Introducing Janet: Early Findings on a Conversational Agent for Virtual Research Environments

Ahmed Salah Tawfik Ibrahim, Leonardo Candela
Istituto di Scienza e Tecnologie dell'Informazione (ISTI)
Italian National Research Council (CNR)
Via G. Moruzzi 1, 56124 Pisa, Italy
Email: ahmed.ibrahim@isti.cnr.it, leonardo.candela@isti.cnr.it

***Abstract*—Conversational agents have the potential to streamline tasks, provide support, and enhance user experience across various domains including Virtual Research Environments (VREs). The recent progress in conversational artificial intelligence and large language models (LLMs) offers novel strategies for the development of these agents. Janet is an attempt to develop an agent that, by leveraging the resources within the VRE, can engage in natural language conversations with VRE users to help them manage their daily activities, find relevant information, and use what the specific environment offers. It is developed following the Retrieval-Augmented Generation paradigm, a technique that reduces the effect of one of the limitations affecting LLMs; namely, hallucination. This paper highlights the lessons learned during the development of Janet.**

***Keywords*—Conversational Agents, Natural Language Processing, Large Language Models, Retrieval-Augmented Generation, Virtual Research Environments**

I. Introduction

Conversational artificial intelligence has witnessed massive improvements over the past couple of years. Specifically, with the release of large language models (LLMs) like ChatGPT and Gemma by OpenAI and Google respectively, the real benefits of such models are starting to unveil. One such benefit is their capacity to respond to prompts in a human-like fashion providing what seems to be factually correct responses. However, what actually happens is that these models generate the most likely response to the prompt. This is determined according to the text they were trained on offline. In other words, they have no access to any knowledge when they generate a response. This is why their responses may be biased, outdated or factually wrong. But what if, when generating a response, these models get access to external knowledge? This is where virtual research environments (VREs) [1] come into the picture. Such environments, conceived to support collaborative research work, are by definition possibly rich in context-specific knowledge that comes in both structured and unstructured forms including papers, datasets, posts and description of other shared resources including services and processes. That is why equipping them with a context-aware conversational agent can facilitate the exploitation of these knowledge sources. Such agent can be implemented as a combination of an LLM and a knowledge retrieval component that would reduce the effect of hallucination by the LLM. Thus, we equip the VRE with a conversational agent that uses the VRE's content as context in an attempt to enhance the user experience.

Janet is the work-in-progress conversational agent conceived to empower D4Science-based VREs [2], [3] which aims at exploiting the domain-specific knowledge of each VRE to assist researchers while leveraging LLMs to provide a conversational interface to the end user. Our plan is to weave together an open set of components that can perform different functionality in response to a textual query then use an LLM to generate a human-like response.

We expect to face several challenges related to the context in which we are developing Janet. VREs are not conceived to support a well-defined task. Indeed, every VRE is a specific environment conceived to support the tasks of its designated community possibly using similar patterns and services. Hence, the agent needs to be adaptive to the needs of the VRE it operates in. Moreover, these environments potentially contain large volumes of data in different modalities, so a way to organize and efficiently retrieve them is needed.

In Section II, we look into the literature for possible solutions to the aforementioned challenges. Then, Section III explains the design decisions characterising Janet. Section IV presents the experimental setup and the main results. Section V highlights the lessons learned and the future plans for Janet. Finally, Section VI concludes the paper by providing a summary.

II. Related Works

Before looking into the possible solutions to the challenges we expect to face, it is useful to highlight what virtual research environments are. VREs, also known as science gateways, are web-based systems that serve a certain scientific community by providing its members with the facilities needed to accomplish their goals while being open and flexible; allowing the community to control the way they share their results [1], [4], [5]. Their main goal is to facilitate collaboration among scientists given the current trend where science is becoming more global and interconnected. The facilities provided by VREs include, but are not limited to, access to datasets, computational resources, services and research artifacts like research papers and reports.

That being said, developing a conversational agent for a VRE is different from developing it for a traditional collaborative environment [6]. To illustrate, we highlight a number of conversational agents that were developed for the latter case. Bert is one such agent that was developed to help a geographically distributed team of astrologists with their observational tasks by taking the burden of notifying them about the occurrence of certain astrological events and providing information about user queries [7]. Another one is InfoBot which is an online tutor for university students which is deployed within their collaborative environment [8]. It helps with answering questions related to a courses material or logistics and it even offers an assessment to the students' understanding via quizzes. One last example is Demic which has been developed and integrated within a virtual social network where information of users' profiles along with the dialog context are used to generate responses [9].

As we can see, agents for a traditional collaborative environment are usually conceived to solve a certain well-defined task whereas agents for VREs must embrace the VRE specificity in terms of tasks, workflows, etc.

A possible solution to this challenge is simply deploying the agent in a way that allows it to improve over time. Evorus [10] is a crowd-based system which incorporates a voting mechanism to select the best response from a set of candidate responses provided by a set of crowd workers and chatbots to a user's query. It is very well-suited for open-domain problems as it improves itself over time by allowing specialized chatbots to be incorporated into the system and then it learns to select the best chatbot to provide an answer to a certain query. It also learns how to reuse previous responses.

That being said, a more recent mechanism that has proven more effective is reinforcement learning from human feedback (RLHF). It involves learning a reward function to score the responses and then optimize the policy used to generate the response using the learned function [11]. To achieve this, both the agent policy and the reward function are modelled as neural networks where the parameters of the policy network are learned via a traditional reinforcement learning method. Basically, the agent interacts with the environment and produces some outputs and then pairs of these outputs are given to a human to choose the preferred output. Then, the reward network takes these comparisons and fits them in order to implicitly learn a reward function that reflects the preferences of the human. Finally, the policy is updated in a traditional fashion where the reward is generated by the reward neural network instead of explicitly defining a reward function. This paradigm has been used by [12] to learn a model to summarize texts where the policy network is a pre-trained LLM that is then optimized using a reinforcement learning algorithm after learning the reward function.

Another expected challenge arises from the VRE's potential of containing volumes of data items, like papers and datasets with their metadata and content written in natural language. They could also contain posts and comments by the users which are also written in natural language. Other modalities for the content may also be present. Hence, the conversational agent within such a VRE will have to be capable of performing the task of information retrieval efficiently.

A possible solution is realized by conversational information retrieval (CIR) systems. These systems make use of advanced language models in order to efficiently answer the user's query. A CIR system is made up of a data layer and an engine [13]. The data layer contains various forms of databases that encapsulate the knowledge of the agent. The engine, however, contains a query analyzer, a dialog manager and a set of components to perform certain actions. Basically, the message analyzer performs a set of steps in order to understand the query. These include resolving co-references and recognizing named entities. The dialog manager tracks the state of the conversation and chooses the component to use from the set of components a CIR system is supposed to realize.

A particular action that is interesting is retrieval-augmented language generation (RAG) which refers to the task of generating text by depending not only on the generative model's parameters but on retrieved content as well. This is useful for tasks like open-book question-answering where the answer to a question is based on a context paragraph. As described by [13], in order to implement this task, a retriever and a generator should be developed. The retriever is typically a neural network used to compute representations of content in such a way that makes similar items close to each other in the representation space. It is then used to efficiently retrieve the most similar item to the query from the data layer. The generator, however, is a conditional sequence-to-sequence model, or an LLM, that generates an answer to a question given a context. As shown by [14]and [15], a sequence-to-sequence encoder-decoder model can be trained by concatenating the query to the relevant context and providing this as input to the encoder in order to generate a representation of them. Then, the decoder is used to transform this combined representation into an answer by training it to minimize the sequence to sequence loss between the decoded answer and the reference answer.

III. Methodology

Janet is developed to operate in the VREs of D4Science. D4Science is an infrastructure allowing the creation and management of VREs with the as-a-Service delivery mode [2], [3]. It provides the VREs with computational resources such as storage capacity and analytics options. In addition, it provides a social networking platform and a publishing platform to enable collaboration among the members. Fig. 1 by [3] shows an overall picture of the framework supported by D4Science. Janet initial prototype makes use of the posts in the social platform, the datasets and the textual content, like papers, in the storage.

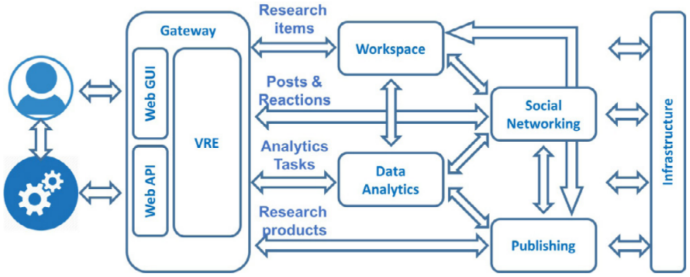


Fig. 1: D4Science Framework from [3]

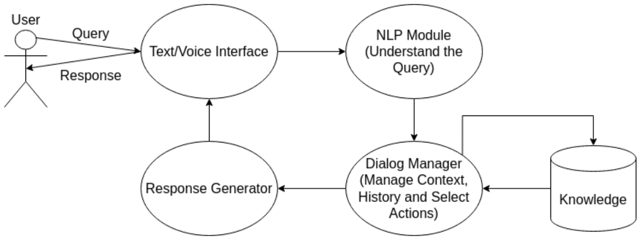

User
Query
Response
Text/Voice Interface
NLP Module (Understand the Query)
Dialog Manager (Manage Context, History and Select Actions)
Knowledge
Response Generator

Fig. 2: General Architecture of Conversational Agents

A. General Architecture

The philosophy behind the design of Janet is that of modularity and continuous improvement. This is why we follow the architecture referenced in [16] which is illustrated in Fig. 2.

This architecture allows for the development of different components that can implement various functionality. In particular, the NLP module is responsible for deciding what functionality the user is looking for. Then, the dialog manager decides which component to use based on the desired functionality and the dialog history. The chosen component may query a knowledge source before preparing the content of the response. Finally, the response generator uses the content returned by the component to build a human-like reply.

B. Proposed Implementation

In the initial prototype, we propose an initial set of components that can enable a general set of functionality. These functionality include question answering, resource retrieval, paper summarization and resource recommendation. The VRE resources that were considered for the initial prototype are the textual content of papers, the textual content of the metadata of papers, and datasets and the textual content of the posts inside the VRE. The components that can be used to realize these functionality are a neural retriever, a language generator and a recommender. The neural retriever is implemented as a sentence transformer; namely, the mpnet-base sentence transformer which maps paragraphs into dense 768-dimensional vectors. The language generator, however, is nothing but an LLM; in particular, a fine-tuned T5 model [17]. T5 was chosen for the initial prototype as it is available for free, unlike the more powerful LLMs that emerged recently like GPT 3.5 and subsequently GPT 4. Finally, the recommender is implemented by profiling the users' interests which are extracted from their queries. To do this, an entity extractor is used to extract topics or resources of interest from the user's natural language queries. These interests are then ranked according to their frequency and recency.

These components are augmented with a knowledge base that is constructed from the resources that can be found in the VRE. It is worth noting that an ideal knowledge base would encompass various forms of knowledge representation which may include, but are not limited to, a graph database and a vector database. However, in our initial prototype, we make use of a vector index for its simplicity. In particular, it is implemented using FAISS [18].

In order to account for ambiguity and offensive language, we plug an ambiguous query classifier and an offensive language classifier into the NLP module. Their job is to flag the queries for the dialog manager to decide the proper action to take.

Upon receiving a query and extracting the relevant information from it; i.e., the requested functionality (or the intent), the user's interests and the requested resources (if any), by the NLP module, the dialog manager decides which components to use. To do this, it has been implemented as a finite state machine as shown in Fig. 3. The intent represents the requested functionality which determines the components to use. For question answering, the retriever is used to get the most relevant content from the vector index which is then passed as context to the LLM (or the generator) to generate the response. As for paper summarization, the LLM is prompted to summarize a paper. The paper is determined by extracting its title, topic or author from the query using the entity extractor and then the retriever fetches it and passes it to the LLM. Resource retrieval and recommendation work similarly as the entity extractor and the retriever are used to determine which resources to fetch.

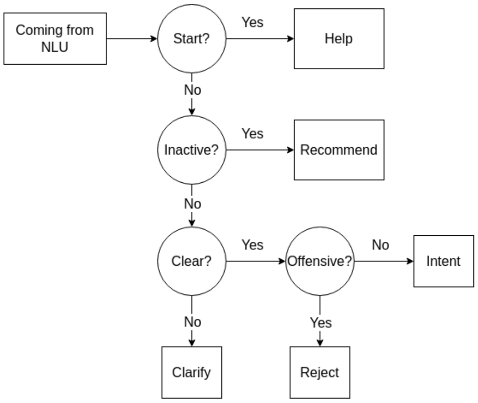


Fig. 3: Dialog Manager Finite-State Machine

To account for continuous self-improvement, users are asked to answer a set of questions related to each response they receive from Janet. The goal is to collect a large enough dataset of user feedback about the fluency, the correctness and the usefulness of the responses. This, in turn, shall be used to train a reward model which will be used to improve the way the responses are generated following the RLHF paradigm [11].

Fig. 4 summarizes the proposed system architecture where we employ a master-slave architecture. The master is part of the underlying infrastructure and is responsible for creating and updating the models used by the different components of the agent. The workers are deployed into the VREs and they contain all the implemented components. The knowledge base, however, is different for each worker as it is derived from the VRE in which the worker is deployed. The workers collect the user feedback and forward it to the master where the enhancement should happen.

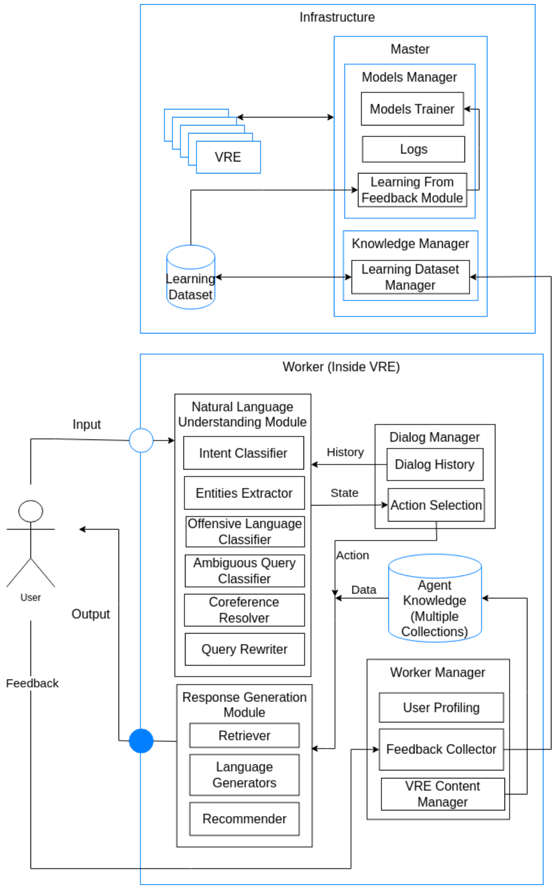


Fig. 4: System Architecture

IV. Experimental Results

In this section, we report (*i*) the results obtained during the development of the tools, and (*ii*) the initial evaluation of the entire system after deployment.

A. Evaluation of the Components

In order to develop each of the reported components, we had to perform a training phase to a number of neural networks using different datasets.

1) Intent Classifier: As mentioned previously, the dialog manager determines the component to use based on the functionality embedded in the user's query. This functionality is determined by an intent classifier which is nothing but a distilled version of the famous transformer BERT [19]. The dataset used to finetune it was curated manually. It contains examples in the form of <text, intent> which amount to 275 labeled sentences. The labels (intents) include (*a*) question-answering (QA), (*b*) chitchatting, (*c*) retrieving a catalogue item (papers or datasets) and posts, (*d*) summarizing a paper, (*e*) affirmation, (*f*) negation, (*g*) listing catalogue items (papers and datasets) and VRE topics, (*h*) asking for help regarding how to use Janet. The transformer was finetuned on 80% of the dataset and tested against the remaining 20%. In the end, we saved the parameters that had the highest F1-score, which was 94.5%.

2) Entity Extractor: In order to extract useful information from the query, an entity extractor was trained. In particular, the information that was considered for the prototype include the topic of interest, the type of the resource, the title of the resource, the author of the resource and the exact date of publication. Therefore, we manually curated a dataset whose examples are in the form of <text, entities> where entities is a list of tuples containing an entity label and the start and end indices of the character span of that entity in the text. This dataset contains 97 tuples where each tuple is a sentence plus the entities within it. The neural network that was finetuned on this dataset is RoBERTa which is based on BERT and is the base of the state of the art token classifiers [20]. Similar to the intent classifier, the dataset was split to perform training and testing which resulted in saving the model with an F1-score of 100% on the validation set which was achieved probably due to the small size of the dataset.

3) Neural Retriever: The neural retriever, as mentioned previously, is a finetuned mpnet sentence transformer. The dataset it was finetuned on was constructed from different sources ending up with 313,156 examples in the form of <query, context>. These sources include manually curated question-context pairs extracted from the user manual of D4Science, the MS-Marco v2.1 dataset [21], the PUBMED QA dataset [22], the QASPER dataset [23], and the ScienceQA dataset [24]. The dataset was split into training and test sets. Then, the sentence transformer was finetuned for only 10 epochs because it required 4 hours to complete one epoch. The best model in terms of the mean average precision at 100 on the test set scored 81.12%.

4) Generator: The generator, or the LLM, was developed by finetuning the T5 transformer. T5 provides the possibility of performing multi-task training by simply appending a task prefix to each training example. For example, a QA training example can be in the form <question: text context: text, answer> where question: and context: are QA-specific prefixes. Therefore, we curated a dataset containing examples in the form of <text, target text>, where text is simply the query augmented with task specific prefixes for each of the tasks we wanted our prototype to support. For the QA, the

TABLE I: Number of Responses evaluated by Length and Fluency

	Short	Appropriate	Long
Response Length	16	78	6
	Basic	Intermediate	Fluent
Response Fluency	19	6	75

TABLE II: Number of Responses evaluated by Factuality and Usefulness

	True	False
Response Factuality	94	6
	Useful	Useless
Response Usefulness	69	31
Evidence Usefulness	13	6

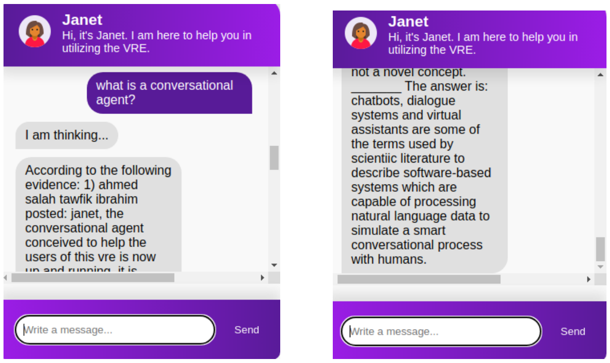
same dataset used for the retriever was modified to have the questions appended to the contexts as the query string and the answer as the target. For the summarization, we used the OpenAI Summarize From Feedback dataset [25] and the XSum dataset [26]. Finally, for general chitchatting, we used the PersonaChat dataset [27]. The training was then performed for a total of 25 epochs on the training split of the dataset which lasted for 6 days. The best model had a RougeL score of about 29.3 on the test set.

B. Overall Evaluation of the Prototype

In order to test the overall prototype, a Janet worker instance was deployed in one VRE that was supplied with a number of research papers, datasets and posts. Then, we let the users have a total of 100 interactions; i.e., query-reply pairs. Users were then asked to fill out a questionnaire after each reply they got from Janet. Due to the fact that it is a preliminary work, the users that participated in the evaluation were internal to our organization and the evaluation was open for a couple of days. We managed to collect the feedback of 6 users, so we cannot draw statistically significant conclusions from their participation. Nonetheless, their contribution in the study was helpful in guiding our future plans to improve Janet as our goal in this initial prototype was not to have a rigorous evaluation of Janet.

The questionnaire is aimed at evaluating the length of the response (short, long or acceptable), the fluency of the response (basic, intermediate, fluent), the factuality of the response and the speed of generating it. Furthermore, users are asked to report if the answer to their question, when performing a question-answering task, was contained in the evidence provided by the agent. It's worth noting that out of the 100 interactions, 19 were question-answering tasks. They are also asked to specify the intent, or the goal, they were trying to achieve by the query. Finally, users are asked to provide a better response if they are willing to, which will be used in the future to enhance the language generation.

Tables I and II report the number of responses for each possible value of each evaluation metric.



Janet
Hi, it's Janet. I am here to help you in utilizing the VRE.

what is a conversational agent?

I am thinking...

According to the following evidence: 1) ahmed salah tawfik ibrahim posted: janet, the conversational agent conceived to help the users of this vre is now up and running. it is

(a) Asking a Question

Janet
Hi, it's Janet. I am here to help you in utilizing the VRE.

not a novel concept. ________ The answer is: chatbots, dialogue systems and virtual assistants are some of the terms used by scientiic literature to describe software-based systems which are capable of processing natural language data to simulate a smart conversational process with humans.

(b) Getting the Answer

Fig. 5: Question Answering Example

V. Lessons Learned and Future Plan

This preliminary prototype showcases that a conversational agent is a useful addition to the VREs. It can save a lot of time and effort for the users when they are using the resources. For instance, it can answer questions whose answers may be contained in one of the papers, removing the need to read the whole paper to find a specific piece of information. Fig. 5 outlines one of the question-answering interactions with Janet, which displays the relevant content it retrieved in addition to the generated answer.

Consequently, it became evident that the way we split the textual content has a significant effect on Janet's performance. Text needs to be split in a way that preserves the context so that, when retrieval is performed, all the relevant content can be fetched correctly. The plan is to experiment with different splitting strategies to select the best ones.

Thus, organizing unstructured data, such as PDF files, into structured knowledge, like a graph, is core to the performance of Janet. Therefore, our efforts will be dedicated to developing a suite of components aimed at extracting knowledge from various unstructured sources with the consideration of different data modalities. The aim is building a comprehensive knowledge base, potentially made of a graph database and a vector database, that would empower Janet.

Moreover, our existing set of components requires reevaluation and improvement. For instance, the neural retriever was randomly selected from publicly available sentence transformers. We plan to experiment with different retrievers to optimize performance. Additionally, incorporating a re-ranker, proven effective in practice, can further enhance the retrieval system.

Similarly, upgrading the generator with a more powerful LLM is feasible now that Google's Gemma is accessible for free. An interesting observation was that prompting is an effective technique that enhances the performance of LLMs. That is why we will be studying how to incorporate prompting into our response generation pipeline. In other words, the goal is to enrich the user's queries with effective prompts with the aim of enhancing the quality of the response. The power of prompting can also be utilized to develop Janet's components. For example, the LLM can be used to transform unstructured text into a structured form like a graph. So, the LLM can be incorporated into the knowledge creation and organization.

Another observation is that implementing RLHF requires hiring crowd workers. In order to avoid this, we plan to put the burden of collecting feedback on the users of Janet once the aforementioned enhancements are implemented. This feedback is crucial for our continuous improvement pipeline.

VI. Conclusion

In this paper, we introduced Janet, our prototype of a conversational agent for VREs. We highlighted our approach in implementing it introducing its general architecture and showcasing its current limitations. In particular, we built our conversational agent leveraging one possible solution to the problem of hallucination of LLMs. Namely, we exploited the existence of rich sources of information inside VREs to implement a RAG-based conversational agent. In doing so, we focused on building a set of components that can not only index and retrieve the VRE's textual content, but can also generate textual responses utilizing the retrieved content. The prototype has been tested on a small scale which showcased some of its limitations which serve as a guide to our future enhancements. This prototype serves as a first step in the process of building a conversational agent for VREs.

Acknowledgment

The work presented in this paper was supported by Blue-Cloud 2026 (grant agreement No. 101094227), Skills4EOSC (grant agreement No. 101058527), and SoBigData RI PPP (grant agreement No. 101079043) funded by the European Union's Horizon Europe.

References

[1] L. Candela, D. Castelli, and P. Pagano, "Virtual Research Environments: An Overview and a Research Agenda," *Data Science Journal*, vol. 12, no. 0, pp. GRDI75–GRDI81, 2013. [Online]. Available: http://datascience.codata.org/articles/abstract/10.2481/dsj.GRDI-013/

[2] ——, "The D4Science Experience on Virtual Research Environments Development," *Computing in Science & Engineering*, pp. 1–9, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10167494/

[3] M. Assante, L. Candela, D. Castelli, R. Cirillo, G. Coro, L. Frosini, L. Lelii, F. Mangiacrapa, P. Pagano, G. Panichi, and F. Sinibaldi, "Enacting open science by D4Science," *Future Generation Computer Systems*, vol. 101, pp. 555–563, Dec. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X1831464X

[4] K. A. Lawrence, M. Zentner, N. Wilkins-Diehr, J. A. Wernert, M. Pierce, S. Marru, and S. Michael, "Science gateways today and tomorrow: positive perspectives of nearly 5000 members of the research community," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 16, pp. 4252–4268, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3526

[5] M. Arezoumandan, L. Candela, D. Castelli, A. Ghannadrad, D. Mangione, and P. Pagano, "Virtual Research Environments Ethnography: a Preliminary Study," in *Proceedings of the 14th International Workshop on Science Gateways, Trento, Italy*, 2022.

[6] A. S. T. Ibrahim and L. Candela, "Conversational agents for virtual research environments: a survey of the literature," ISTI Technical Report ISTI-2023-TR/007, 2023. [Online]. Available: https://doi.org/10.32079/ISTI-TR-2023/007

[7] S. S. Poon, R. C. Thomas, C. R. Aragon, and B. Lee, "Context-Linked Virtual Assistants for Distributed Teams: An Astrophysics Case Study," in *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 361370. [Online]. Available: https://doi.org/10.1145/1460563.1460623

[8] L.-K. Lee, Y.-C. Fung, Y.-W. Pun, K.-K. Wong, M. T.-Y. Yu, and N.-I. Wu, "Using a multiplatform chatbot as an online tutor in a university course," in *2020 International Symposium on Educational Technology (ISET)*, 2020, pp. 53–56.

[9] D. Griol, A. Sanchis, J. M. Molina, and Z. Callejas, "Developing enhanced conversational agents for social virtual worlds," *Neurocomputing*, vol. 354, pp. 27–40, 2019, recent Advancements in Hybrid Artificial Intelligence Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219304576

[10] T.-H. K. Huang, J. C. Chang, and J. P. Bigham, "Evorus," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, apr 2018. [Online]. Available: https://doi.org/10.1145%2F3173574.3173869

[11] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[12] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize with human feedback," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 3008–3021.

[13] J. Gao, C. Xiong, P. Bennett, and N. Craswell, *Neural Approaches to Conversational Information Retrieval*. Springer Cham, 2023.

[14] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474.

[15] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," 2020. [Online]. Available: https://arxiv.org/abs/2007.01282

[16] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Machine Learning with Applications*, vol. 2, p. 100006, 2020.

[17] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[18] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[19] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[21] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "MS MARCO: A human generated machine reading comprehension dataset," *CoRR*, vol. abs/1611.09268, 2016. [Online]. Available: http://arxiv.org/abs/1611.09268

[22] Q. Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu, "Pubmedqa: A dataset for biomedical research question answering," 2019.

[23] P. Dasigi, K. Lo, I. Beltagy, A. Cohan, N. A. Smith, and M. Gardner, "A dataset of information-seeking questions and answers anchored in research papers," 2021.

[24] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan, "Learn to explain: Multimodal reasoning via thought chains for science question answering," in *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[25] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano, "Learning to summarize from human feedback," in *NeurIPS*, 2020.

[26] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," *ArXiv*, vol. abs/1808.08745, 2018.

[27] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, "Personalizing dialogue agents: I have a dog, do you have pets too?" *ArXiv*, 2018.