of the same one used by the Baseline Model, fine-tuned on 1367 images depicting the laboratory objects.

4.5 Results

Figure 4 illustrates the distribution of answers to questions requiring to express a satisfaction score. As shown in the boxplots, overall satisfaction is higher with our proposed system (Question 1.1 - compare top - baseline - to bottom - HERO-GPT). The naturalness of the system is also superior to the Baseline Model (Question 1.2), but participants expressed a prefer-

[14] https://platform.openai.com/docs/guides/embeddings/embedding-models

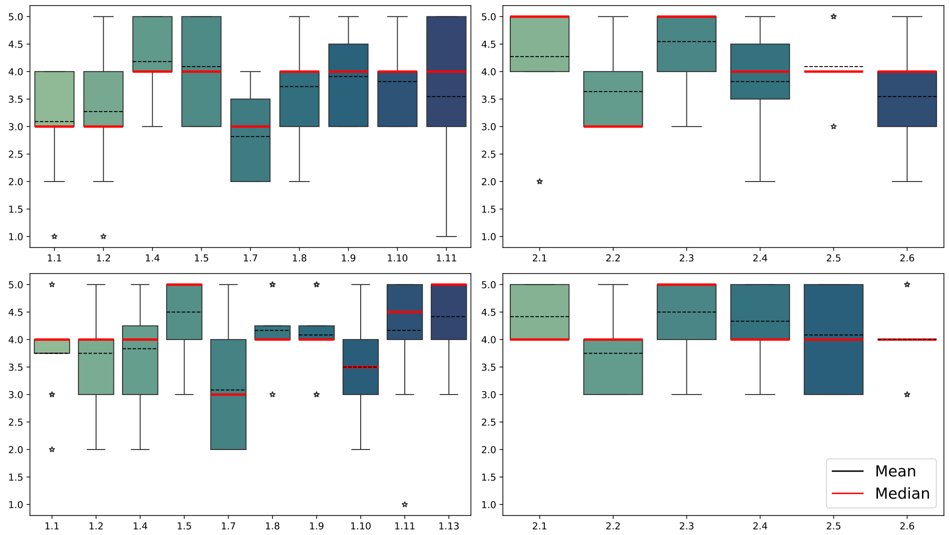


Figure 4: Distribution of satisfaction scores. Plots positioned on the left represent responses from the first questionnaire, whereas plots on the right illustrate responses from the second questionnaire. The upper boxplots correspond to the Baseline Model, while the lower boxplots pertain to HERO-GPT. Please refer to the supplementary material for additional discussion and visualizations.

ence for the answers and the intent recognition mechanism to visual questions implemented in the Baseline Model (Question 1.4). Similarly, HERO-GPT's intent recognition achieved a slightly lower score compared to the Baseline Model (Question 1.7). This result is expected, given that the Baseline Model's intent recognition component is tailored for the considered context. Usefulness and Clarity of answers both obtained higher scores with our proposed system (Questions 1.8 and 1.9). This outcome is attributed to the Language Model's capability to enhance responses by providing additional details on some of the questions proposed by our users. Baseline Model achieved a faster response time compared to HERO-GPT (Question 1.10) due to real-time response generation in the latter. During the testing phase of HERO-GPT, 83.3% of participants repeatedly used the photo-sending feature to communicate with the assistant (Question 1.3) with an accuracy of about 88% (Question 1.13), demonstrating the essential role of multi-modality in modern AI assistants. The entirety of participants believed that our assistant can be used in other contexts (Question 1.6), while only 30% favored the Baseline Model over our proposed system (Question 1.14, with 50% preferring our assistant, and the remaining 20% expressing no preference). Lastly, participants exhibited a preference for the proposed assistants over the provided paper instruction manuals (Questions 2.1 through 2.6) in both tests, with 100% of participants demonstrating a preference for one of the assistants.

5 CONCLUSIONS

This study addressed critical challenges associated with the implementation of virtual assistants, such as the difficulty of expansion and the inability to generalize across different contexts. To mitigate these challenges, we introduced HERO-GPT, a Multi-Modal system based on Large Language Models. To evaluate the system's performance, we performed a series of user tests in an industrial context with 12 volunteers. We compared the system to a classic paper instruction manual support and a Baseline Model developed through the use of conventional methodologies. Experimental results indicate that our participants expressed a clear preference towards our system compared to the other proposed methods. Future development could involve integrating our system with wearable devices and incorporating a speech-to-text model to allow a hands-free experience. Additionally, a comprehensive testing phase could be undertaken to evaluate HERO-GPT's ability of adapting to diverse contexts.

ACKNOWLEDGEMENTS

This research has been supported by Next Vision[15] s.r.l., by Research Program PIAno di inCEntivi per la Ricerca di Ateneo 2020/2022 — Linea di Inter-

[15] https://www.nextvisionlab.it/

vento 3 “Starting Grant” - University of Catania, and by the project Future Artificial Intelligence Research (FAIR) – PNRR MUR Cod. PE0000013 - CUP: E63C22001940006.

REFERENCES

Bonanno, C., Ragusa, F., Furnari, A., and Farinella, G. M. (2023). Hero: A multi-modal approach on mobile devices for visual-aware conversational assistance in industrial domains. In *International Conference on Image Analysis and Processing*, pages 424–436. Springer.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chen, Q., Zhuo, Z., and Wang, W. (2019). Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Cui, C., Wang, W., Song, X., Huang, M., Xu, X.-S., and Nie, L. (2019). User attention-guided multimodal dialog systems. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 445–454.

Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M., Parikh, D., and Batra, D. (2017a). Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Das, A., Kottur, S., Moura, J. M., Lee, S., and Batra, D. (2017b). Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2951–2960.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hassan, A. and Mahmood, A. (2018). Convolutional recurrent deep learning model for sentence classification. *Ieee Access*, 6:13949–13957.

Huang, T.-H., Chang, J. C., and Bigham, J. P. (2018). Evorus: A crowd-powered conversational assistant built to automate itself over time. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–13.

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Niu, Y., Zhang, H., Zhang, M., Zhang, J., Lu, Z., and Wen, J.-R. (2019). Recursive visual attention in visual dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6679–6688.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Ragusa, F., Furnari, A., Lopes, A., Moltisanti, M., Ragusa, E., Samarotto, M., Santo, L., Picone, N., Scarso, L., and Farinella, G. M. (2023). Enigma: Egocentric navigator for industrial guidance, monitoring and anticipation. In *VISIGRAPP (4: VISAPP)*, pages 695–702.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Shih, K. J., Singh, S., and Hoiem, D. (2016). Where to look: Focus regions for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4613–4621.

Sreeharsha, A., Kesapragada, S. M., and Chalamalasetty, S. P. (2022). Building chatbot using amazon lex and integrating with a chat application. *Interantional Journal of Scientific Research in Engineering and Management*, 6(04):1–6.

Tan, H. and Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.