A Modular LLM-based Dialog System for Accessible Exploration of Finite State Automata

Stefano Vittorio Porta[1], Pier Felice Balestrucci[1], Michael Oliverio[1,*], Luca Anselma[1] and Alessandro Mazzei[1]

[1]Computer Science Department, University of Turin, Italy

Abstract

In the field of assistive technologies, making accessible to visually impaired users complex visual content such as graphs or conceptual maps remains a significant challenge. This work proposes a modular dialog system that leverages a combination of neural Natural Language Understanding (NLU) and Retrieval-Augmented Generation (RAG) to translate graphical structures into meaningful text-based interactions. The NLU module combines a fine-tuned BERT classifier for intent recognition together with a spaCy-based Named Entity Recognition (NER) model to extract user intents and parameters. Moreover, the RAG pipeline retrieves relevant subgraphs and contextual information from a knowledge base, reranking and summarizing them via a language model. We evaluate the system across multiple specific tasks, achieving over 92% F1 in intent classification and NER, and demonstrate that even open-weight models, like DeepSeek-r1 or LLaMA-3.1, can offer competitive performance compared to GPT-4o in specific domains. Our approach enhances accessibility while maintaining modularity, interpretability, and performance on par with modern LLM architectures.

Keywords

Dialogue Systems, Retrieval-Augmented Generation, Large Language Models, Education

1. Introduction

Accessing graphical structures, such as tables, diagrams, and conceptual maps, poses a significant barrier to visually impaired people (VIP), especially in an educational setting, where ensuring equal opportunities for all students is a fundamental requirement. Despite decades of progress in assistive technologies, visual content remains one of the most challenging formats to make accessible. The World Health Organization estimates that at least 2.2 billion people live with near or distance-vision impairment.[1]

While the meaningful alternative text may bridge the accessibility gap, it is rarely implemented effectively. Indeed, for complex visual context a meaningful textual description can be too long for cognitive load constraints. A recent survey about images shared on major social-media and educational platforms found that fewer than 1% were accompanied by any alt text at all, and much of that text was limited to vague placeholders such as *"diagram"* or *"graph"* [1].

Natural Language Processing and Generation (NLP/G) offer promising, yet largely underexplored, approaches for the effective communication of graphical information. The widespread integration of speech-to-text and text-to-speech technologies in modern devices underscores their potential to mitigate accessibility barriers. In this context, *dialog systems* (DSs) can be a powerful tool for teaching graphical structures to VIP, as demonstrated in [2] for instance.

There are various frameworks available to build DSs. Rule-based approaches, such as AIML [3] and VoiceXML,[2] enable the DSs to provide very accurate responses, a critical feature in educational contexts. However, they typically demonstrate limited Natural Language Understanding (NLU), as highlighted in one of our previous works [4]. Alternatively, modular systems, such as the GUS architecture [5], emphasize understanding user utterances by identifying user intent and populating slot frames with information extracted from those utterances. The main challenge for this type of framework lies in the need for annotated dialog examples with labeled slots. End-to-end systems, such as large language models (LLMs), have emerged in recent years as the most popular approach for building DSs, largely due to their ease of use via prompt engineering. Nonetheless, these LLMs face two significant issues that impact their reliability in critical domains like education: (i) the presence of hallucinations in their responses and (ii) a lack of domain-specific knowledge [6]. More recent architec-

CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy

*Corresponding author.

stefano.porta800@edu.unito.it (S. V. Porta); pierfelice.balestrucci@unito.it (P. F. Balestrucci); michael.oliverio@unito.it (M. Oliverio); luca.anselma@unito.it (L. Anselma); alessandro.mazzei@unito.it (A. Mazzei)

0009-0002-4091-4301 (S. V. Porta); 0009-0001-2161-2263 (P. F. Balestrucci); 0009-0007-3448-2377 (M. Oliverio); 0000-0003-2292-6480 (L. Anselma); 0000-0003-3072-0108 (A. Mazzei)

[1]https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2]https://www.w3.org/TR/voicexml20/

tures combine LLMs with modular architectures, such as Retrieval-Augmented Generation (RAG) systems [7], which integrate the text generation capabilities of LLMs with an information retrieval module for selecting and presenting the most relevant information to the user.

In [4], we introduced AIML+, a novel framework based on AIML, specifically developed for building DSs to assist visually impaired students in navigating graphical structures. The use of AIML was motivated by the need to provide accurate responses to users, although it also revealed limitations in terms of NLU. Building on this, and with the goal of creating a reliable system suitable for critical domains such as education, this paper extends our previous work by integrating LLMs into rule-based DSs, resulting in a RAG pipeline. This work aims to improve the often brittle NLU of traditional rule-based approaches and to reduce hallucinations in NLG.

Specifically, our proposal employs a hybrid architecture that combines: (i) an NLU module based on intent classifier and NER to interpret user utterances; (ii) a rule-based information retrieval module to extract relevant information; and (iii) an LLM-based NLG module to generate the system response.

The paper is structured as follows. Section 2 reviews related work in the field of accessible technologies and dialog systems. Section 3 presents the proposed methodology. Section 5 focuses on the performance of the NLU pipeline, Section 6 explains the Dialog Manager and Retrieval Layer logic, while Section 7 evaluates the generation module through both human and automatic assessments. We conclude with a discussion of our findings and future directions in Section 8.[3]

2. Related Work

Accessible technologies have explored various strategies to convey graphical information to VIP, including haptic feedback (e.g., vibrations and touch cues) [8, 9], sonification (data-to-sound mappings) [10, 11], and textual descriptions [12, 13]. While effective in specific contexts, these approaches often lack flexibility, interactivity, and generalizability—particularly when dealing with complex or symbolic visual content. To address these limitations, DSs have been proposed as a more dynamic and user-adaptive interface for mediating access to graphical structures.

Early DSs often relied on hand-crafted rules to parse user input and generate responses. AIML [3], for instance, encodes pattern-response pairs via XML, enabling deterministic rule-based dialogs. Although accessible and interpretable, these systems lack the robustness required to handle ambiguous or context-dependent queries, especially in domains that involve structured or graphical information.

To overcome these limitations, modern DSs increasingly adopt neural NLU methods. Intent classification is commonly modeled as a supervised classification task, where transformer-based models such as BERT have demonstrated state-of-the-art performance [14, 15].

Early NER systems relied on hand-crafted rules and domain-specific features, which required significant human effort and expertise [16]. Recent advances leverage distributed representations, context encoders, and tag decoders, achieving state-of-the-art results with less manual feature engineering [17, 18]

In parallel, RAG has emerged as a prominent approach to enabling language models to ground their responses in external knowledge. Although initially developed for open-domain QA and document-based tasks, its use in structured or symbolic domains, such as graphs, is gaining attention, particularly in educational or assistive settings [19, 20]. However, these systems often focus on general factual retrieval and rarely address the accessibility needs of users navigating inherently visual content.

This work builds upon the NoVAGraphS project, which first proposed transforming non-visual access to graphical content into a dialog-based paradigm via hand-crafted AIML conversational systems [2]. We build on this work by introducing a neural NLU pipeline and a RAG component specifically tailored to the retrieval and generation of descriptions from symbolic graph structures.

3. Methodology

We propose a modular dialog system based on transformer-based components used for both NLU and NLG (see Figure 1). To build the NLU module, we extended an existing resource [21] by applying both automatic data augmentation and manual annotation. In this way, we have been able to train models for both the tasks of (1) Intent Classification and (2) Named-Entity Recognition. The output of the NLU module is then passed to the dialog management module, a rule-based system responsible for retrieving the specific information requested by the user, referred to as *retrieved evidence* in this paper. The retrieved evidence originates from structured knowledge bases that, in the experimentation described below, consists of a specific diagram. The NLG module employs a prompt built by the Dialog Manager to generate from LLMs natural and contextually relevant responses by leveraging both the current user intent and the retrieved evidence.

Given our task-based approach, we focus on dialogs about Finite State Automata (FSA) as a specific case study.

[3] All code and experimental results are publicly available at https://github.com/stefa168/tesi_tln.

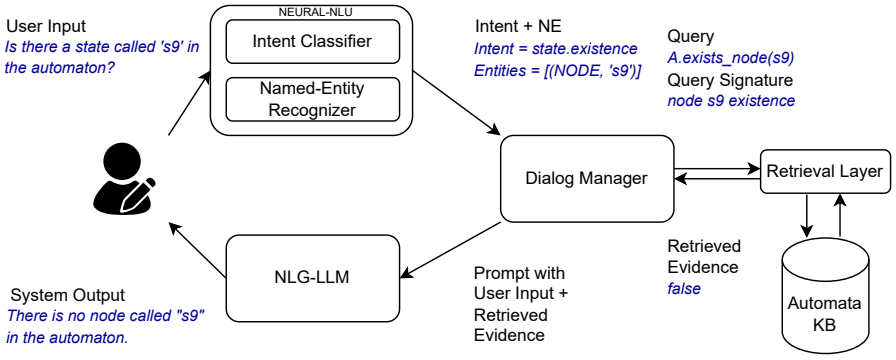


Figure 1: The user input is first processed by the Neural NLU module, which performs intent classification and named-entity recognition. The Dialog Manager then generates a query for the Retrieval Layer, which interrogates the Automaton Knowledge Base (KB) and returns the relevant evidence. This retrieved evidence, together with the original user input, is used to prompt the LLM-based NLG module, which generates a natural language response.

FSA are mathematical models of computation typically taught in computer science degree programs which are often represented as structured graphs. They are formally defined as a quintuple consisting of: (1) a finite set of states Q, (2) a finite set of input symbols Σ, (3) a transition function $\delta : Q \times \Sigma \rightarrow Q$ that maps each state and input symbol to a new state, (4) a start state $q_0 \in Q$, and (5) a set of accepting (or final) states $F \subseteq Q$.

4. Data Collection and Annotation

To develop the NLU module, we built upon an existing resource, the NoVAGraphS corpus [21]. The corpus consists of 32 human–computer conversations focused on the domain of FSA, comprising a total of 706 dialog turns. Since our work focuses on understanding user input, we exclusively use the 353 human utterances from the dataset.

Based on this corpus, we extended the dataset through data augmentation techniques by using a mix of commercial and open-weight LLMs, including `GPT-4o`, `GPT-o1`, and `GPT-o3.mini`, as well as two locally run models, `Llama3.1` and `DeepSeek R1`, generating paraphrases of the original utterances.[4] To ensure data quality, we manually reviewed the synthetic utterances to verify their correctness. In addition, we also included 100 random off-topic questions extracted from the SQuAD 2.0 dataset [22, 23], selected to represent out-of-domain input[5].

The final dataset contains 1,080 user utterances. All utterances, both original and synthetic, were manually annotated by one of the authors—proficient in English—for both intent and entity information.

Intents We used a hierarchical labeling annotation to better capture the specific topic of each user utterance. The resulting dataset consists of two levels of classes: main intents and sub-intents. Specifically, we defined 7 main intents representing the general topic of the question (Table 1). For four of these main intents (AUTOMATON, TRANSITION, STATE, and GRAMMAR) an additional annotation level, called sub-intent, was introduced. This second level includes a total of 32 sub-intents (Table 2), which specify the question's more fine-grained topic depending on the main intent category.

Table 1
Taxonomy of the main intents annotated in the corpus

Main Intent	Description
TRANSITION	Questions concerning transitions between states
AUTOMATON	Questions concerning the automaton in general
STATE	Questions concerning the states of the automaton
GRAMMAR	Questions concerning the grammar recognized by the automaton
THEORY	Questions about general automata theory
START	Questions that initiate interaction with the system
OFF_TOPIC	Questions not relevant to the domain that the system must be able to handle

Entities Entity annotation was performed using the open-source web tool Doccano, resulting in a total of 632 labeled spans across the dataset[6]. Following the

[4] https://openai.com/index/hello-gpt-4o/, https://openai.com/index/openai-o3-mini/, IntroducingOpenAIo1, https://ollama.com/library/deepseek-r1:8b, https://huggingface.co/meta-llama/Llama-3.1-8B

[5] https://huggingface.co/datasets/rajpurkar/squad_v2

[6] https://github.com/doccano/doccano An entity is encoded as `[init-char,fin-char,type]`

Table 2
Sub intents annotated in the dataset divided by main intent.

Main Intent	Sub Intent	Description
AUTOMATON	DESCRIPTION	General descriptions about the automaton
	DESCRIPTION_BRIEF	Brief general description about the automaton
	DIRECTIONALITY	Questions regarding whether the entire automaton is directional
	LIST	General information about nodes and edges
	PATTERN	Presence of particular patterns in the automaton
	REPRESENTATION	Spatial representation of the automaton
TRANSITION	COUNT	Number of transitions
	CYCLES	Questions about loops between nodes
	DESCRIPTION	General descriptions about edges
	EXISTENCE_BETWEEN	Existence of an edge between two nodes
	EXISTENCE_DIRECTED	Existence of an edge from one node to another
	EXISTENCE_FROM	Existence of an outgoing edge from a node
	EXISTENCE_INTO	Existence of an incoming edge to a node
	INPUT	Receiving input from a node
	LABEL	Indication of which edges have a certain label
	LIST	Generic list of edges
	SELF_LOOP	Existence of self-cycles
STATE	COUNT	Number of states
	DETAILS	Specific details about a state
	LIST	General list of states
	START	Which is the initial state
	FINAL	Existence of a final state
	FINAL_COUNT	Number of final states
	FINAL_LIST	List of final states
	TRANSITIONS	Connections between states
GRAMMAR	ACCEPTED	Grammar accepted by the automaton
	EXAMPLE_INPUT	Example input accepted by the automaton
	REGEX	Regular expression corresponding to the automaton
	SIMULATION	Simulation of the automaton with user input
	SYMBOLS	Symbols accepted by the grammar
	VALIDITY	Validity of a given input
	VARIATION	Request for simulation on a modified automaton

annotation process, three entity classes emerged:

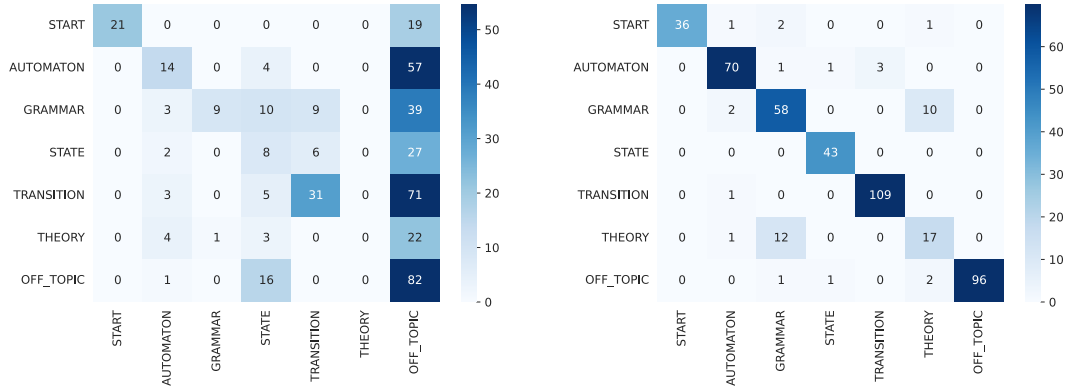
- INPUT: for text fragments containing inputs or sequences of symbols. For example, in the sentence *"Does it only accept 1s and 0s?"* there are two entities of type `INPUT: [20,21,"input"], [27,28,"input"]`;
- NODE: for text fragments containing nodes or states of the automaton. For example, in the sentence *"Is there a transition between q2 and q0?"* there are two entities of type `NODE: [30,32,"node"], [37,39,"node"]`;
- LANGUAGE: for text fragments containing information about the language accepted by the automaton. For example, in the sentence *"Does the automaton accept strings over the alphabet {0,1}?"* there is one entity of type `LANGUAGE: [53,58,"language"]`.

5. Neural NLU

The first module of our architecture handles NLU through a two-step pipeline: (i) **Intent Classification** and (ii) **Named-Entity Recognition**. The goal is to extract a structured representation of the user's utterance by identifying the intent and the entities in the user input. For example:

> **Input**: *"Is there a state called s9 in the automaton?"*
> **Output**: {
> `Intent = state.existence,`
> `Entities = [(NODE, 's9')]`
> }

To build the NLU module, we trained two models for Intent Classification and Named-Entity Recognition using the corpus described in Section 4, and we evaluated them against the AIML system we proposed in [24].



(a) AIML baseline

(b) BERT model

Figure 2: Confusion matrices for the AIML baseline and the fine-tuned BERT model on the main intent classification.

Intent Classification For intent classification, we fine-tuned a `BERT-base-uncased` model[7] for both main and sub-intent classification. The dataset was split into 60% training, 20% development, and 20% testing. We fine-tuned with the following hyper-parameters: 20 epochs, LR 2×10^{-5}, linear warm-up 10%, batch 16. Training was logged with Weights & Biases. Our approach significantly outperforms the AIML baseline, achieving a macro-F1 score of 0.92 on main intents and 0.86 on sub-intents. This marks a substantial improvement over AIML, which scores only 0.33 and 0.20, respectively (see Table 3). Figure 2 compares the confusion matrices for both systems, showing that BERT produces far fewer off-topic errors and handles ambiguous utterances more robustly.

Table 3
Performance on main and sub-intent classification for the fine-tuned BERT model and the AIML baseline (↑ **higher is better**).

Model	Main Intent F1	Sub-intent F1	NER
BERT (ours)	**0.92**	**0.86**	**0.92**
AIML baseline	0.33	0.20	-

Named Entity Recognition NER is handled using a simplified spaCy v3 pipeline that exclusively employs the NER component on top of a blank model,[8] fine-tuned on our annotated dataset with the same data split (60/20/20). The pipeline is based on the transformer architecture [25] and identifies domain-specific entities such as `states`, `transitions` and `input strings`. It achieves an F1-score of 0.92 on the test set (see Table 3).

[7] https://huggingface.co/google-bert/bert-base-uncased
[8] https://spacy.io/usage/v3

6. Dialog Manager and Retrieval Layer

The Dialog Manager is responsible for orchestrating the interaction flow by interpreting the NLU output and coordinating the appropriate system response. This involves analyzing the classified intent and any associated entities, and invoking the corresponding function from the Retrieval Layer.

The Retrieval Layer is activated whenever the recognized intent is relevant to the domain, thus neither `START` nor `OFF TOPIC`. Indeed, `START` typically triggers a welcome message, while `OFF TOPIC` handles inputs outside the system's scope. Since these cases do not require access to the automaton's knowledge, retrieval is skipped.

For domain-specific intents (e.g., checking the existence of a state), the Dialog Manager uses a rule-based system that maps intent–entity pairs to specific queries. This design ensures transparency and precise control over system behavior. For instance, when the intent is `state.existence` and the entity is a node identifier like `'s9'`, the Dialog Manager calls the function `exists_node('s9')`. This function queries the underlying automaton representation to determine whether the specified node exists. The automaton is stored in a Knowledge Base (KB) constructed using the NetworkX Python library,[9] which allows efficient graph manipulation. The automaton's structure is serialized in DOT format, a standard for graph description, and visualized using Graphviz.[10]

The Retrieval Layer then returns a structured output (e.g. `false`, if the node is not found), which is passed to the NLG module for the generation of the final response.

[9] https://networkx.org/
[10] https://graphviz.org/

7. LLM-based NLG

For the NLG module, we adopt a prompting strategy based on LLMs that uses both the user input and the output of the Dialog Manager to generate contextually relevant and accurate responses. This technique is widely adopted in RAG systems [7], as it enables the model to ground its answers in retrieved evidence, reducing hallucinations and increasing factual accuracy. Our prompt template drives the model to act as a domain-specific expert — in this case, for finite state automata — instructing it to use only the retrieved data without introducing extraneous information or explicit references to the source. This approach helps maintain concise, focused answers that avoid potential confusion or unverifiable content.

```
System prompt:
"You are a helpful assistant expert
in finite state automata.
Answer the question given by the
user using the retrieved data,
using plain text only. Avoid
referring to the data directly;
there is no need to provide any
additional information.
Keep the answer concise and short,
and avoid using any additional
information not provided.
The system has retrieved the
following data:
{Retrieved Evidence}
The user has asked the following
question:
{User Input}"
```

We evaluate this module by comparing five LLMs with different characteristics: two commercial models, `GPT-4o` and `GPT-o3-mini`, and three open-weight models, `DeepSeek-r1-8B`, `Gemma2-9B`, and `LLaMA3.1-8B`.[11]

To assess the quality of the generated answers, we conducted a human evaluation using the *FactGenie* platform [26]. A group of 12 volunteer annotators labeled each generation according to four error categories defined by the taxonomy in Kasner and Dusek [27]. In particular: INCORRECT indicates that the text contradicts the data; NOT-CHECKABLE means the information cannot be verified; MISLEADING refers to text that is deceptive given the context or omits crucial information; and OTHER includes problematic cases that do not fit into the other categories. In addition to human annotation, we also performed automatic labeling using GPT-4.5[12] (*LLM-as-a-Judge*), applying the same error taxonomy. The annotator pool included 8 students from the Department of Computer Science, 2 with an engineering background, and 2 from the Departments of History and Biology. The average age was 28, with a range from 21 to 68 years. Each annotator evaluated a subset of the responses, with overlapping assignments to ensure that all 75 generated answers were reviewed by multiple judges.

Table 4
Average percentage of answers containing at least one labeled error, computed by aggregating the four error categories (INCORRECT, NOT-CHECKABLE, MISLEADING, OTHER). Lower values indicate better performance.

Generator	Human error ↓	GPT-4.5 error ↓
GPT-o3-mini	**7.3**	**6.6**
GPT-4o	8.7	7.1
DeepSeek-r1-8B	26.7	13.3
Gemma2-9B	33.3	26.7
LLaMA3.1-8B	46.7	33.3

Table 4 summarizes the aggregated error rates across the four categories, demonstrating that `GPT-o3-mini` consistently achieves the lowest error rates under both human and GPT-4.5 evaluation. Among the open-weight models, `DeepSeek-r1-8B` shows the most competitive performance, outperforming other open models by a substantial margin. These results highlight the effectiveness of the prompting strategy in generating accurate and reliable responses grounded in retrieved data.

In addition to the error-based evaluation, we introduced four qualitative dimensions to assess the overall quality of the interactions: CLARITY, USEFULNESS, OVERALL APPRECIATION, and FACTUAL ACCURACY. These dimensions offer a more holistic perspective on the responses, going beyond binary correctness.

- CLARITY: whether the response is understandable and well-structured;
- USEFULNESS: whether the response is helpful and provides relevant information;
- OVERALL APPRECIATION: whether the response is perceived as satisfactory or positively received by the annotator;
- FACTUAL ACCURACY: whether the response is entirely correct and free from factual errors.

The same group of 12 human annotators performed labeling according to these dimensions.

Table 5 shows that `GPT-o3-mini` receives the most favorable user judgments across all dimensions. Among open-weight models, `DeepSeek-r1-8B` is the most positively rated, while `LLaMA3.1-8B` and `Gemma2-9B` receive consistently lower preferences from annotators.

[11] https://openai.com/index/hello-gpt-4o/, https://openai.com/index/openai-o3-mini/, https://ollama.com/library/deepseek-r1:8b, https://huggingface.co/google/gemma-2-9b, https://huggingface.co/meta-llama/Llama-3.1-8B

[12] https://openai.com/index/introducing-gpt-4-5/

Table 5
Percentage of answers regarding how they were perceived by human annotators. Arrows indicate the direction of better results (↓ lower is better, ↑ higher is better). Abbreviations: CL = clarity, US = usefulness, OA = overall appreciation, FA = factual accuracy.

Model	CL ↑	US ↑	OA ↑	FA ↑
GPT-o3-mini	**92.7**	**98.0**	**95.3**	**98.7**
GPT-4o	86.0	90.0	86.7	90.0
DeepSeek-r1 8B	69.3	82.0	68.0	70.0
LLaMA3.1 8B	63.3	68.0	58.0	71.3
Gemma2 9B	56.0	58.7	36.7	66.0

8. Conclusions

This work presents a significant advancement over previous systems aimed at the exploration of graphical structures, by proposing a hybrid modular architecture that integrates NLU and NLG techniques based on Transformers and LLMs. The implemented DS addresses several key limitations of rule-based DSs, such as rigid pattern matching, limited context handling, and difficulties in interacting with external data sources.

Compared to AIML, our system stands out for its greater expressive flexibility and its ability to adapt to complex conversational flows, thanks to a more articulated dialog management mechanism. The introduction of a neural classifier for intent recognition, along with a spaCy-based NER module, has substantially improved the robustness of natural language understanding, achieving F1 scores above 90% for both Intent Classification and NER. Moreover, the RAG component has significantly reduced hallucinations and ambiguity in generation, providing contextually accurate responses that are well-grounded in structured data.

The results demonstrate that a hybrid and modular approach can ensure accessibility, reliability, and control—fundamental features for the adoption of DSs in educational and assistive contexts. Our framework therefore represents a concrete step toward more interpretable, adaptable, and user-centered intelligent DSs. In future works we plan to evaluate the complete system with blind people.

9. Limitations

While the system shows strengths in modularity, accuracy, and integration of LLMs, a significant limitation persists: its accessibility has yet to be validated with learners. Although designed with accessibility in mind, the system's real-world effectiveness and usability—especially for visually impaired individuals interacting with graphical content—remain untested. Conducting a structured evaluation with these target users is crucial to determine its pedagogical impact and practical usability.

References

[1] R. Power, The ALT Text: Accessible Learning with Technology, 2024.

[2] P. F. Balestrucci, L. Anselma, C. Bernareggi, A. Mazzei, Building a spoken dialogue system for supporting blind people in accessing mathematical expressions, in: F. Boschetti, G. E. Lebani, B. Magnini, N. Novielli (Eds.), Proceedings of the 9th Italian Conference on Computational Linguistics (CLiC-it 2023), CEUR Workshop Proceedings, Venice, Italy, 2023, pp. 70–77. URL: https://aclanthology.org/2023.clicit-1.10/.

[3] R. Wallace, The Elements of AIML Style, ALICE A.I Foundation, 2001. Available at https://files.ifi.uzh.ch/cl/hess/classes/seminare/chatbots/style.pdf.

[4] M. Oliverio, M. Piroi, D. De Giorgi, P. F. Balestrucci, C. Manolino, A. Mazzei, L. Anselma, C. Bernareggi, M. Serio, C. Sabena, T. Armano, S. Coriasco, A. Capietto, Novagraphs: Towards an accessible educational-oriented dialogue system, in: Proceedings of the Second International Workshop on Artificial INtelligent Systems in Education co-located with 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), 2024.

[5] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, T. Winograd, Gus, a frame-driven dialog system, Artificial Intelligence 8 (1977) 155–173. URL: https://www.sciencedirect.com/science/article/pii/0004370277900182. doi:https://doi.org/10.1016/0004-3702(77)90018-2.

[6] A. Abusitta, M. Q. Li, B. C. Fung, Survey on explainable ai: Techniques, challenges and open issues, Expert Systems with Applications 255 (2024) 124710.

[7] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, 2024. URL: https://arxiv.org/abs/2312.10997. arXiv:2312.10997.

[8] C. Bernareggi, C. Comaschi, G. Dalto, P. Mussio, L. Parasiliti Provenza, Multimodal exploration and manipulation of graph structures, in: Proceedings of the 11th International Conference on Computers Helping People with Special Needs, ICCHP '08, Springer-Verlag, Berlin, Heidelberg, 2008, p. 934–937. doi:10.1007/978-3-540-70540-6_140.

[9] C. Bernareggi, D. Ahmetovic, S. Mascetti, muGraph: Haptic Exploration and Editing of 3D Chemical Di-

agrams, in: Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 312–317. doi:10.1145/3308561.3353811.

[10] D. Ahmetovic, C. Bernareggi, J. a. Guerreiro, S. Mascetti, A. Capietto, Audiofunctions.web: Multimodal exploration of mathematical function graphs, in: Proceedings of the 16th International Web for All Conference, W4A '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–10. doi:10.1145/3315002.3317560.

[11] J. Su, A. Rosenzweig, A. Goel, E. de Lara, K. N. Truong, Timbremap: enabling the visually-impaired to use maps on touch-enabled devices, in: M. de Sá, L. Carriço, N. Correia (Eds.), Proceedings of the 12th Conference on Human-Computer Interaction with Mobile Devices and Services, Mobile HCI 2010, Lisbon, Portugal, September 7-10, 2010, ACM International Conference Proceeding Series, ACM, 2010, pp. 17–26. doi:10.1145/1851600.1851606.

[12] V. Sorge, M. Lee, S. Wilkinson, End-to-end solution for accessible chemical diagrams, in: Proceedings of the 12th International Web for All Conference, W4A '15, Association for Computing Machinery, New York, NY, USA, 2015. doi:10.1145/2745555.2746667.

[13] S. Chockthanyawat, E. Chuangsuwanich, A. Suchato, P. Punyabukkana, Towards automatic diagram description for the blind, in: i-CREATe. The International Convention on Rehabilitation Engineering and Assistive Technology, 2017, pp. 1–4. doi:10.13140/RG.2.2.11969.04961.

[14] Z. Zhang, Z. Zhang, H. Chen, Z. Zhang, A joint learning framework with bert for spoken language understanding, IEEE Access 7 (2019) 168849–168858. doi:10.1109/ACCESS.2019.2954766.

[15] M. Roman, A. Shahid, S. Khan, A. Koubâa, L. Yu, Citation intent classification using word embedding, IEEE Access 9 (2021) 9982–9995. doi:10.1109/ACCESS.2021.3050547.

[16] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, Lingvisticae Investigationes 30 (2007) 3–26. doi:10.1075/LI.30.1.03NAD.

[17] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, IEEE Transactions on Knowledge and Data Engineering 34 (2018) 50–70. doi:10.1109/TKDE.2020.2981314.

[18] P. Liu, Y. Guo, F. Wang, G. Li, Chinese named entity recognition: The state of the art, Neurocomputing 473 (2021) 37–53. doi:10.1016/j.neucom.2021.10.101.

[19] B.-S. Posedaru, F.-V. Pantelimon, M.-N. Dulgheru, T.-M. Georgescu, Artificial intelligence text processing using retrieval-augmented generation: Applications in business and education fields, Proceedings of the International Conference on Business Excellence 18 (2024) 209 – 222. doi:10.2478/picbe-2024-0018.

[20] F. Miladi, V. Psyché, D. Lemire, Leveraging gpt-4 for accuracy in education: A comparative study on retrieval-augmented generation in moocs (2024) 427–434. doi:10.1007/978-3-031-64315-6_40.

[21] E. Di Nuovo, M. Sanguinetti, P. F. Balestrucci, L. Anselma, C. Bernareggi, A. Mazzei, Educational dialogue systems for visually impaired students: Introducing a task-oriented user-agent corpus, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 5507–5519.

[22] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, SQuAD: 100,000+ questions for machine comprehension of text, in: J. Su, K. Duh, X. Carreras (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 2383–2392. URL: https://aclanthology.org/D16-1264. doi:10.18653/v1/D16-1264. arXiv:1606.05250.

[23] P. Rajpurkar, R. Jia, P. Liang, Know what you don't know: Unanswerable questions for SQuAD, in: I. Gurevych, Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 784–789. URL: https://aclanthology.org/P18-2124. doi:10.18653/v1/P18-2124. arXiv:1806.03822.

[24] P. F. Balestrucci, E. Di Nuovo, M. Sanguinetti, L. Anselma, C. Bernareggi, A. Mazzei, An educational dialogue system for visually impaired people, IEEE Access (2024).

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

[26] Z. Kasner, O. Platek, P. Schmidtova, S. Balloccu, O. Dusek, factgenie: A framework for span-based evaluation of generated texts, in: S. Mahamood, N. L. Minh, D. Ippolito (Eds.), Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations, Association for Computational Linguistics, Tokyo, Japan, 2024, pp. 13–15. URL: https://aclanthology.org/2024.inlg-demos.5/. doi:10.18653/v1/2024.inlg-demos.5.

[27] Z. Kasner, O. Dusek, Beyond traditional benchmarks: Analyzing behaviors of open LLMs on data-to-text generation, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 12045–12072. URL: https://aclanthology.org/2024.acl-long.651/. doi:10.18653/v1/2024.acl-long.651.