Prompting GPT-based Dialogue Agents with Knowledge Graphs for Consistent Personality

Anonymous ACL submission

Abstract

This paper introduces a novel approach for grounding GPT-based dialogue agents using knowledge graphs (KGs) to enhance consistency, minimize hallucinations, and improve response speed for real-time spoken dialogue systems. We evaluate our KG-grounded model against four baselines: a persona-grounded model, a relevance-based classifier, a retrieval-augmented generation (RAG) model, and a random baseline. Experimental results show that the KG-grounded model outperforms most baselines in terms of response time and reduced hallucination rates while maintaining similar response relevance as other generative approaches. By integrating structured knowledge from KGs, our method significantly advances the development of reliable, context-aware conversational agents for real-time applications, making it a strong candidate for future spoken dialogue systems.

1 Introduction

Dialogue agents have become ubiquitous in various industries, serving as virtual assistants (Harms et al., 2018) (Campagna and Ramesh, 2017), customer service representatives (Paikens et al., 2020), and companions in everyday interactions (Webb et al., 2010). As these agents continue to evolve, ensuring their ability to engage users in natural and coherent conversations remains a challenge. This paper addresses this challenge and focuses on the development of one-on-one dialogue agents tailored for chit-chat and closed domain question answering scenarios.

Unlike generic dialogue systems, our use-case involves dialogue agents that are imbued with the persona of a fixed character, aimed at enhancing the conversational experience by infusing a consistent personality into their interactions. This approach not only fosters a sense of familiarity and rapport between the user and the agent but also contributes to the overall naturalness and coherence of the conversation. This approach can be used for role-play training (Kenny et al., 2007), education (Swartout et al., 2010), and culture preservation (Traum et al., 2015). Content has been authored by skilled writers (Swartout et al., 2010) or taken from natural interviews (Traum et al., 2015).

Previous studies (Leuski and Traum, 2011) (Pal et al., 2023) have demonstrated the efficacy of statistical models employing cross-language relevance to select appropriate response content from the previous utterance and dialogue context. These models excel in selecting relevant responses from a curated set of responses. Such an approach not only confines the conversation within relevant topic domains but also guarantees a consistent personality throughout the interaction. However, this method's reliance on a predetermined set of responses can lead to a reduction in response diversity, potentially resulting in repetitive and unnatural conversational exchanges.

Generative models like GPT (Brown et al., 2020) offer a solution to the limitations of fixed response set models, as they generate responses dynamically. However, they face the challenge of producing responses that are factually incorrect or incongruent with the character's personality—a phenomenon known as 'hallucination' (Ji et al., 2023). In our research, we explore the task of grounding GPT-based dialogue agents with knowledge graphs (Hogan et al., 2021) to ensure the consistent portrayal of the assigned character's personality traits throughout the conversation. By using knowledge graphs, we aim to augment the agent's comprehension of the context, preferences, and idiosyncrasies associated with the character, thereby fostering more authentic and compelling exchanges.

Our work has the following contributions:

- We propose a method for grounding GPT-

based dialogue agents using knowledge graphs, providing a structured framework for maintaining consistent personality and domain knowledge in conversational AI systems.

- We empirically compare the KG-grounded model (KGGPT) against baseline approaches (persona-grounded, relevance-based classifier, RAG and Random), focusing on response time, response relevance and hallucination rates in dialogue generation.

Results show that the KGGPT model has the fastest response time among all generative models, the lowest hallucination error rates, and a response relevancy comparable to other generative approaches. These findings suggest that, for certain applications, we can effectively augment authored content with generated material without compromising accuracy or personality consistency.

2 Related Work

Statistical retrieval approaches to responding in dialogue have been shown to perform well in domain constrained scenarios. NPCEditor (Leuski and Traum, 2010) is one common tool implementing a cross-language retrieval model that has been leveraged to develop and implement diverse dialogue agents trained on datasets spanning various domains. Notably, it has been employed in crafting Sgt. Blackwell (Leuski et al., 2006), a virtual soldier, which garnered widespread recognition, including display in the Cooper-Hewitt National Design Museum in New York, from December 2006 until July 2007, as part of the National Design Triennial. This agent is designed to disseminate information regarding his role in the military, share personal anecdotes, and engage in casual conversation to captivate the audience. The content was written by a screenwriter with familiarity with both the Army and the technology institute that created the character, and garned much interest in personality-related questions (Robinson et al., 2008). We use a dataset of prompts spoken by museum visitors to SFT Blackwell in our study.

GPT (Brown et al., 2020), a series of autoregressive large language models developed by OpenAI, has seen significant advancement with the introduction of ChatGPT (OpenAI, 2024), showcasing remarkable efficacy in crafting both general-purpose and task-specific dialogue agents. Nonetheless, while GPT excels in generating contextually appropriate responses, it often encounters issues with factual accuracy, leading to instances of information hallucination. This becomes particularly problematic in dialogue scenarios where the agent serves as a source of information, potentially disrupting user engagement and immersion if the model generates preferences and traits incongruent with the agent's intended personality.

One such attempt to provide a consistent personality is persona based grounding (Tang et al., 2021). In this approach, the character's personality traits and preferences are condensed into a biographical summary, which is then provided as an input prompt to the model. While effective for straightforward characters, this approach struggles to encompass all the nuanced details and preferences of more complex characters within the constraints of the prompt window. Thus, for more intricate characters like ours, we must summarize the biography content in an effort to make it compatible with the prompt window. As we later show, this step leads to omission of certain details which makes the model prone to hallucination errors.

Another prevalent method is called retrieval augmented generation (RAG) (Gao et al., 2024). In this approach, the information regarding the dialogue scope, in our case, the personality, life experiences and preferences of our agent are stored in an external database. Vector databases are often preferred for their similarity based search algorithms. Once the user asks a query, a database lookup is performed and the necessary information is fetched. This information is then passed into the model along with the user's query for response generation. RAG has shown promising results in reducing hallucination error rates. However, performing a database lookup via external APIs comes with an overhead that leads to an increase in time taken for response generation. In our scenario, we require the model to generate responses fast so as to simulate human-like conversation and not break audience immersion.

A knowledge graph (Hogan et al., 2021) is a tool used to structure data in the form of interconnected entities and their relationships. It uses triplets as its underlying data structure. A triple consists of three parts: a subject, an object and the relationship. The subject and object are entities,

while the relationship describes the connection of these entities. Previously, knowledge graphs have been used to create retrieval augmented generation (RAG) systems, however these systems too have an extra overhead due to the reasons mentioned above. Our approach uses knowledge graphs without having to store it in an external database or query it separately.

3 Models

In this section we describe the models used for the competitive analysis. We have compared five different approaches for picking a dialogue response. There are two selection approaches that choose from the available set of pre-authored responses. These are the NPCEDitor, using the approach deployed in the museum, and a random baseline that picks one of these answers at random. We have also compared three generative models that are guided by the authored material, but not limited to it. We describe these in more detail below.

3.1 NPCEditor

The NPCEditor introduced in Leuski and Traum (2010) utilizes cross language relevance modelling to select an appropriate response from a list of predefined responses. The first step in it's operation is the ingestion of the predefined set of responses. Upon ingestion, the NPCEditor creates a probability distribution vector for each response. The vectors are created using frequency modelling over the entire response vocabulary with Jelinek-Mercer smoothing.

The next step in it's operation is to use the user's query to build a conditional probability distribution vector over the response vocabulary, based on the query. Given a user's query $Q = q_1, q_2, ...q_n$, and the response vocabulary $|A|$, the conditional probability distribution vector can be defined as:

$$P(a|Q) = \frac{P(a, q_1, q_2, ...q_n)}{P(q_1, q_2, ...q_n)} \qquad \forall a \in |A| \quad (1)$$

Finally, once the conditional distribution has been generated, it is compared against individual probability distribution of the responses using KL Divergence. The responses are then ranked in order of similarity with the conditional distribution vector.

$$D(P_Q(A)||P(A)) = \sum_{a \in |A|} P(a|Q)\, log \frac{P(a|Q)}{P(a)} \quad (2)$$

The NPCEditor treats the user's query and responses as two separate languages. The motivation behind this approach is to account for the inherent differences in salient features between queries and responses. Queries are generally questions which have a higher probability of containing 'wh-' words like 'who', 'what', 'where', etc. Furthermore, queries do not contain much information but lay down the structure of the response. Responses on the other hand are much richer in information and have a more uniform distribution of words. Pal et al. (2023) describes the 'Questions v\s Answers Problem' and shows why treating questions and answers as separate languages works well for the NPCEditor. Furthermore, Leuski and Traum (2012) provides an in-depth explanation of the working of the NPCEditor.

3.2 Persona Grounded GPT

Tang et al. (2021) introduced the idea of grounding GPT based models using a set of facts that accurately describe the character's persona. The model takes in the character's persona as input along with the dialogue history and generates new responses that maintain consistency with previous responses. Tang et al. (2021) built their model on top of DialoGPT (Zhang et al., 2020), following the GPT 2 architecture (Radford et al., 2019). However, we decided to use this approach with GPT 3.5 (OpenAI, 2024). Since its release, GPT 3.5 has outperformed its predecessors (Brown et al., 2020) in multiple NLP tasks.

The first step in the process is to generate the character's persona information. In Zhang et al. (2018), the authors use five to six sentences to encapsulate the persona information. While such a small set of sentences works for simple characters, ours is a highly detailed one. Capturing all the idiosyncrasies and personality traits of Sgt. Blackwell is not possible in such a small number of sentences. Hence we use a character summarization step. The responses from the dataset are passed into a GPT-4o model and it is prompted to summarize the information with the following prompt:

Create a character summary for Sgt Blackwell, a virtual soldier in the 1-23rd regiment using the provided information. Do not lose out on any of the details and personality traits of the character. Do not introduce any opinions and preferences not mentioned in the provided information: < responses from dataset >

The entire dataset exceeds the maximum prompt size of the GPT model, hence multiple summaries are created with parts of the dataset. These summaries are then combined using the same prompt. Once the persona information is created after the summarizing step, the next step is to generate responses for the queries in the test dataset. The following prompt is used to generate responses:

You are Sergeant Blackwell. < Persona Information >. Generate a descriptive first person response to the utterance: < query >. Do not use any external information. If you cannot respond, say "Sorry. That's outside my Area of Operation."

The generated persona summary consists of 10,763 characters which translates to 1,933 tokens with the GPT-4o tokenizer.

3.3 Knowledge Graph Grounded GPT

While the NPCEditor performs well on the task of closed domain question answering and chit-chat, it lacks diversity in responses. This shortcoming is addressed in the persona grounded GPT model, however, it cannot capture every detail about the character due to information being lost in the persona summarization phase. Hence, we propose the Knowledge Graph grounded model.

The first step in setting up this model involves the creation of the knowledge graph. Typically, the generation of Knowledge Graphs involves three key stages: Entity Recognition, Triplet Extraction, and Entity Merging. Our dataset comprises of first-person conversational data in an interview-like fashion. We have a list of questions asked which are linked to a list of appropriate responses. Due to the conversational nature of the dataset, there is extensive usage of pronouns which impedes the entity recognition phase. Thus, we decided to use GPT-4o to automate the process of creating the knowledge graph instead of using traditional algorithms. The prompt has been described in Appendix A. The generated set of triplets has 8,481 characters which translates to 1,986 tokens with the GPT-4o tokenizer. To ensure a fair comparison, the size of the set of triplets is kept similar to the size of the persona summary used in the previous model. The Knowledge graph has lesser number of characters overall, but due to more usage of punctuation, used for separating individual triplets, it has a higher token count.

Previously, Knowledge graphs have been used as external knowledge bases, which is explained in section 3.4. The main idea introduced by this paper is to eliminate the need for external database systems and utilize the knowledge graph as part of the prompt. This helps us avoid overhead from making additional API calls.

Finally, the following prompt is used to generate the responses for the queries in the test dataset. The model is given both the user's query and the knowledge graph as input.

Your name is Sergeant Blackwell. A virtual soldier in the 1-23rd Infantry. You are given context in the format [head, relation, tail]. Given an utterance, use relevant information from the context to generate a first person response.
Context: < List of triplets>
Do not use any external information. Include the keyphrase (if present) in your answer. Do not change the structure of the keyphrases and strictly adhere to it. If you cannot respond, say "Sorry. That's outside my Area of Operation".

3.4 Retrieval Augmented Generation

Retrieval augmented generation is a prompt engineering technique that has proven to reduce hallucination error rates in question answering scenarios (Shuster et al., 2021). The main idea behind this method is to query a knowledge base to retrieve relevant information before prompting a generative model with both the query and the fetched information. This allows the model to ground its response on the given information which in turn reduces hallucination error rates. The downside of RAG systems is the added overhead due to database querying. Even though this overhead makes RAG systems undesirable for our use-case, it is still important to compare the hallucination error rates of our proposed method with that of a RAG system since it is a well established method.

We created a simple RAG system utilizing BERT embeddings (Devlin et al., 2019) to represent text and compare similarity. Upon receiving a question from the user, the model calculates its embedding and then fetches the most similar questions from the dataset using cosine similarity. The answers to these fetched questions are then used to generate an answer for the user's question.

3.5 Random

In the random baseline approach, we utilize all predefined responses available in the dataset. For each query in the test set, one response is randomly selected from this pool. This selection process is done with uniform probability, meaning each response has an equal chance of being chosen. Importantly, responses are sampled with replacement, allowing the same response to be potentially chosen multiple times across different queries. This method provides a simple benchmark for evaluating the performance of more sophisticated models.

4 Evaluation

In this section, we talk about the evaluation of the models. Each model was used to generate responses for a fixed test dataset. Then they were compared against each other and a random baseline in the following experiments.

4.1 Response Time (Latency)

End-to-end spoken dialogue systems are comprised of multiple modules, typically starting with an automatic speech recognition (ASR) component that transcribes the user's spoken input (Yao et al., 2010). The focus of this paper is on the subsequent response generation module, which is followed by a text-to-speech (TTS) system to deliver the output. For interactions to feel natural and human-like, it is crucial that these modules operate with minimal latency to achieve real-time performance. Hence, response generation latency is a critical factor in evaluating system efficacy. Table 1 presents the average response times for the models evaluated in this study.

Apart from the random baseline, the NPCEditor is the fastest model. This is expected as the other models involve generation of responses using GPT whereas the NPCEditor is a simple and lightweight relevance based classifier. However, it is important to note that the RAG model is the slowest model, due to the added overhead of calculating BERT embeddings and retrieving similar questions from the dataset before the response generation can begin. This further highlights the challenges of using RAG in spoken dialogue systems. Amongst all the generative models, the Knowledge Graph grounded model is the fastest.

Model	Response Time (seconds)
NPCEditor	0.01 ± 0.02
PersonaGPT	1.01 ± 0.55
RAG	1.26 ± 0.64
KGGPT	0.82 ± 0.36
Random	0.00 ± 0.00

Table 1: Average Response Times (in seconds) of different models.

4.2 Response Relevance

Ensuring that generated responses are contextually appropriate and relevant to the user's input is a key factor in maintaining coherent and meaningful interactions. This experiment evaluates the relevance of system responses by measuring how well the generated utterances align with the user's question.

We chose text entailment scores as a metric to evaluate the relevance of the models' generated responses. Ideally, a good response should always be entailed from the question. Since BERT (Devlin et al., 2019) was trained on the next sentence prediction task, a form of text entailment (Sun et al., 2022), we used it to generate the entailment scores. Table 2 shows the entailment rates for the different models.

Model	Entailment Rates
Random	5.67%
NPCEditor	63.56%
PersonaGPT	97.78%
RAG	98.74%
KGGPT	97.47%

Table 2: Entailment Rates for different generative models

Through this study we highlight the drawbacks of the NPCEditor. Even though it produces fast responses, it is limited to a predefined set of responses which cannot generalize to previously unseen questions. The generative models can overcome this challenge and show high entailment

rates.

4.3 Hallucination Rates

Hallucinations, or the generation of factually incorrect or contextually irrelevant information, present a significant challenge in generative dialogue systems. These errors can undermine user trust and compromise the overall utility of the system. This experiment focuses on evaluating the hallucination error rates across different generative models by examining instances where the system produces information that is not supported by the given context or knowledge sources. Since the random baseline and the NPCEditor model return responses from a curated list and do not generate any new text, these models do not hallucinate. Thus, they have been excluded from this experiment.

(Ravi et al., 2024) released Lynx, a model trained on detecting hallucinated responses given an input query and a context. Scores from this model were used to derive the hallucination error rates. Table 3 presents the hallucination error rates observed for the different models.

Model	Hallucination Rates
PersonaGPT	44.83%
RAG	44.08%
KGGPT	40.15%

Table 3: Hallucination Rates for different generative models

The KGGPT model makes the fewest hallucination errors which could be attributed to the model having access to structured information as context.

5 Conclusion and Future Scope

The results of this study highlight the strengths and limitations of various approaches to developing dialogue agents for domain-specific scenarios. Among the models evaluated, the Knowledge Graph Grounded GPT (KGGPT) stands out as a particularly promising solution. It strikes a balance between generating natural, engaging conversations and maintaining factual accuracy and consistency in character portrayal. The model's design, which does not require extensive fine-tuning, makes it highly scalable and adaptable. By simply updating or expanding the knowledge graph with new triplets, the conversational scope can be broadened, and existing information seamlessly updated. Additionally, its independence from external databases simplifies deployment, as all operations are handled within a single API, enhancing usability and reducing overhead.

A key area for future exploration involves examining the model's performance across varying sizes and structures of knowledge graphs. While a smaller, more focused knowledge graph might lower operational costs and improve response latency, it could also lead to increased hallucination rates. Conversely, a larger, more comprehensive knowledge graph could enhance the accuracy of responses but may impact efficiency. Future work should aim to identify an optimal balance between these factors, ensuring that the model remains both effective and efficient across different use cases.

Moreover, while the KGGPT model excels in reducing hallucination errors compared to other generative approaches, there is still room for improvement. Future research could focus on integrating advanced fact-checking mechanisms or leveraging reinforcement learning to further minimize inaccuracies. Additionally, exploring multi-turn dialogue management would help maintain context across interactions, enhancing coherence in prolonged conversations.

In summary, our findings underscore the potential of knowledge graph-grounded models in creating robust, scalable dialogue agents. By addressing the current limitations and exploring new avenues for enhancement, such as dynamic graph expansion, real-world deployment, and integration of other structured data formats, we can refine these models to better serve a wide array of applications—from virtual assistants and educational tools to customer service bots and interactive storytelling.

Limitations

While our study has shown promising results, there are several limitations that warrant consideration in future research. Firstly, our investigation focused exclusively on a single character domain, Sgt. Blackwell, with a limited dataset. This restricts the generalizability of our findings to other characters and domains, necessitating future studies across diverse contexts. Additionally, the current system does not account for multi-turn dialogue management, which could lead to occasional inconsistencies when handling back-and-forth exchanges over extended conversations.

Future iterations should include mechanisms for maintaining context over multiple turns, ensuring coherent and contextually appropriate responses throughout the interaction.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Giovanni Campagna and Rakesh Ramesh. 2017. Deep almond : A deep learning-based virtual assistant [language-to-code synthesis of trigger-action programs using seq 2 seq neural networks].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Jan-Gerrit Harms, Pavel Kucherbaev, Alessandro Bozzon, and Geert-Jan Houben. 2018. Approaches for dialog management in conversational agents. *IEEE Internet Computing*, PP:1–1.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

P. Kenny, A. Hartholt, J. Gratch, W. Swartout, D. Traum, S. Marsella, and D. Piepol. 2007. Building interactive virtual humans for training environments. In *Proc. I/ITSEC*.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 18–27, Sydney, Australia. Association for Computational Linguistics.

Anton Leuski and David Traum. 2010. NPCEditor: A tool for building question-answering characters. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Anton Leuski and David Traum. 2011. Npceditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine*, 32:42–56.

Anton Leuski and David Traum. 2012. A statistical approach for text processing in virtual humans.

OpenAI. 2024. ChatGPT. `https://openai.com/gpt`. [Online; accessed 13-April-2024].

Pēteris Paikens, Artūrs Znotiņš, and Guntis Bārzdiņš. 2020. Human-in-the-loop conversation agent for customer service. In *Natural Language Processing and Information Systems*, Lecture notes in computer science, pages 277–284. Springer International Publishing, Cham.

Debaditya Pal, Anton Leuski, and David Traum. 2023. Comparing statistical models for retrieval based question-answering dialogue: Bert vs relevance models. *The International FLAIRS Conference Proceedings*, 36(1).

Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Selvan Sunitha Ravi, Bartosz Mielczarek, Anand Kannappan, Douwe Kiela, and Rebecca Qian. 2024. Lynx: An open source hallucination evaluation model.

Susan Robinson, David Traum, Midhun Ittycheriah, and Joe Henderer. 2008. What would you ask a conversational agent? observations of human-agent dialogues in a museum setting. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. 2022. NSP-BERT: A prompt-based few-shot learner through an original pre-training task —— next sentence prediction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3233–3250, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

William Swartout, David Traum, Ron Artstein, Dan Noren, Paul Debevec, Kerry Bronnenkant, Josh Williams, Anton Leuski, Shrikanth Narayanan, Diane Piepol, et al. 2010. Ada and grace: Toward realistic and engaging virtual museum guides. In *Intelligent Virtual Agents: 10th International Conference, IVA 2010, Philadelphia, PA, USA, September 20-22, 2010. Proceedings 10*, pages 286–300. Springer.

Fengyi Tang, Lifan Zeng, Fei Wang, and Jiayu Zhou. 2021. Persona authentication through generative dialogue.

David Traum, Andrew Jones, Kia Hays, Heather Maio, Oleg Alexander, Ron Artstein, Paul Debevec, Alesia Gainer, Kallirroi Georgila, Kathleen Haase, et al. 2015. New dimensions in testimony: Digitally preserving a holocaust survivor's interactive storytelling. In *Interactive Storytelling: 8th International Conference on Interactive Digital Storytelling, ICIDS 2015, Copenhagen, Denmark, November 30-December 4, 2015, Proceedings 8*, pages 269–281. Springer.

Nick Webb, David Benyon, Jay Bradley, Preben Hansen, and Oli Mival. 2010. Wizard of oz experiments for a companion dialogue system: Eliciting companionable conversation. In *International Conference on Language Resources and Evaluation*.

Xuchen Yao, Pravin Bhutada, Kallirroi Georgila, Kenji Sagae, Ron Artstein, and David Traum. 2010. Practical evaluation of speech recognizers for virtual human dialogue systems. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

A Prompt for KG Generation

The following prompt was used to generate Knowledge Graph triplets from the interview-like dataset:

You are given an interview response from Sgt Blackwell. You need to extract triplets from his response to create a knowledge graph. The triplets should be in the form of [head, relation, tail]. The triplets must be short and concise. Each piece of information should be contained within a single triple. Similar triplets should be merged into a single triple.

The triplets must start with a <start> tag and end with a <end> tag.

For example, for the conversation:
Question: What is your occupation?
Response: I am a police officer.
The triple would be <start>["Sgt Blackwell", "occupation", "police officer"]<end>.

Each response may have multiple triplets. Extract every entity and a relation. Do not include any information that is not present in the response. Do not create triplets for out-of-domain responses like "I don't know", "Why don't you just search it up?", etc.