Istituto di Scienza e Tecnologie
dell'Informazione "A. Faedo"
Consiglio Nazionale delle Ricerche

ISTI Technical Reports

Conversational agents for virtual research environments: a survey of the literature

Ahmed Salah Tawfik Ibrahim, ISTI-CNR, Pisa, Italy
Leonardo Candela, ISTI-CNR, Pisa, Italy

ISTI-TR-2023/007

Conversational agents for virtual research environments: a survey of the literature

Ahmed Salah Tawfik Ibrahim, Leonardo Candela

ISTI-TR-2023/007

Conversational artificial intelligence is becoming a rather hot topic in the field of artificial intelligence especially after the release of ChatGPT by OpenAI. That is why it is useful to conduct a survey of the literature regarding conversational agents in order to have an idea about the techniques used to develop such systems. Therefore, the aim of this report is to aggregate and summarize the previous research efforts in order to provide insight that would aid future researchers. In addition, the survey was conducted within the context of developing Janet, a conversational agent for virtual research environments; thus, it can be used as a starting point for similar projects.

Keywords: Virtual Assistant, Virtual Research Environment, Chatbot.

Citation
Ibrahim A.S.T., Candela L. *Conversational agents for virtual research environments: a survey of the literature*. ISTI Technical Reports 2023/007. DOI: 10.32079/ISTI-TR-2023/007.

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"
Area della Ricerca CNR di Pisa
Via G. Moruzzi 1
56124 Pisa Italy
http://www.isti.cnr.it

Technical Report, June 30, 2023
Consiglio Nazionale delle Ricerche - Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"

Conversational Agents for Virtual Research Environments: A Survey of the Literature

Ahmed Salah Tawfik Ibrahim, Leonardo Candela*

Abstract
Conversational artificial intelligence is becoming a rather hot topic in the field of artificial intelligence especially after the release of ChatGPT by OpenAI. That is why it is useful to conduct a survey of the literature regarding conversational agents in order to have an idea about the techniques used to develop such systems. Therefore, the aim of this report is to aggregate and summarize the previous research efforts in order to provide insight that would aid future researchers. In addition, the survey was conducted within the context of developing Janet, a conversational agent for virtual research environments; thus, it can be used as a starting point for similar projects.

Keywords
Virtual Assistant — Virtual Research Environment — AI Chatbot

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", Consiglio Nazionale delle Ricerche, Via G. Moruzzi 1, 56124, Pisa, Italy
*__Corresponding authors__: leonardo.candela@isti.cnr.it

Contents

1. Introduction

Conversational agents, also referred to as virtual assistants or chatbots, are now a default service offered by most websites and companies. They are used to fulfill a wide range of needs depending on the sector where they are deployed. The capabilities of these conversational agents are always improving thanks to the recent advances in the field of natural language processing (NLP) which aims at making machines able to understand and/or generate human language.

In the past, NLP techniques were just a collection of statistical methods that were used to perform a set of limited tasks. Over the years, and with the introduction of deep learning, new models were introduced and they outperformed the existing statistical methods. These models include recurrent neural networks (RNN) and its variants like long-short-term memory networks (LSTM) and gated recurrent units (GRU). However, the real breakthrough happened in 2018 with the introduction of BERT [11] which is a transformer model that was able to outperform all the existing models on many NLP tasks. After BERT, other transformer models were developed and tasks, that were once thought of as impossible, are now possible thanks to transformers.

In the field of information retrieval, as an example, thanks to the representations of queries by these large language models, it is now possible to retrieve relevant content with high accuracy in a very short time. Consensus[1] is a service that retrieves evidence from scientific papers. It basically extracts the most relevant sentences to the user's query from the papers and presents them to the user.

In the field of language generation, however, OpenAI has released ChatGPT[2] in the end of 2022. It is basically a large language model, based on the most recent GPT transformer model, that is capable of engaging in a conversation with the user about virtually anything. It can answer questions, write code, summarize text, translate between languages and many other tasks by engaging in a conversation with the user. Using that technology, many task-specific initiatives were born. For instance, ChatPDF[3] is a service that makes use of GPT 3.5 to perform the task of machine comprehension on a given PDF file. It can then answer questions related to that PDF file by extracting the most relevant paragraph and then using ChatGPT to generate the answer. Another service that was developed is Jenni[4] which aims at easing the process of writing using artificial intelligence. Again, it makes use of ChatGPT in order to perform a set of tasks like paraphrasing and autocompletion.

In summary, thanks to the current advances in NLP, specifically thanks to the emergence of transformer models, it is now possible to build useful applications that exploit these advances, as shown in the mentioned examples.

[1] https://consensus.app/
[2] https://openai.com/blog/chatgpt
[3] https://www.chatpdf.com/
[4] https://jenni.ai/

However, transformers are merely a component in these complex systems and there are various techniques and considerations to be made when developing conversational agents. Therefore, in the coming sections, we will try to cover the most important aspects regarding conversational agents and their development.

This report is organized as follows. Sec. 2 provides a theoretical overview of the development of conversational agents without delving into the technical details. Sec. 3 describes some of the main techniques and practical considerations to be made when developing conversational agents. Finally, Sec. 4 summarizes the main findings and provides a set of questions to be answered when designing a conversational agent.

2. Theoretical Overview

2.1 Conversational Agents

A conversational agent is simply an application that takes input from the user, analyzes that input and provides an answer or performs an action depending on the context in which it is being used [29]. In other words, it is supposed to mimic a human conversation with the purpose of providing assistance to the user with the task at hand [23].

Conversational agents are being used in many domains such as health-care and business with different purposes like providing information, supporting users or executing actions [23].

Conversational agents can be categorized based on different aspects. For example, they could be open-domain, closed-domain or cross-domain regarding the domain of knowledge they cover [1]. Another classification can be regarding their goal as they can be used to retrieve information, have a chat or perform a task [1].

The implementation of conversational agents has undergone several developments over the years. It started with the rule-based implementations where the task of interest is predefined and the conversation flow is more or less predetermined; however, with the rise of machine learning and deep learning techniques, a new technique was introduced based on machine learning (ML-based) that did not have a predefined conversation flow [22]. Both techniques have their strengths and weaknesses. Rule-based conversational agents are easy to implement but very difficult to maintain and generalize and so they work best when the task of interest is well-defined and limited whereas ML-based conversational agents are difficult to develop and train but if a large enough corpus is available, they generalize better and they perform better in open-domain tasks [23]. It is worth mentioning that rule-based conversational agents usually make use of a special language to describe the rules and patterns like the Artificial Intelligence Markup Language (AIML) or others.[1]. ML-based conversational agents, however, do not require such languages as they do not require patterns to be specified apriori. Instead, they are usually trained to either retrieve the response or generate it utilizing Natural language processing (NLP) and understanding (NLU) in doing so.[1].

When it comes to developing conversational agents, there are various tools that include libraries, frameworks, platforms and services [24]. The choice of the tool highly depends on the requirements in terms of deployment, dialog management and integration as these aspects, among others, differ from one tool to another. The general architecture (Figure 1) of a conversational agent consists of multiple components but mainly it must include an interface to get the input via text or speech, a message analyzer to understand the input, a dialog manager, a knowledge layer to find the answer to the query and finally a response generator [1].

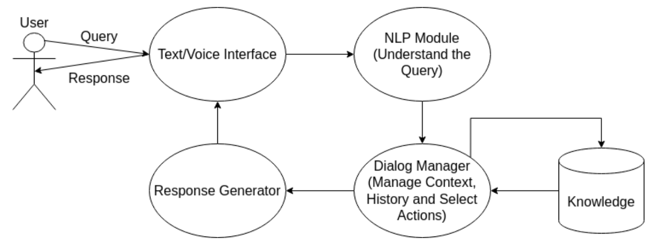

Figure 1. General Architecture of Conversational Agents

The dialog manager is the brain of the conversational agent as it keeps track of the dialog state and history and decides the next action to take based on them [7]. The dialog manager can be modeled in different ways that include deep learning models like neural networks and statistical models like hidden Markov models [7].

Conversational agents are very versatile and can be used in different environments. One conversational agent based on voice was developed to offer support and knowledge to illiterate farmers [17] while another was situated at an airport terminal to assist travelers with various needs [6]. It has even been developed to help workers' productivity in the workplace by reminding them to take breaks and helping them avoid distractions [2].

This versatility of conversational agents has led to coming up with a classification of conversational agents based on the role they play within a specific environment. Bittner et al. provide in their work three different categories of conversational agents: facilitators, peers and experts [4]. They observed that facilitators are usually deployed within closed-domains where they assist a person or a team with a certain task that has a well known complex model in the system and so they have a proactive behavior. Expert conversational agents, on the other hand, act reactively upon the user's request. Peers, however, are somewhere in between as they behave both proactively and reactively by both providing answers to queries and recommendations. This categorization helps with making design decisions when developing a conversational agent. For example, the socio-emotional behaviors of a conversational agent highly depend on its role as peers should show a level of emotional engagement unlike experts which have to just provide helpful answers. Furthermore, based on whether or not the application domain is open or closed, one can decide which role the conversational agent can play. Peers can work both in closed and open domain applications while facilitators usually cannot as they are based on sophisticated modeling of the task.

2.2 Conversational Agents in Collaborative Environments

Seeber et al. describe in their paper a research agenda that should be considered when developing a conversational agent to become part of a team in a collaborative environment [28]. They outlined three main aspects to be considered when designing a conversational agent for collaborative purposes. The first aspect is the machine artifact which includes appearance, conversation, architecture and knowledge processing among other things. The second one is concerned with the collaboration itself in terms of team, task and process design. The last aspect is the institution design in terms of responsibilities and liabilities in addition to human training. These three aspects describe a very general methodology for approaching the problem of designing a conversational agent for a collaborative environment.

Bert is an example of a conversational 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 [25]. It plays the role of a peer according to the taxonomy presented earlier and that's why it has emotional features as it can tell jokes and react to jokes.

Another example is InfoBot which is an online tutor for university students which is deployed within their collaborative environment and it helps with answering questions related to a course's materials or logistics and it even offers a sort of assessment to the understanding of the student via quizzes [19].

Another conversational agent, Demic, 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 [13]. These are examples of conversational agents that have been developed for a specific purpose within a collaborative environment.

Virtual research environments (VREs), despite being an example of a collaborative environment, are different from the traditional ones. This is because a traditional collaborative environment is usually conceived to solve a certain well-defined task whereas each VRE is conceived to serve the needs of a community of practice where the task definition may not be specified apriori.

A possible solution to this problem is having a human teach the conversational agent how to perform the tasks of interest as shown in [3] where basically the conversational agent, PLOW, is supposed to either keep track of the actions the user explicitly records regarding a certain task or keep track of the user actions on the browser regarding the task and learn from those action traces how to perform the task.

A similar solution has been proposed by Hancock et al. where the conversational agent was trained for three auxiliary tasks to improve itself over time: (*i*) dialogue, (*ii*) satisfaction, and (*iii*) feedback [14]. Basically, for the dialogue task, they collected pairs of context and response whose sources were either a dataset or the human-bot interaction after deploying the conversational agent. As for the satisfaction task, they collected pairs of context and satisfaction score which they obtained via crowdsourcing in order to train a model that is able to predict when it may have made a mistake by predicting a satisfaction score whose value is then used for the third task; the feedback. Basically, in the feedback task, the bot asks the user to provide a better response if the estimated satisfaction score was below a threshold and then collect these context and feedback pairs to improve itself.

Building upon the idea of improving the conversational agent over time, Evorus [15] 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.

Another problem that may arise with VREs, however, is that the volume of the VRE content, which is made up of papers, datasets, posts among other things, is huge and to have an all-knowing conversational agent poses a challenge. A solution to this problem has been proposed by Zhang et al. where they model topics as a distribution over words and tools or datasets as distributions over topics in their Domain-specific Topic Model [32]. The distributions are learned by means of unsupervised learning and Markov chain Monte Carlo inference algorithm. This way of modelling allows a conversational agent to offer useful recommendations for scientists in terms of tools and datasets they can use in their research. The same team has developed a recommender conversational agent, Vidura, that has at its core a Mamdani fuzzy rule system that offers useful help to neuroscientists based on their proficiency level in both computing and neuroscience [8].

Another example of a conversational agent in a science gateway is GeCoAgent which is used in a biology gateway with the aim of supporting biologists with data exploration, analysis and visualization tasks [10]. It does so by modelling the processes carried out by biologists normally as tasks and functions and it leverages upon this model by creating a finite state automaton that is used to control the flow of a conversation between a biologist and the conversational agent.

3. Technical Overview

In this section, we present an overview of various technical aspects that were found useful in implementing conversation-based systems. In particular, we start by discussing transformer models as they are the basis of most, if not all, modern conversational systems due to their outstanding performance in NLP tasks. Then, we discuss the topic of conversational information retrieval which introduces a set of challenges to consider. These challenges include retrieval-augmented language generation, query disambiguation and context awareness. Finally, we had to include reinforcement learning, with an emphasis on reinforcement learning from human feedback, thanks to its proven power in improving the performances of existing models in NLP tasks.

3.1 Transformers

Before 2018, state of the art NLP models were based on recurrent neural networks such as long-short-term memories and gated recurrent units until Vaswani et al. introduced their groundbreaking transformer model in their paper attention is all you need [31]. They kept the general encoder-decoder architecture employed by the recurrent neural networks; however, they got rid of the recurrent nature of the neural network as they proved in their paper that the attention mechanism is what is needed [31].

Attention, as they describe it, is a way to compute a representation of a sequence by considering the relations between the different positions of the sequence [31]. More specifically, it maps the sequence and a set of key-value pairs represented by vectors into another output sequence. The output sequence is computed by a weighted sum over the values where the weights represent the importance of a given value for the output. In doing so, the weight of each value is computed by a function of the query and they key of that value.

By building up their encoder-decoder model using the aforementioned attention mechanism, they managed to build a simpler model that outperformed the existing state of the art models back then on most NLP tasks, becoming the new state of the art model itself [31].

Ever since the publication of that paper, different transformer models have been developed and they vary in their general architecture. For example, BERT [11] is an encoder-only model that is very suitable for computing representations of sequences. GPT [26] is another transformer model that is decoder-only and is mainly used for language generation. We also have T5 [27] which is an encoder-decoder model. Some of these models are widely used in classification tasks, like BERT while others are used for language generation tasks like T5 and the GPT family.

3.2 Conversational Information Retrieval Systems

A VRE has the potential of containing volumes of data items, like papers and datasets with their metadata written in natural language. Furthermore, the content of the papers is also written in natural language. The VRE also contains posts by the users which are also written in natural language. Hence, the conversational agent within such a VRE will have to be capable of performing the task of information retrieval in order to allow the user to exploit the resources easily as this capability is useful when performing tasks like question-answering, search or recommendation.

As stated earlier, an ML-based conversational agent could be used to retrieve information that satisfy the user's expressed need. Therefore, it is useful to consider conversational information retrieval (CIR) systems in the context of developing a conversational agents for scientists. CIRs are different from traditional IR systems as they leverage the dialogues between the user and the agent where the agent may play a proactive role in order to satisfy the user's need [12]. These systems nowadays make use of advanced language models in order to efficiently answer the user's query. Leveraging upon the general architecture of a conversation agent, a CIR system is made up of the components shown in figure 2 [12].

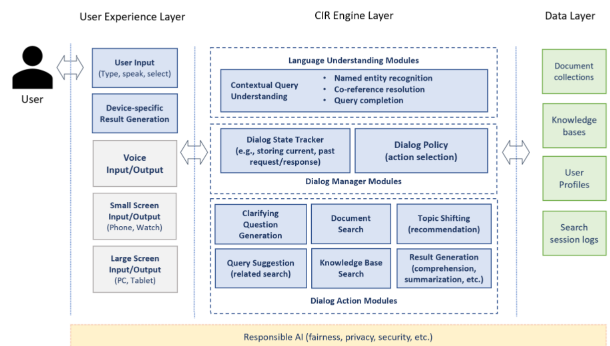

Figure 2. CIR System General Architecture by Gao et al. [12]

Basically, the message analyzer performs a set of actions in order to understand the query. These actions include resolving co-references and recognizing named entities [12]. The dialog manager tracks the state of the conversation and chooses the next action to take from a set of actions a CIR system is supposed to realize. These actions include search, recommendation, query disambiguation and result generation [12]. The knowledge is contained within knowledge bases, document collections and/or user profiles and logs [12].

In the following subsections, an overview of the techniques used to realize the main tasks of a CIR system is presented.

Retrieval-Augmented Language Generation

Retrieval-augmented language generation 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 Gao et al., in order to implement this task, a document retriever and an answer generator should be implemented [12] as shown in Figure 3.

The retriever is typically a neural network used to compute representations of text in such a way that makes relevant texts close to each other in the representation space. It is then used to perform a similarity search over the representation of a query and the representations of the collection of contexts in order to retrieve the most relevant text to the query. After

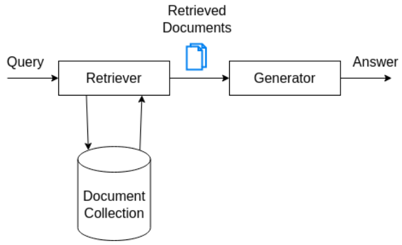

Figure 3. Retriever-Generator Model

retrieving the most relevant context, it will be passed along with the query to the generator to create the answer.

The generator is a conditional sequence-to-sequence model that generates an answer to a question given a context. As shown by Lewis et al. [20] and Izacard and Grave [16], 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 would be used to decode this combined representation into an answer by training it to minimize the sequence to sequence loss between the decoded answer and the reference answer [20, 16]. Figure 4 illustrates the way the generator works.

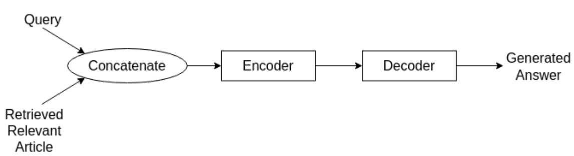

Figure 4. Generator Components

Handling Ambiguity

When dealing with conversational systems, ambiguity becomes one of the key points to consider. This is because ambiguity is embedded within human language; therefore, it is important to take it into account when designing conversational systems.

As defined by Keyvan and Huang [18], ambiguity occurs when the system is unable to understand the intent behind a given query with enough confidence. This can be due to a number of factors including when the query depends on the conversation history or when the user poorly forms their query. Because of this variety in causes, different approaches exist in the literature for handling ambiguous queries which can be grouped into three main approaches; query rewriting, question suggestion and asking for clarification [18].

Asking for Clarification It is recommended to resort to the technique of asking a clarifying question when the system can return multiple responses to the user's query [18]. This technique can be implemented in different ways which include clarifying question retrieval and clarifying question generation [18]. However, usually there is a classification step first which decides if a given query requires asking a clarifying question followed by another step where the clarifying question is retrieved or generated [18].

Query Rewriting The technique of query rewriting is very useful when the query can be modified, either by using lexical rules or by using the conversation history, in order to make it clear [18]. This can be done via stemming in order to reduce the mistakes in the user's input, query expansion by adding terms to the query that could enhance it or query substitution where co-references are resolved by what they refer to based on the chat history [18]. Deep learning techniques are mainly used to achieve this technique where the query and the chat history are input to a sequence to sequence model whose job is to modify the query based on the chat history if needed [18].

Question Suggestion Question suggestion is a technique that involves asking the user a relevant question to their query which may help them better express their need [18]. It is especially useful in cases where the original query contains errors, is formed using few terms or when the suggested question is an unambiguous form of that original query [18].

Context-Awareness in Conversational Agents

Context-Awareness in conversational agents means that the agent is able to contextualize a given query according to the conversation history with the user. In other words, it is a process that takes as input the contextual query along with the history and modifies the query according to the history in order to come up with a decontextualized query that can be used directly without the history to form a satisfying answer [12]. In essence, it is very similar to the query rewriting technique explained in the previous section.

There are two main techniques for the task of contextualizing a query: namely, heuristics for query expansion and neural query rewriting [12].

The heuristics include extracting keywords that could correspond to topics and subtopics from the history, then giving the query an ambiguity score in order to determine if it is ambiguous or not [12]. These heuristics make use of retrieval scores when the system is retrieval-based. The keyword extractor assigns scores to each term based on the retrieval score of that term with respect to the relevant documents to the query; i.e., the higher that score is, the more important the term is, the more likely it is for the term to be a subtopic or a topic [12]. Similarly, the ambiguity score of the query is calculated based on the retrieval score of the query with respect to the most relevant document; i.e., the higher that score is, the less likely it is that the query is ambiguous [12].

The neural query rewriting, as mentioned earlier, is essentially a generative sequence-to-sequence model that is trained to generate a modified query based on the conversation history [12]. Nowadays, they are implemented using transformers as they beat the previous sequence-to-sequence models in terms of accuracy and performance [12].

Reinforcement Learning

Reinforcement learning, in simple words, is a form of learning that involves an agent interacting with an environment according to a certain policy which gets updated taking into account the observations collected by the agent from that environment as shown in figure 5.

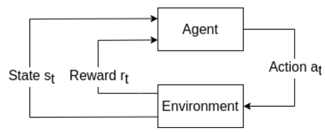

Figure 5. Reinforcement Learning Illustration by Buffet et al. [5]

Mathematically speaking, it is a Markov decision process defined over a state space and an action space with a scalar reward function and a transition function. The transition function is a probability distribution that calculates the probability of transitioning into state s' from state s after taking action a. The reward function, however, calculates a scalar value representing the reward of taking action a while in state s. The policy of the agent is simply a mapping from a state into an action. Reinforcement learning involves solving a Markov decision process which means finding the policy that maximizes the discounted sum of future rewards. In other words, it does not just consider the immediate gain from taking an action in a given state as it takes into consideration the future gain as well discounted by a scalar factor.

There exists different techniques for reinforcement learning which are beyond the scope of interest for this work; however, what is particularly interesting for the purposes of this work is how the policy and the reward can be viewed in the context of language modelling and conversational agents.

According to Li et al. [21], a model that generates natural language, like any sequence to sequence or encoder-decoder, is simply the policy which an agent uses to generate language. They argue in their paper that such models usually are trained to minimize an arbitrary loss function, like the maximum-likelihood estimate (MLE), which does not reflect actual reward. So, they define their own custom reward function which they try to maximize using reinforcement learning. Everything else is straight-forward; an action is simply the outpt of the model; a state is defined as the input to the model and the policy is nothing but the parameters of the model which are initialized to the values obtained from pre-training with MLE loss. They use the policy gradient method to optimize the policy and they observed that they got better results in terms of output diversity, dialogue length and human satisfaction with the output for long conversations. This work is particularly interesting as it ushered into a promising direction which is reinforcement learning from human feedback discussed below.

Reinforcement Learning from Human Feedback

Reinforcement learning from human feedback (RLHF) involves modelling the reward function and then using that model to optimize the policy [9]. To achieve this, the agent tries to learn a policy that is preferred by a human observer. In doing so, both the agent policy and the reward function are 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 function network takes these comparisons and fits them in order to produce a reward model 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 it.

This paradigm has been used by Stiennon et al. [30] to learn a model to summarize texts. They simply used a pre-trained language model for generating summaries which was trained using supervised learning on an annotated dataset with texts and their corresponding summaries. Then, they provided human judges with pairs of summaries that were sampled from various sources including the pre-trained language model in order to let the human decide the summary they prefer. Using this dataset, they trained a neural network with the purpose of coming up with a reward model based on human preferences. Finally, they use this reward model in order to optimize the policy, which is simply the language model, using reinforcement learning. The results they found were that further optimizing a language model with human feedback outperforms the language model resulting from supervised training only.

4. Conclusion and Insights

In conclusion, we have shown the main technical and theoretical considerations to be made when designing a conversational agent in general and within a collaborative environment, like a VRE, in particular.

In general, if the interactions are limited and can be predefined, then a rule-based approach would be the better option while if the interactions cannot be predicted apriori, then the machine learning approaches are the reasonable choice.

Then, one needs to decide the role the conversational agent would play within a collaborative setting in order to make the suitable design decisions; for example, if the agent is supposed to be a peer, then it should be equipped with emotional intelligence capabilities that would allow it to be human-like in its interactions.

Furthermore, the designers should decide if ambiguity and context-awareness are to be accounted for and to accordingly choose the appropriate techniques to resolve ambiguity and account for context as they have significant effects on the performance of the agent.

It is also vital to take into consideration how knowledge is organized and retrieved. If knowledge is organized into a document vector collection, then a neural retriever is the reasonable choice.

Finally, designers need to choose the suitable response generation mechanism and use the appropriate technique to implement it. For instance, if the response is to be generated via generative models, then the designers need to decide whether to finetune a language model, like a transformer, or use reinforcement learning to enhance a language model or both.

Acknowledgments

This work received funding from the European Union's Horizon 2020 research and innovation programme under: Blue Cloud project (grant agreement No. 862409).

Author contributions

According to CRediT taxonomy, authors contributed as follows. Conceptualization, Writing – original draft, Writing – review & editing: ASTI and LC; Data curation, Software: ASTI. Funding acquisition, Supervision: LC.

References

[1] E. Adamopoulou and L. Moussiades. Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006, 2020. ISSN 2666-8270. doi: https://doi.org/10.1016/j.mlwa.2020.100006. URL https://www.sciencedirect.com/science/article/pii/S2666827020300062.

[2] S. Ahire, M. Rohs, and S. Benjamin. Ubiquitous Work Assistant: Synchronizing a Stationary and a Wearable Conversational Agent to Assist Knowledge Work. In *2022 Symposium on Human-Computer Interaction for Work*, CHIWORK 2022, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450396554. doi: 10.1145/3533406.3533420. URL https://doi.org/10.1145/3533406.3533420.

[3] J. Allen, N. Chambers, G. Ferguson, L. Galescu, H. Jung, M. Swift, and W. Taysom. Plow: a collaborative task learning agent. pages 1514–1519, 01 2007.

[4] E. Bittner, S. Oeste-Reiß, and J. M. Leimeister. Where is the Bot in our Team? Toward a Taxonomy of Design Option Combinations for Conversational Agents in Collaborative Work. 2019. doi: 10.24251/HICSS.2019.035.

[5] O. Buffet, O. Pietquin, and P. Weng. Reinforcement learning, 2020. URL https://arxiv.org/abs/2005.14419.

[6] M. Carisi, A. Albarelli, and F. L. Luccio. Design and Implementation of an Airport Chatbot. In *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, GoodTechs '19, page 49–54, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362610. doi: 10.1145/3342428.3342664. URL https://doi.org/10.1145/3342428.3342664.

[7] P. Cañas, D. Griol, and Z. Callejas. Towards versatile conversations with data-driven dialog management and its integration in commercial platforms. *Journal of Computational Science*, 55:101443, 2021. ISSN 1877-7503. doi: https://doi.org/10.1016/j.jocs.2021.101443. URL https://www.sciencedirect.com/science/article/pii/S1877750321001253.

[8] A. A. Chandrashekara, R. K. M. Talluri, S. S. Sivarathri, R. Mitra, P. Calyam, K. Kee, and S. Nair. Fuzzy-based conversational recommender for data-intensive science gateway applications. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4870–4875, 2018. doi: 10.1109/BigData.2018.8622046.

[9] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences, 2017. URL https://arxiv.org/abs/1706.03741.

[10] P. Crovari, S. Pidò, P. Pinoli, A. Bernasconi, A. Canakoglu, F. Garzotto, and S. Ceri. GeCoAgent: A Conversational Agent for Empowering Genomic Data Extraction and Analysis. *ACM Trans. Comput. Healthcare*, 3(1), oct 2021. ISSN 2691-1957. doi: 10.1145/3464383. URL https://doi.org/10.1145/3464383.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[12] J. Gao, C. Xiong, P. Bennett, and N. Craswell. Neural approaches to conversational information retrieval, 2022. URL https://arxiv.org/abs/2201.05176.

[13] D. Griol, A. Sanchis, J. M. Molina, and Z. Callejas. Developing enhanced conversational agents for social virtual worlds. *Neurocomputing*, 354:27–40, 2019. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2018.09.099. URL https://www.sciencedirect.com/science/article/pii/S0925231219304576. Recent Advancements in Hybrid Artificial Intelligence Systems.

[14] B. Hancock, A. Bordes, P.-E. Mazaré, and J. Weston. Learning from dialogue after deployment: Feed yourself, chatbot!, 2019. URL https://arxiv.org/abs/1901.05415.

[15] 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. doi: 10.1145/3173574.3173869. URL https://doi.org/10.1145%2F3173574.3173869.

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

[17] M. Jain, P. Kumar, I. Bhansali, Q. V. Liao, K. Truong, and S. Patel. FarmChat: A Conversational Agent to Answer Farmer Queries. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(4), dec 2018. doi: 10.1145/3287048. URL https://doi.org/10.1145/3287048.

[18] K. Keyvan and J. X. Huang. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Comput. Surv.*, 55(6), dec 2022. ISSN 0360-0300. doi: 10.1145/3534965. URL https://doi.org/10.1145/3534965.

[19] 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)*, pages 53–56, 2020. doi: 10.1109/ISET49818.2020.00021.

[20] 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, 2020. URL https://arxiv.org/abs/2005.11401.

[21] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky. Deep reinforcement learning for dialogue generation, 2016. URL https://arxiv.org/abs/1606.01541.

[22] W. Maeng and J. Lee. Designing a chatbot for survivors of sexual violence: Exploratory study for hybrid approach combining rule-based chatbot and ml-based chatbot. In *Asian CHI Symposium 2021*, Asian CHI Symposium 2021, page 160–166, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450382038. doi: 10.1145/3429360.3468203. URL https://doi.org/10.1145/3429360.3468203.

[23] Q. Motger, X. Franch, and J. Marco. Software-Based Dialogue Systems: Survey, Taxonomy and Challenges. *ACM Comput. Surv.*, 2022. ISSN 0360-0300. doi: 10.1145/3527450. URL https://doi.org/10.1145/3527450.

[24] S. Perez-Soler, S. Juarez-Puerta, E. Guerra, and J. de Lara. Choosing a chatbot development tool. *IEEE Software*, 38 (4):94–103, 2021. doi: 10.1109/MS.2020.3030198.

[25] 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*, CSCW '08, page 361–370, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580074. doi: 10.1145/1460563.1460623. URL https://doi.org/10.1145/1460563.1460623.

[26] A. Radford and K. Narasimhan. Improving language understanding by generative pre-training. 2018.

[27] 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*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[28] I. Seeber, E. Bittner, R. O. Briggs, T. de Vreede, G.-J. de Vreede, A. Elkins, R. Maier, A. B. Merz, S. Oeste-Reiß, N. Randrup, G. Schwabe, and M. Söllner. Machines as teammates: A research agenda on ai in team collaboration. *Information Management*, 57(2):103174, 2020. ISSN 0378-7206. doi: https://doi.org/10.1016/j.im.2019.103174. URL https://www.sciencedirect.com/science/article/pii/S0378720619303337.

[29] S. Singh and H. K. Thakur. Survey of Various AI Chatbots Based on Technology Used. In *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 1074–1079, 2020. doi: 10.1109/ICRITO48877.2020.9197943.

[30] 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, 2020. URL https://arxiv.org/abs/2009.01325.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[32] Y. Zhang, P. Calyam, T. Joshi, S. Nair, and D. Xu. Domain-specific topic model for knowledge discovery through conversational agents in data intensive scientific communities. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4886–4895, 2018. doi: 10.1109/BigData.2018.8622309.