LAMIA: An LLM Approach for Task-Oriented Dialogue Systems in Industry 5.0

Cristina Fernández, Izaskun Fernández, Cristina Aceta

TEKNIKER, Basque Research and Technology Alliance (BRTA),
C/ Iñaki Goenaga 5, 20600 Eibar, Spain
Correspondence: cristina.fernandez@tekniker.es

Abstract

Human-Machine Interaction (HMI) plays an important role in Industry 5.0, improving worker well-being by automating repetitive tasks and enhancing seamless collaboration between humans and intelligent systems. In this context, Task-Oriented Dialogue (TOD) systems are a commonly used approach to enable natural communication in these settings, traditionally developed using rule-based approaches. However, the revolution of Large Language Models (LLMs) is changing how dialogue systems are being developed without the necessity of relying on tedious and rigid handcrafted rules. Despite their popularity, their application in industrial contexts remains underexplored, necessitating a solution to challenges such as hallucinations, lack of domain-specific data, high training costs, and limited adaptability. In order to explore the contribution of LLMs in the industry field, this work presents LAMIA, a task-oriented dialogue system for industrial scenarios that leverages LLMs through prompt tuning. This system has been adapted and evaluated for a bin-picking use case, using GPT-3.5 Turbo, showing to be an intuitive method for new use cases in Industry 5.0.

1 Introduction

Industry 5.0 focuses on human workers and their well-being at the centre of the productive process. In this context, Human-Machine Interaction (HMI) interfaces are an important asset that allow communication between humans and machines (Pizoń and Gola, 2023). This simpler way of interaction, by allowing, for example, the automation of repetitive tasks, improves task efficiency and user experience (Sharma et al., 2023), and allows workers to focus on more creative tasks (Rane, 2023). In this setting, Large Language Models (LLMs) have emerged as powerful tools, enabling more intuitive interactions via applications like virtual assistants and conversational agents, making technology more accessible to a wider audience.

LLMs have significant potential in Industry 5.0, particularly through their role in Task-Oriented Dialogue (TOD) systems, which enable a natural HMI aimed at facilitating problem-solving tasks within specific domains (Ni et al., 2023). However, the use of LLMs in industrial scenarios is not as widespread as in other fields, as these models still face challenges, relevant in those scenarios that admit little or no margin of error, such as production processes. These limitations are hallucinations[1], lack of domain-specific data, and the difficulty and high costs associated with training for new applications.

To address these limitations, this article explores the contribution and role of LLMs in the development of TOD systems in industrial environments. To do this, the use of prompt tuning is explored, as it allows models to be adapted using strategies such as few-shot learning or step-by-step thinking (Ye et al., 2023; Microsoft, 2024) by teaching the model how to behave with prompts in natural language.

The research has led to the development of LAMIA (Large lAnguage Models for Industrial Assistance), a TOD system designed for industrial environments. Optimized via prompt tuning, LAMIA reduces hallucinations and the need for large amounts of data, enhances adaptability, and mitigates high computational costs. Having been implemented with the LLM GPT-3.5 Turbo, LAMIA presents a cost-effective solution for seamless human-machine interaction in Industry 5.0.

[1] Hallucinations in LLMs occur when the model "produces outputs that deviate from users' intent, exhibit internal inconsistencies, or misalign with the factual knowledge, making the deployment of LLMs potentially risky in a wide range of applications" (Liu et al., 2024).

Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology, pages 205–214
May 27–30, 2025. ©2025 Association for Computational Linguistics

2 Related work

2.1 Task-Oriented Dialogue Systems

Task-Oriented Dialogue (TOD) systems are designed for task completion in specific domains, such as ticket booking or table reservation, unlike open-domain systems, which are used for casual conversation (Ni et al., 2023). In Industry 5.0, TOD systems play a key role in Human-Machine Interaction (HMI), automating simple tasks to reduce the cognitive load on operators through natural language communication (Aceta et al., 2022).

According to the designs used for the development of TOD systems, pipelines typically follow either a modular approach or an end-to-end approach. The modular approach consists of separate components for Natural Language Understanding (NLU), Dialogue State Tracking (DST), Dialogue Policy Learning (DP), and Natural Language Generation (NLG) (Li et al., 2021). In contrast, the end-to-end approach, introduced by Wen et al. (2016), creates a trainable end-to-end model that still connects in a modularized way, but does not directly modularize the user goal.

Over the years, the main strategies used to develop these systems were rule-based methods or the use of annotated data (Sekulić et al., 2024). However, both present challenges in Industry 5.0, as rule-based methods require extensive manual adaptation, while data-driven approaches suffer from a lack of domain-specific data in industrial settings (Li et al., 2022) and high computational demands.

2.1.1 Large Language Models in TOD Systems

The state-of-the-art technology that has emerged as a useful tool for a wide variety of applications in the NLP field is Large Language Models (LLMs). LLMs are advanced AI models, often based on Transformer architectures, that can understand and generate human language by being trained on vast amounts of text data (Ozdemir, 2023).

In TOD systems, various LLMs have been employed, such as Tk-Instruct-11B, Alpaca-LoRa-7B, BART and GPT-3.5 (Hudeček and Dusek, 2023; Marselino Andreas et al., 2022; Li et al., 2022; Hu et al., 2024). Fine-tuning methods such as LoRA (Low Rank Adaptation) have been widely used to adapt these models by modifying only a few parameters for specific tasks (Marselino Andreas et al., 2022; Li et al., 2022). Reinforcement Learning from Human Feedback (RLHF) has also shown effectiveness in tuning models based on human input (Ouyang et al., 2022). However, both approaches are data-driven and memory-intensive, requiring significant computational and data resources for training, as gradients and optimizer states for all parameters must be stored (Liu et al., 2022).

2.2 Prompt Tuning

To address the challenges of rule-based methods and the scarcity of annotated data, the LLM prompt tuning strategy has gained popularity. Authors like Cao (2023); Hudeček and Dusek (2023) have applied this approach in TOD systems, which avoids retraining by freezing model parameters and using natural language prompts for adaptability (Liu et al., 2022).

Prompt tuning has demonstrated performance comparable to fine-tuning for large models in simple tasks (Liu et al., 2022; Lester et al., 2021; Kim et al., 2021; Liu et al., 2021), and has outperformed in cross-lingual tasks (Zhao and Schütze, 2021). Furthermore, Addlesee (2024) highlights that prompt tuning improves textual grounding and accuracy, reducing hallucinations in models.

Achieving optimal results with prompt tuning requires effective *prompt engineering* (Zhao and Schütze, 2021), which involves crafting prompts to communicate tasks clearly and iterating based on model responses (Ye et al., 2023). In this line, various strategies have been suggested, such as think step by step, few-shot learning, chain of thought, and in-context learning, among others, to improve performance (Microsoft, 2024).

2.2.1 Prompt Tuning in TOD Systems

Some studies have explored optimizing LLMs TOD systems using prompt tuning. Hudeček and Dusek (2023) developed a modular TOD system, finding that while LLMs excelled in tasks such as domain detection and response generation, they struggle with state-tracking. Similarly, Cao (2023) created DiagGPT, a multi-agent TOD system with strong conversational abilities but less stability than rule-based models. Gupta et al. (2022) introduced InstructDial, which outperformed the base models in unseen dialogue tasks.

The results depended on prompt complexity, with detailed instructions and in-context examples often enhancing performance (Hudeček and Dusek, 2023; Gupta et al., 2022). In terms of the pipeline used, the most popular approach is a modular system in which different dialogue tasks are

Dialogue turn	User message	Agent response	External database
1st turn	“Pon los magenta” / *“Put the magenta ones”*	“¿Cuál es la caja en la que debes colocar el cartucho de tinta magenta?” / *“Which box should I put the magenta ink cartridge in?”*	{_id: Object {. . . }, element_cartridge: "magenta", type: “color”}
2nd turn	“Caja 1” / *“Box 1”*	“¿Quieres colocar el cartucho de tinta magenta en la caja 1?” / *“Do you want to place the magenta ink cartridge in box 1?”*	{_id: Object {. . . }, element: "1", type: “box”}
3rd turn	“Sí” / *“Yes”*	“¡Entendido! Comienzo a trabajar en eso de inmediato. Estoy aquí para ayudarte en lo que necesites.” / *“I understand! I’ll start working on it right away. I’m here to help you with whatever you need.”*	

Table 1: Example of human-machine dialogue of LAMIA in a bin-picking robot.

handled sequentially (Hudeček and Dusek, 2023; Cao, 2023).

All in all, these studies using prompt tuning for TOD systems have not been applied to industrial applications. For this reason, this research focuses on exploring the role of LLMs in TOD systems for industry, using prompt tuning for adaptable systems with low computational costs, minimal data requirements, and reduced hallucinations, crucial for manufacturing environments.

3 Approach and implementation

To address the challenges in building TOD systems for Industry 5.0, we have developed LAMIA, a TOD system for industry implemented through prompt tuning. Among its capabilities, LAMIA allows operators to maintain natural language conversations and send the final task to the target system.

The research was conducted at Tekniker, a technology research centre in the Basque Country, where we had access to KIDE4I (Aceta et al., 2022). KIDE4I is a rule-based TOD system for Industry 5.0 composed of four modules (Key Element Extraction, Polarity Interpreter, Semantic Repository and Dialogue Manager) which rely on syntactic and morphological parsers, and ontologies for storing domain knowledge and managing the dialogue process. More specifically, its use case of a bin-picking robot and its evaluation framework served as our reference to assess the performance of LAMIA in a real-world scenario.

3.1 Dialogue structure for LAMIA’s bin-picking use case

In the bin-picking robot use case in Aceta et al. (2022), the robot handles ink cartridges, identifying their colour or brand, and sorts them into two containers based on operator instructions. For that goal, the dialogue system supports multi-turn interactions in Spanish, and it is capable of receiving instructions in natural language, asking clarifying questions, and sending structured information to the target robot to execute actions, such as relocation. Communication includes both voice commands and gestures —which have to be accompanied by an adverb of place or a demonstrative pronoun, also known as *pointers*—. The system uses predefined world knowledge, including cartridge colours, brands, and container identifiers. Table 1 shows an example dialogue from LAMIA. In addition, Example 1 illustrates the structured output sent to the robot.

Example 1: Dialogue system’s structured output.

- {“task”: {“amount”: 0, “pointer”: 0, “action”: “PICKING”, “destination”: “1”, “colour”: “magenta”, “trademark”: “”}}

3.2 TOD system design

LAMIA’s pipeline has been built iteratively to optimize performance and adaptability. In other words, its creation was based on various rounds to determine which strategies worked best. As seen in Section 2.1, there are currently two strategies applicable to the implementation of TOD systems: *end-to-end* and *modular*. Therefore, for the construction of LAMIA, both approaches have been explored to obtain the final pipeline with the best performance.

3.2.1 End-to-end strategy

In an initial approach, we attempted to build an end-to-end system using a single call to GPT-3.5 Turbo. The goal was to create a prompt that instructed the LLM to handle multiple tasks: understanding user input, detecting key elements (colours, brands, and box numbers), verifying real-world knowledge, retrieving elements from prior interactions, generating natural language responses, and creating a JSON output for the target robot.

For doing this, we used prompt techniques, such as clear instructions, context, and few-shot learning, providing detailed task descriptions and example outputs. However, the prompt was too complex, causing the LLM to miss some instructions and produce hallucinations after several tests. These limitations have previously been demonstrated (Lester et al., 2021; Kim et al., 2021; Liu et al., 2021), showing that prompt tuning performs better on simple tasks. Specifically, the JSON output was frequently incorrect, with inconsistent keys and values, leading to errors in the robot's task execution. Due to these issues, this approach was discarded, as the LLM's hallucinations posed too much risk for a reliable performance.

3.2.2 LAMIA's design: A modular strategy

After identifying the limitations of the end-to-end approach, we explored the modular pipeline, which breaks down tasks into simpler components. Based on the pipelines of Ni et al. (2023) and Aceta et al. (2022), LAMIA's architecture is composed of seven modules that perform different NLP tasks that work sequentially (see Figure 1). After a few tests, the same as those conducted to discard the end-to-end approach, it was observed that only those modules handling natural language input or output benefit from LLMs, as they performed poorly with JSON-based tasks. Thus, LAMIA's modules are the following:

1. **Polarity Interpreter**: Performs two tasks: content detection and polarity detection. The first detects whether the input has semantic content or is just an affirmation/negation. Its output conditions the pipeline that the input will follow, as depicted in Figure 1. The second task, polarity detection, classifies the input without semantic content as positive or negative. Both tasks imply a call to the LLM with an instruction prompt.

2. **Natural Language Understanding / Key Element Extraction (NLU)**: Extracts the key elements (e.g., cartridge type, box) from the inputs with semantic content, using an LLM.

3. **Database Query**: Verifies the extracted values against the database (in this case, MongoDB). This module does not use an LLM.

4. **Dialogue State Tracking (DST)**: Compares current and previous interactions to track the dialogue status. This module also does not require an LLM.

5. **Natural Language Generation (NLG)**: Converts actions into the corresponding natural language response, using an LLM.

6. **JSON Manager**: Converts the instruction into the final format required by the robot, without using an LLM.

7. **Dialogue Policy / Orchestrator**: Manages actions based on interactions, keeping or discarding the JSON history as needed. Without an LLM.

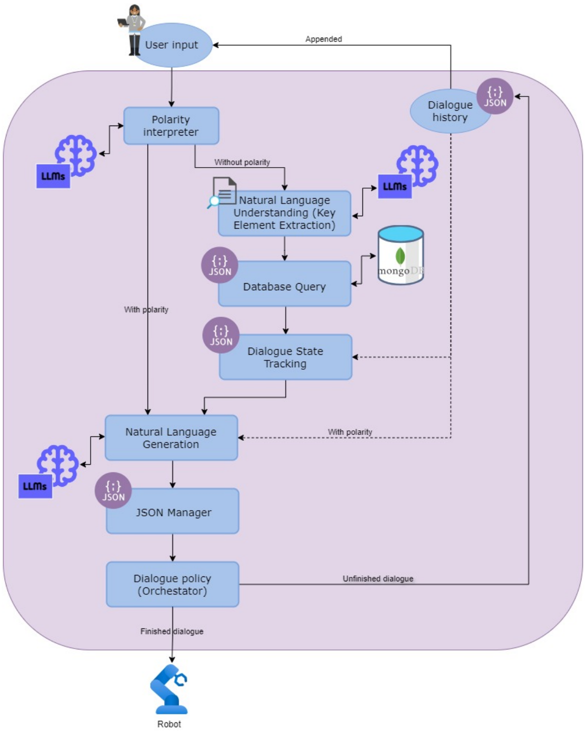

Figure 1: LAMIA's pipeline.

This modular approach proved to be more effective than the end-to-end strategy as, being simpler tasks, it avoided errors in JSON structures and ensured that each task was completed successfully. This final pipeline was used to build LAMIA.

3.3 LAMIA's implementation

For the construction of the pipeline and its components, an iterative implementation approach has also been followed. Consequently, in this section, we will introduce the selected LLM, the definitive prompt strategies, and tools that were utilized, including different libraries, the database, etc.

3.3.1 Selected Large Language Model

The implementation of LAMIA's modules has been made using the LLM GPT-3.5 Turbo, a proprietary model. Developed by OpenAI, GPT-3.5 Turbo is acclaimed for its customization capabilities and strong performance (Peng et al., 2023). The different models and versions of GPT have previously been used for optimization in TOD systems by authors such as Marselino Andreas et al. (2022); Cao (2023) with good performance. We deployed it using Azure OpenAI Studio[2].

3.3.2 Prompt strategies

The LAMIA system uses prompt tuning to adapt LLMs for specific tasks, allowing easier modification and lower computational costs since retraining is unnecessary. As discussed in Section 2.2, the effectiveness of LLM is highly dependent on iterative prompt design, clear strategies, and avoiding ambiguities.

Through various tests, the most effective strategies were the following:

- Few-shot learning: This strategy is based on giving examples of the output. An example of its use is present in the prompt for the NLG module which contains "For example: 'Which ink cartridge do you prefer, black or yellow?', 'I didn't understand you, tell me what the task is'", etc.

- Specifying output structure: Mainly used in those modules that required a structured output, such as JSON —i.e., Polarity Interpreter and NLU. For example, the NLU prompt included: "Output must be in JSON format: 'element cartridge': ", 'element box': ", 'element pointer': ''".

- Providing context: To indicate its function and the type of input the LLM will receive, the three LLM-based modules contained this strategy. For example, the prompt for the NLG included: "Context: You are a virtual assistant programmed to start every interaction by asking the user to specify exactly what he/she wants to do".

- Clear and repeated instructions: The use of instructions with minimal ambiguities as possible, and their reiteration at the end. For example, in the use of clear instructions, the NLU prompt included at the beginning: "Your task is to analyse the input provided by the user to identify and extract specific information related to cartridges (e.g., colour or brand), boxes (e.g., location in number) or pointers (e.g., adverb of place or demonstrative pronoun) [. . .]". Regarding repeated instructions, the NLG module prompt reinforced the idea of "return only the question" at the end after being already mentioned.

- Using syntax in prompts: All prompts used phrases that indicate the information that the LLM had to follow. Some examples are: "Context:", "For example:", "User message:", etc.

In addition, a temperature setting of 0 was used to ensure minimal randomness and high control in the outputs, making the system suitable for industrial use with limited variety of responses but reliable task completion.

3.3.3 Selected tools

To implement LAMIA's modules that are composed by an LLM call (Polarity Interpreter, NLU and NLG), we have used the Langchain[3] library. This library has allowed us to initialize the LLMs APIs, create chains to concatenate inputs and outputs, and format the prompts. Furthermore, this library has also been used in the Database Query module to connect the system with the MongoDB database.

4 Evaluation

To evaluate LAMIA, we utilized the KIDE4I's evaluation framework from Aceta et al. (2022). The LAMIA system has been evaluated by comparing its performance with the KIDE4I standard to assess whether LAMIA offers improvements over traditional rule-based systems.

The evaluations consist of two key components:

- Dialogue: This aspect takes into account the dialogue as a whole by assessing three aspects:

[2] https://oai.azure.com/portal

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

- Dialogue completion rate: Whether the dialogue has been *fully completed*, *partially completed* —the user had to rephrase the instruction— or *not completed*.
- Dialogue completion steps: Number of turns it took to complete the dialogue.
- Error analysis: Cases where the dialogue has not been completed due to a specific error.

- Interaction: This measures the system's response time for each interaction.

Furthermore, LAMIA LLM-based modules (NLU, Polarity Interpreter, and NLG) have also been evaluated against the corresponding modules in KIDE4I. The modules of both systems feature similar functionalities. The NLU module in LAMIA aligns with KIDE4I's Key Element Extraction (KEE) module, and both incorporate a Polarity Interpreter module. However, the NLG module in LAMIA, which is responsible for text generation, does not have a direct counterpart in KIDE4I, but this module has also been analysed, as it is developed with an LLM. LAMIA's adaptability and economic costs have also been assessed.

To follow these evaluations, we have used the same dialogue battery used to assess the KIDE4I system in Aceta et al. (2022)'s work, composed of 75 dialogues.

4.1 LAMIA vs KIDE4I results

For the LAMIA assessment, we compared its performance with the rule-based system KIDE4I (Aceta et al., 2022). In addition, LAMIA modules developed with LLMs have also been examined to evaluate the actual performance of their use in these systems.

4.1.1 General comparison

Focusing on the evaluation of the whole system, both systems had similar dialogue completion rates, but LAMIA performed better in partially completed and uncompleted dialogues, with fewer uncompleted cases (see Table 2). Moreover, both showed similar performance in terms of the number of steps needed to complete dialogues, with LAMIA having a slightly lower average and maximum number of steps (see Table 3).

KIDE4I showed better response times overall, with an average response time of 0.74 seconds vs LAMIA's 1.26 seconds (see Table 4). More specifically, LAMIA's response time is influenced by the complexity of LLM calls, which are not present in a rule-based system like KIDE4I. In Table 5, it can be seen that LLM-based modules require more time to respond. However, the average response time of LAMIA is comparable to other use case of KIDE4I (KIDE4Guide) with 1.25 s. For this case, Aceta et al. (2022) affirm that it is still a fast time, which does not affect the user experience negatively.

	LAMIA		KIDE4I	
	%	#	%	#
Fully completed	90.66	68	82.66	62
Partially completed	6.66	5	0	0
Not completed	2.66	2	17.33	13
Total dialogues	75		75	

Table 2: Dialogue completion rate for LAMIA and KIDE4I, with their percentages (%) and absolute numbers (#).

	LAMIA	KIDE4I
Average	2.4109	2.5947
Max	5	6
Min	2	2

Table 3: Average, maximum, and minimum dialogue completion steps in LAMIA and KIDE4I.

A further level of assessment is necessary to analyse the errors that have led to the dialogues in both systems being uncompleted or partially completed. The errors reported in the LAMIA system are two:

- Lack of synonymous key elements in the database.
- Bad element detection by the NLU in one case, which resulted in a partially completed dialogue.

Concerning KIDE4I's errors, we have to consider that the modules of this system are not the same as the ones in LAMIA, but they are comparable. The errors reported by Aceta et al. (2022) are as follows:

[4] The Dialogue Policy time is not represented, since it acts as an orchestrator and its response time is the same as the total interaction time.

	LAMIA	KIDE4I
Average	1.2615 s	0.7493 s
Max	1.9504 s	5.3110 s
Min	0.6885 s	0.1100 s

Table 4: Average, maximum, and minimum response time in LAMIA and KIDE4I.

	Average	Max	Min
Polarity Interpreter	0.3843 s	0.612 s	0.1954 s
NLU	0.5231 s	0.74 s	0.3341 s
Database Query	0.043 s	0.078 s	0.0416 s
DST	0.0007 s	0.0012 s	0.0005 s
NLG	0.6337 s	1.0979 s	0.4607 s
JSON Manager	0.0008 s	0.0005 s	0.0015 s

Table 5: Average, maximum, and minimum response time per module in LAMIA.[4]

- Erroneous analysis of structures or lemmas in the syntactic analysis.
- Out-of-scope structures in the definitions and/or rules.

In summary, LAMIA showed better performance in terms of dialogue completion, with fewer uncompleted dialogues compared to KIDE4I. Although KIDE4I had faster response times, LAMIA still allowed for fluent conversations, despite the longer LLM processing times.

4.1.2 Modular comparison

To complete the evaluation of the contribution of LLMs in LAMIA, we analysed the performance of key modules —NLU, Polarity Interpreter, and NLG— by comparing them with their counterparts in the rule-based KIDE4I system, where applicable. For those tasks that are not comparable with any KIDE4I's component, such as Polarity Interpreter content detection or the NLG, we have also extracted their ratios without making a comparison.

The Polarity Interpreter in both systems showed similar performance in classifying polarity, with no errors in LAMIA and only one out-of-scope error in KIDE4I (see Table 6). For content detection, LAMIA performed almost perfectly, with only one classification error due to a misspelled word (see Table 7).

LAMIA's NLU module outperformed KIDE4I's KEE module, with a higher rate of fully detected elements (96.63% vs 64.66%) and fewer partial —not all the elements of the input were detected— or wrong/null detections (see Table 8). This improvement minimized confusion and reduced the number of dialogue turns required.

	LAMIA (PI-Polarity Interpreter)		KIDE4I (PI)	
	%	#	%	#
Good classification	100	77	98.73	78
Wrong classification	0	0	0	0
Out-of-scope errors	-	-	1.26	1
Total	77		79	

Table 6: Polarity Interpreter (polarity classifier task) performance in LAMIA and KIDE4I with the percentages (%) and absolute numbers (#).

	LAMIA (PI-Content classifier)	
	%	#
Good classification	99.48	195
Wrong classification	0	0
Out-of-scope errors	0.51	1
Total	196	

Table 7: Polarity Interpreter (content classifier task) performance in LAMIA with the percentages (%) and absolute numbers (#).

The NLG module in LAMIA participated a total of 196 times. We analysed these responses by categorizing them into well- and wrong-generated responses. This assessment ensured that the interaction was appropriate to the dialogue's state, contained accurate key elements, and adhered to grammatical norms. The results revealed that 100% of the responses were well generated, without errors and hallucinations in the use of key elements and suitable for the dialogue states.

Overall, the modular analysis showed that the Polarity Interpreter performed equally well in both systems. However, LAMIA, using GPT-3.5 Turbo, significantly outperformed KIDE4I in NLU, with better key element detection and fewer errors. Additionally, the NLG module in LAMIA performed flawlessly, showing the capabilities of LLMs when generating natural language responses.

	LAMIA (NLU)		KIDE4I (NLU)	
	%	#	%	#
Fully detected	96.63	115	64.66	86
Partially detected	1.68	2	17.29	23
Wrong/null detection	0.84	1	12.78	17
Out-of-scope errors	0.84	1	5.26	7
Total	119		133	

Table 8: NLU performance in LAMIA and KIDE4I with the percentages (%) and absolute numbers (#).

4.1.3 Other evaluated aspects

Other aspects to take into account when evaluating this kind of system for Industry 5.0 are the applicability of the pipeline and its cost to new use cases. The target system must be functional in different industrial use cases and easy to build to reduce costs and development time. The following are the changes that should be made to adapt the system to a new use case:

- Create new records in the database or connect the system to an existing one.
- Change of dictionary names within the pipeline.
- Slight prompt changes to adapt to the new task.

Another key consideration is the cost of using LLMs. In this work, LAMIA was deployed using GPT-3.5 Turbo, a proprietary model whose use requires payment. The infrastructure utilized to deploy this model has been Azure OpenAI Studio, which operates on a pay-as-you-go pricing model. The specific setup used in this work —GPT-3.5-Turbo-0613 with a 16k context window— costs €0.0015 per 1000 input tokens and €0.0019 per 1000 output tokens (in Central Sweden). As an example of the total cost, the reproduction of the dialogues used to assess LAMIA with this setup had a total cost of €0.85, which is not high, considering that they were 75 dialogues with an average of 2.41 interactions (enough to complete the target tasks). Although not free, it offers good performance without being expensive, making it a viable option in real-world manufacturing contexts. However, companies must evaluate whether these costs are justified based on their production needs and expected gains in productivity.

5 Conclusions

This study offers new insights into the application of Large Language Models (LLMs) in the development of applications for Industry 5.0. The research is focused on exploring the contribution of these models in applications for Human-Machine Interaction (HMI) such as Task-Oriented Dialogue (TOD) systems. With this objective, this article presents LAMIA, a prompt-optimized LLM-based TOD system for Industry 5.0. This system also searches for solving the most criticized limitations present in traditional and LLM-based applications, such as difficult adaptability to new use cases and domains due to handcrafted rules, LLMs' hallucinations, lack of domain-specific data, and the difficulty and high costs associated with re-training these models for new use cases and domains.

For this, LAMIA leverages prompt tuning strategies, which have shown significant advantages in intuitive development, adaptability to new domains, and use cases with low computational costs. In addition, the system has performed better than traditional systems, demonstrating that it is efficient for use in a real industrial setting by being able to complete the task and maintain a smooth dialogue.

Moreover, this study also reveals the specific contribution of LLMs and the prompt tuning strategy in this kind of system. The end-to-end approach test showed that prompt tuning does not perform as well with complex tasks or several tasks at once, as already demonstrated by Liu et al. (2022), which is the reason for using a modular pipeline. However, preliminary experiments in these modules also showed that the effectiveness of LLM varies by task, performing almost perfect for NLP tasks, such as generation, classification, or slot filling, and not being the most suitable option for those that manage structured formats. Therefore, the contribution and adaptation of LLMs in TOD systems must be consistent and adapted to the purposes of these models, which are natural language understanding, processing, and generation.

Future research should focus on addressing the main limitation of LAMIA, the lack of synonymity in the database, which is the main cause for the presence of uncompleted dialogues or the increase of dialogue completion steps. The integration of ontologies as a database could help mitigate this issue by expanding the range of recognized terms, and, therefore, improving the fluidity of the conversation with the system, reducing overall times, and thus reducing costs. Furthermore, further investigation of the adaptability of the system is needed, as the synonymity problem could not be present in other use cases, with a real assessment of its scalability looking for out-of-scope problems. An interface deployment for industrial uses is also necessary, with a user study to assess operator's experience. Moreover, the use of a proprietary LLM can be a handicap for most industries, as it involves a cost. For this reason, the implementation of LAMIA with an open-source model must also be considered and evaluated. Finally, ethical considerations, such as data privacy and transparency, should also be addressed as the system moves toward production use.

Acknowledgments

This project has received funding from the Department of Economic Development and Infrastructure of the Basque Government under grant number KK-2024/00064 (IKUN).

References

Cristina Aceta, Izaskun Fernández, and Aitor Soroa. 2022. KIDE4I: A generic semantics-based task-oriented dialogue system for human-machine interaction in industry 5.0. *Applied Sciences*, 12(3):1192.

Angus Addlesee. 2024. Grounding LLMs to In-prompt Instructions: Reducing Hallucinations Caused by Static Pre-training Knowledge. In *Proceedings of Safety4ConvAI: The Third Workshop on Safety for Conversational AI@ LREC-COLING 2024*, pages 1–7.

Lang Cao. 2023. DiagGPT: An LLM-based chatbot with automatic topic management for task-oriented dialogue. *arXiv preprint arXiv:2308.08043*.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey P Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. *arXiv preprint arXiv:2205.12673*.

Songbo Hu, Xiaobin Wang, Zhangdie Yuan, Anna Korhonen, and Ivan Vulić. 2024. DIALIGHT: Lightweight Multilingual Development and Evaluation of Task-Oriented Dialogue Systems with Large Language Models. *arXiv preprint arXiv:2401.02208*.

Vojtěch Hudeček and Ondrej Dusek. 2023. Are Large Language Models All You Need for Task-Oriented Dialogue? In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.

Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Jeon Dong Hyeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, Heungsub Lee, Minyoung Jeong, Sungjae Lee, Minsub Kim, Suk Hyun Ko, Seokhun Kim, Taeyong Park, Jinuk Kim, Soyoung Kang, Na-Hyeon Ryu, Kang Min Yoo, Minsuk Chang, Soobin Suh, Sookyo In, Jinseong Park, Kyungduk Kim, Hiun Kim, Jisu Jeong, Yong Goo Yeo, Donghoon Ham, Dongju Park, Min Young Lee, Jaewook Kang, Inho Kang, Jung-Woo Ha, Woomyoung Park, and Nako Sung. 2021. What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3405–3424, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Chen Li, Xiaochun Zhang, Dimitrios Chrysostomou, and Hongji Yang. 2022. ToD4IR: A humanised task-oriented dialogue system for industrial robots. *IEEE Access*, 10:91631–91649.

Zekun Li, Hong Wang, Alon Albalak, Yingrui Yang, Jing Qian, Shiyang Li, and Xifeng Yan. 2021. Making something out of nothing: Building robust task-oriented dialogue systems from scratch. *Proceedings of Alexa Prize TaskBot*.

Fang Liu, Yang Liu, Lin Shi, Houkun Huang, Ruifeng Wang, Zhen Yang, Li Zhang, Zhongqi Li, and Yuchi Ma. 2024. Exploring and evaluating hallucinations in LLM-powered code generation. *arXiv preprint arXiv:2404.00971*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *arXiv:2103.10385*.

Vinsen Marselino Andreas, Genta Indra Winata, and Ayu Purwarianti. 2022. A comparative study on language models for task-oriented dialogue systems. *arXiv e-prints*, pages arXiv–2201.

Microsoft. 2024. Prompt engineering techniques.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, and Erik Cambria. 2023. Recent advances in deep learning based dialogue systems: A systematic survey. *Artificial intelligence review*, 56(4):3055–3155.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Sinan Ozdemir. 2023. *Quick Start Guide to Large Language Models: Strategies and Best Practices for Using ChatGPT and Other LLMs*. Addison-Wesley Professional.

Andrew Peng, Michael Wu, John Allard, Logan Kilpatrick, and Steven Heidel. 2023. Gpt-3.5 turbo fine-tuning and api updates.

Jakub Pizoń and Arkadiusz Gola. 2023. Human–Machine Relationship—Perspective and Future Roadmap for Industry 5.0 Solutions. *Machines*, 11(2):203.

Nitin Rane. 2023. ChatGPT and Similar Generative Artificial Intelligence (AI) for Smart Industry: role, challenges and opportunities for industry 4.0, industry 5.0 and society 5.0. *Challenges and Opportunities for Industry*, 4.

Ivan Sekulić, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, André Ferreira Manso, and Roland Mathis. 2024. Reliable LLM-based User Simulator for Task-Oriented Dialogue Systems. *arXiv preprint arXiv:2402.13374*.

Rashmi Sharma, Sejal Tyagi, and Shivam Chaudhary. 2023. Dialogue System for Human Computer Interaction. *JOURNAL OF TECHNICAL EDUCATION*, page 13.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

Qinyuan Ye, Maxamed Axmed, Reid Pryzant, and Fereshte Khani. 2023. Prompt engineering a prompt engineer. *arXiv preprint arXiv:2311.05661*.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. *arXiv preprint arXiv:2109.03630*.