Evaluating Large Language Models for Document-grounded Response Generation in Information-Seeking Dialogues

Norbert Braunschweiler and **Rama Doddipatla** and **Simon Keizer**
and **Svetlana Stoyanchev**
Toshiba Europe Limited
Cambridge Research Laboratory, 208 Cambridge Science Park
Cambridge, UK
{norbert.braunschweiler,rama.doddipatla,simon.keizer,
svetlana.stoyanchev}@toshiba.eu

Abstract

In this paper, we investigate the use of large language models (LLMs) like *ChatGPT* for document-grounded response generation in the context of information-seeking dialogues. For evaluation, we use the *MultiDoc2Dial* corpus of task-oriented dialogues in four social service domains previously used in the *DialDoc 2022 Shared Task*. Information-seeking dialogue turns are grounded in multiple documents providing relevant information. We generate dialogue completion responses by prompting a *ChatGPT* model, using two methods: *ChatCompletion* and *LlamaIndex*. *ChatCompletion* uses knowledge from *ChatGPT* model pretraining while *LlamaIndex* also extracts relevant information from documents. Observing that document-grounded response generation via LLMs cannot be adequately assessed by automatic evaluation metrics as they are significantly more verbose, we perform a human evaluation where annotators rate the output of the shared task winning system, the two *ChatGPT* variants outputs, and human responses. While both *ChatGPT* variants are more likely to include information not present in the relevant segments, possibly including a presence of hallucinations, they are rated higher than both the shared task winning system and human responses.

1 Introduction

Accessing domain-specific knowledge in task-oriented dialogue modeling is a crucial aspect to provide information-seeking users with relevant, trustworthy and detailed information. Often this knowledge has to be retrieved from various knowledge sources stored in diverse formats and multiple documents. Once the relevant knowledge has been retrieved, a dialogue system then needs to combine it with the dialogue context and user query to generate an informed, coherent and fluent natural language response.

In this paper, we present a comparison of methods for knowledge-grounded response generation in task-oriented dialogues which include traditional retrieval-augmented generation models as well as state-of-the-art large language models (LLMs), such as *ChatGPT* (Ouyang et al., 2022), one of the *GPT-model* variants released by *OpenAI* (OpenAI, 2022). While there is a wide range of possibilities to utilize LLMs for this task, we focus on two methods of prompting LLMs to investigate their capabilities for this particular scenario.

The first method, uses a chat-interface ("Chat-Completion"[1]) which takes as input 1) a dialogue context, 2) a short description of the system role and its domain, and 3) the last user utterance. It thus effectively provides the LLM with context information about the topics in conversation so far and the general topic the user query is rooted in. The second method uses the *LlamaIndex* (Liu, 2022) tool that combines information extraction from multiple documents with the LLM. In this approach, dialogue context is used to extract relevant information from indexed documents and passed as context to the LLM to generate a grounded response.

Our study assesses how suitable these two methods are for the given task by using the *MultiDoc2Dial* (Feng et al., 2021) corpus for evaluation which includes task-oriented dialogues grounded in multiple documents.

Our paper is structured as follows: Following the description of the *MultiDoc2Dial* corpus in the next section, the *DialDoc Shared Task* is introduced which includes the test-set utilized in this paper. Then, the four response generation methods compared in our study are described in detail, in particular focusing on the two new methods accessing the *GPT-model*. This is followed by a definition of the experimental design and a presentation of

[1] https://platform.openai.com/docs/guides/chat/chat-completions-beta

Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants!, pages 46–55
September 12, 2023 ©2023 Association for Computational Linguistics

Domain	#doc	#dial	2seg	>2seg	single
ssa	109	1191	701	188	302
va	138	1337	648	491	198
dmv	149	1328	781	257	290
sta	92	940	508	274	158
total	488	4796	2638	1210	948

Table 1: Statistics of the *MultiDoc2Dial* corpus.

	Train	Val	Test
#dialogues	3474	661	661
#queries	21453	4201	4094
avgQueryLength	104.6	104.2	96.5
avgResponseLength	22.8	21.6	22.3

Table 2: Train/val/test-splits of *MultiDoc2Dial*.

the results of objective and human evaluations and finished in a conclusion.

2 MultiDoc2Dial corpus

The *MultiDoc2Dial* dataset[2] contains 4.8k dialogues (61078 turns) between an information-seeking user and an agent. Dialogues include an average of 14 turns and are grounded in 488 documents from 4 domains: Department of Motor Vehicles (*dmv*), Social Security Administration (*ssa*), Federal Student Aid (*sta*) and Veterans Affairs (*va*). Documents include HTML mark-ups (e.g. title, list) and document section information (title, text body, spans/sections). The dialogue data includes annotations at the turn-level for dialogue act, speaker role, human-generated utterance and the associated grounding span with document information. We chose this dataset for our evaluation as it provides a) task-oriented dialogues in which agent responses are grounded in multiple documents from four domains, b) the content of these documents, and b) manually entered agent responses plus gold standard passages from associated documents which provide relevant grounding information extracted from the document that may be used to assess the correctness of model generated responses.

Table 1 shows an overview of the *MultiDoc2Dial* (Feng et al., 2021) corpus providing the number of documents and dialogues in each domain as well as the number of dialogues which include one or more segments where a segment includes turns grounded in the same document.

2.1 DialDoc Shared Task

The *MultiDoc2Dial* corpus was also used as training, validation and test set for the *DialDoc 2022 Shared Task*[3] on “Open-Book Document-grounded Dialogue Modeling” (Feng et al., 2022). This shared task mainly focused on building open-book goal-oriented information seeking conversation systems, where an agent could provide an answer or ask follow-up questions for clarification or verification (Feng et al., 2022). The main goal was to generate grounded agent responses in natural language based on the dialogue context and domain knowledge in the documents. Specifically, taking as input 1) latest user turn, 2) dialogue history and 3) all domain documents and then generate as output the agent response in natural language. A subtask was grounding span prediction which aimed at locating related spans from multiple documents. A summary of the shared task is provided in Feng et al., 2022.

Table 2 shows the splits of the *MultiDoc2Dial* corpus into sets for training, validation and testing used in the *DialDoc 2022 Shared Task*.

As part of the *DialDoc Shared Task*, organizers provided a baseline model and published a leaderboard[4] of participants models performances in various stages during the shared task. For our comparison, we are using both this baseline model as well as the responses generated by the shared task winning team CPII-NLP (Li et al., 2022), in the *SEEN* evaluation setting where the test dialogues shared the same domains as the training data (Feng et al., 2022).

The shared task evaluation is based on a subset of 661 turns selected from the 4094 test-set turns. These 661 turns also formed the test-set of the models compared in this study, including the GPT-based models introduced in the next section.

3 Response generation methods

This section introduces the response generation methods compared in this paper. The selection of methods was based on approaches which could a) serve as baselines because they had published performance results on the *MultiDoc2Dial* corpus, b) were available at the time of writing, c) utilized the capabilities of the *GPT-models*, and d) could be evaluated in both objective and human evaluation.

[2] http://doc2dial.github.io/multidoc2dial/

[3] https://doc2dial.github.io/workshop2022/

[4] https://eval.ai/web/challenges/challenge-page/1437/leaderboard/3575

Since the character of this study is to get an initial indication of the abilities of *GPT-models* for the given task of document-grounded response generation for information-seeking dialogue modeling, we restricted the number of approaches to a total of four, including two baseline models using established architectures and two methods employing *GPT-models*.

3.1 Baselines

As baselines, we selected two of the models which were part of the *DialDoc 2022 Shared Task* (Feng et al., 2022) described in the previous section 2, i.e., the baseline model for the *DialDoc Shared Task* (henceforth called *RAGBase*) and the shared task winning model from team *CPII-NLP* (Li et al., 2022) (henceforth called *CPII-NLP*).

The *RAGBase* model applies a Retrieval-Augmented Generation (RAG) (Lewis et al., 2021) architecture described in Feng et al., 2021 which combines a retriever model employing a *Dense Passage Retriever (DPR)* (Karpukhin et al., 2020) with a generator adopting the *BART-large* (Lewis et al., 2020) model which was pre-trained on the CNN dataset (Feng et al., 2021).

Model *CPII-NLP*, which significantly outperformed the baseline model in the *DialDoc Shared Task*, also includes a *Dense Passage Retriever (DPR)* and a *BART-large* model for generation, but extends this architecture with a re-ranker (following the retrieval step) that includes an ensemble of 3 cross-encoder models using *BERT* (Devlin et al., 2019), *RoBERTa* (Liu et al., 2020), *ELECTRA* (Clark et al., 2020), while the *BART-large* model was jointly trained with a grounding span predictor. The 3 components are individually optimized, while passage dropout and regularization techniques are adopted to improve the response generation performance.

The two baseline models provided benchmarks for assessing the two new models introduced next, which use two variants of utilizing one *GPT-model* for knowledge-grounded response generation.

3.2 GPT-based models

To enable an equal comparison we selected one LLM from the *GPT-models* repertoire available from OpenAI[5]: The *gpt-3.5-turbo* model. It includes optimizations for chat and usage costs are significantly lower (1/10th) than for the *text-davinci-003* model[6].

Method	Features
RAGBase	Retrieval-Augmented Generation (RAG), DialDoc baseline model
CPII-NLP	Pipeline system of retriever, re-ranker, generator, DialDoc winner
GPTChat	GPT-ChatCompletion-API & system intro prompt, no grounding
GPTLama	GPT & Knowledge-grounded prompt generation via LlamaIndex

Table 3: Methods for knowledge-grounded response generation which are compared in this paper.

The *gpt-3.5-turbo* model has a token input limit of 4096 tokens, uses training data up to September 2021 and has usage costs of $0.002/1k tokens.

We select two variants of accessing *gpt-3.5-turbo* motivated by their availability at the time of writing and their capabilities to a) represent dialogue context (*GPTChat*) and b) retrieving relevant knowledge from the associated documents (*GPTLama*).

The goal of this comparison is to evaluate these models in the selected scenario of knowledge-grounded response generation for task-oriented dialogue introduced in section 2. Generating a response for a user query which takes a) dialogue context into account, and b) knowledge retrieved from multiple documents.

Table 3 shows the four systems compared in this study. Next, the two models for knowledge-grounded response generation utilizing the *gpt-3.5-turbo* LLM are introduced in detail.

3.2.1 ChatCompletion method: *GPTChat*

OpenAI provides a *ChatCompletion*-module[7], which allows input to *GPT-models* in the form of a structured dialogue including user-queries and agent-responses. The model can then use the dialogue context to generate responses in the agent-role for a given user query. While this method does not provide a knowledge retrieval step, it still can be guided by providing additional "system"-messages which can include instructions such as *You are a helpful assistant* or background knowledge such as the domain the user query refers to,

[5] https://openai.com/product#made-for-developers

[6] https://platform.openai.com/docs/models/gpt-3-5

[7] https://platform.openai.com/docs/guides/chat/chat-completions-beta

e.g. *Hello, this is the service agent from the U.S. Department of Motor Vehicles – how can I help you?*.

The *GPTChat* system, therefore, provides a reference for how well the *GPT-model* performs on the given task without having additional knowledge extracted from the associated documents, but instead relies on the capabilities of the *GPT-model* to understand the input and retrieve information from the data it was trained on. As the topics covered in the four domains of the *MultiDoc2Dial* corpus were in the public domain (websites), it is likely that they were part of the *GPT-model's* training data. But even if it was part of the training data, there is still the challenge of understanding the user input in the context of the dialogue and then formulating an adequate answer for it.

The dialogues from the test-set of the *MultiDoc2Dial* corpus were used as input to the *ChatCompletion-API*[8] by providing both user questions and agent-responses in the chat format required by the API and shown in Table 8 in appendix A.1.

To provide the *GPT-model* some context information about the domain of the conversation it was given an initial system-prompt with a domain-specific content in the "system" field; these prompts are listed in section A.2 in the appendix, and were intended to specify the role of the *assistant* and its behavior.

Each dialogue in the test-set was first split into sub-dialogues at the turn level, always ending with a user turn in order to get an "assistant"-response generated by the *GPT-model*. As such, each split formed an individual dialogue for the *GPT-model* with its own context including all previous turns up to the user turn to be answered until the full dialogue was represented. Therefore, the amount of context was growing for each additional turn in the dialogue. However, none of the dialogues in the test-set exceeded the token limit of 4096 set in the *gpt-3.5-turbo* model. The average number of tokens in the full dialogues was 413 with a maximum of 669 tokens.

3.2.2 LlamaIndex method: *GPTLama*

In contrast to the *GPTChat* model, which does not access the associated documents of the *MultiDoc2Dial* corpus for grounding its responses, another method was selected which provided that functionality: *LLamaIndex* (Liu, 2022). This tool, which we used in the system henceforth called *GPTLama*, enables the connection of LLM's with external or private data. It provides tools to load external/private documents and parse them into data structures which can be queried efficiently to retrieve relevant information for a given user input both of which can then be combined into a prompt send to a chosen LLM.

By design, the quality of the response will therefore largely depend on the data retrieved by the query of the indexed documents and to a lesser extend on the language reasoning capabilities of the LLM.

By using *LlamaIndex* tools[9] we aimed at improved accuracy of response generation by grounding it in the associated documents. Since it first retrieves knowledge from documents and then sends it together with the user input to the LLM for response generation its closer to the retrieval-augmented response generation method in the two baseline models.

We used *semantic search*[10] for queries over our documents which were first converted into a *LlamaIndex* vector store by saving each of the 488 documents into a single file and running the *GPTVectorStoreIndex.from_documents(documents)*[11] command across these individual files.

One of the challenges in prompt creation for *LlamaIndex* is to ensure that the user input provides sufficient context information for a) querying the vector index and b) using it for the response generation from the *GPT-model*. In our scenario of document-grounded response generation for task-oriented dialogue modeling, an isolated user turn, i.e., the last utterance of dialogue context, might not include sufficient content to retrieve relevant information from the associated documents. In the *MultiDoc2Dial* corpus, a typical initial user input asks a specific question related to one of the four domains and includes already information to link it to one of these domains. However, some inputs can be as simple as *Hi there* or *I need to know how to apply, please* which do not indicate any particular domain and are therefore not useful for retrieving relevant information from the documents. There-

[8] https://platform.openai.com/docs/guides/chat/introduction

[9] We used LlamaIndex version 0.6.0

[10] https://gpt-index.readthedocs.io/en/latest/use_cases/queries.html#semantic-search

[11] https://gpt-index.readthedocs.io/en/latest/getting_started/starter_example.html#build-and-query-index

fore, to provide sufficient information about the domain of the user input it was provided as part of the query string, e.g. *Question for the U.S. department for Veterans Affairs (VA) service agent:* (see section A.3 for a full list of prompts) and then followed by the user query. This way the retrieval step had at least a chance to locate domain related documents. In addition, by specifying the role of the "responder" as "service agent" it provided additional instructions for the LLM how to respond.

Another aspect is the representation of dialogue context within the *LlamaIndex* framework. Since it does not provide the same dialogue representation format as in the *ChatCompletion-API* introduced in section 3.2.1 it is not as straightforward to be included. To provide dialogue context we used the *QuestionAnswer*-prompt[12] method in *LlamaIndex* which requires both a *query_str* field and a *context_str* field, with the user query included in the *query_str* field and the *context_str* including the information retrieved from associated documents by using the *query_str* as search query.

We used the *query_str* field to enter the last user turn of the dialogue under consideration and loaded the *context_str* with previous dialogue context as well as "system"-instructions specifying the response behavior of the LLM. Examples of domain-specific entries used for *query_str* and enriched *context_str* are in appendix sections A.3 and A.4 respectively.

Because of the additional information from the retrieval step, the *LlamaIndex* method uses more tokens per dialogue call than the *ChatCompletion* method, while still remaining below the token limit of 4096 for the test-set dialogues. The average number of tokens send to the *GPT-model* via the *GPTLama* method was 1114 with a maximum of 1590 tokens.

4 Experimental design

We compare four response generation methods in a task-oriented dialogue scenario of which three are using document-grounded generation, in which user input requires the system to find and retrieve relevant knowledge from multiple documents and one method relying on the abilities of the selected LLM to retrieve relevant knowledge from its own training data.

Methods are compared using a) objective metrics, i.e., *RougeL* (Lin, 2004), *METEOR* (Banerjee and Lavie, 2005), token-level *F1-score* (Rajpurkar et al., 2016), and *SacreBLEU* (Post, 2018), and b) by human evaluation.

5 Results

5.1 Objective evaluation

Table 4 shows results of objective metrics for the four knowledge retrieval methods on the 661-turn test-set from the *MultiDoc2Dial* corpus. We report the locally replicated performance figures in the *SEEN* category for models *RAGBase* based on the re-implementation of the *DialDoc*-baseline model and for model *CPII-NLP* based on responses provided by the authors (Li et al., 2022). Both of these performance figures are deviating only marginally from published results.

It can be seen, that the objective metrics for the *GPT*-based models are significantly lower than both the baseline model as well as the winning model. One of the reasons for this lower performance is that *GPT*-generated responses are on average much longer than the relatively short responses in the reference corpus as seen in the average number of words per response column (Avg#words). While these objective metrics provide a useful indication for the quality of the response, they are based on the word overlap between reference and prediction. As only one 'gold' reference is available, the automatic measures may fail to recognize an appropriately generated response which addresses a different aspect than the 'gold' reference.

Therefore, we decided to conduct a human evaluation of agent's responses, which is described next.

5.2 Human evaluation

For the human evaluation, 25 dialogue snippets with the length between three and nine turns (to reduce cognitive load on the annotators) were randomly selected from the test subset of the *MultiDoc2Dial* corpus. Each dialogue snippet starts from the beginning of the dialogue and ends on a user's turn. We refer to these snippets as dialogue context.

As we address a 'grounded' dialogue task, each dialogue context is associated with *grounding segment(s)* from the relevant documents that guides the agent's response. We use four experimental conditions to generate the agent response for each dia-

[12] https://gpt-index.readthedocs.io/en/v0.5.27/how_to/customization/custom_prompts.html#example

System	Avg#words	F1	SacreBLEU	METEOR	RougeL	Total
RAGBase	19.2 (10.2)	35.59	22.49	34.62	33.84	126.55
CPII-NLP	20.6 (11.4)	52.17	37.46	51.71	50.15	191.51
GPTChat	69.6 (20.4)	17.59	2.30	24.18	13.88	57.95
GPTLama	59.2 (46.7)	17.33	4.89	22.71	15.30	60.25

Table 4: Comparison of response generation methods on *MultiDoc2Dial*-corpus 661-turn test-set.

System	Appropriateness	InfoNotInGround
	Q1	Q3
Reference	4.07 (1.33)	10.6%
CPII-NLP	3.90 (1.39)	16.4%
GPTChat	4.17 (1.22)	88.0%
GPTLama	4.19 (1.26)	84.0%

Table 5: Results of human evaluation of response generation methods for appropriateness (5-point Likert scale, 5=completely appropriate) and percent responses containing information that is not in the reference (InfoNotInGround).

logue context: *GPTChat*, *GPTLama*, *CPII-NLP*[13], and the 'gold' human agent's response (*Reference*). We include the *Reference* condition in the human evaluation for the comparison with the automatically generated ratings and to determine how often the agents actually used the *grounding segment* to produce their response. We omit the baseline from the human experiment as it has already been shown that *CPII-NLP* significantly outperforms the baseline in human rating (Feng et al., 2022).

The annotator is presented with a dialogue context, the corresponding grounding segments, the agent's response from of the experimental conditions, and three questions (see Table 6). Q1 asks to rank appropriateness of the response in context of the dialogue on a 1-5 scale. Q2 asks whether the generated response included *relevant* information from the grounding segments[14]. Annotators could select between four options, including *'None of the above'* which may happen when the reference did not contain relevant information. Q3 aimed to detect whether the agent response contained information other than the reference. After submitting the response, the annotator could choose to continue to another question.

Table 5 shows the results of Q1 (Appropriateness) and Q3 (InfoNotInGrounding). The two *GPT-model* based methods outperform both *Reference* and *CPII-NLP* model. *GPT-model* responses contain more information, which may have had a positive effect on the perception of the annotators. Both *GPT*-based methods show very similar appropriateness scores indicating that the *GPTChat* method providing dialogue context with an introductory system prompt was sufficient to provide a response judged the most appropriate by the annotators.

As expected for Q3, *Reference* has the lowest proportion of responses with information outside of grounding segments (10.6%). Both *GPT-models* used significantly more information than what was provided in the grounding reference, with 88.0% for the *GPTChat* method and a slightly lower percentage for the *GPTLama* system (84.0%), which indicates that the information retrieved from documents seemed to have a small impact on the generated content.

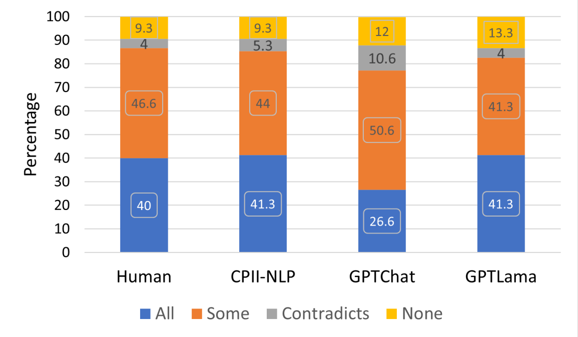

Figure 1: Percentage of agent responses containing all/some/contradictory/no information from reference grounding passage (Q2).

Figure 1 shows the percentages of agent responses containing all/some/contradictory/no information from the reference grounding passages. Results show that human reference as well as system *CPII-NLP* have the highest proportion of responses using *all* or *some* information from the

[13] The authors kindly shared the generated responses with us

[14] We noticed that not all information in grounding segments is relevant and emphasized that the annotators should look only for information in the grounding segments that is relevant to dialogue context.

	Question	Answering options
Q1	Rate the appropriateness of the last agent's response *(Appropriateness)*	1: Completely Inappropriate; 2: Somewhat Inappropriate; 3: Uncertain; 4: Somewhat Appropriate; 5: Very Appropriate
Q2	Does the last agent's utterance contain all/some/contradictory RELEVANT information from the reference? *(InfoInGrounding)*	'All Relefant info', 'Some relevant info', 'Contradicts', 'None of the above'
Q3	Does the last agent utterance contain information that is NOT in the grounding segment? *(InfoNotInGrounding)*	yes, no

Table 6: Human evaluation questionnaire.

grounding segment, followed by system *GPTLama* and the lowest percentage by *GPTChat*. *GPTChat* has the lowest proportion (26.6%) with *all* and the highest with *some* information (50.6%). 'None of the above' response for Q2 indicates that a response does not contain relevant information of the grounding segment or that the grounding segment does not contain any relevant information. *GPTChat* and *GPTLama* have 12.0% and 13.3% of such responses. Surprisingly, *Reference* also contains 9.3% of such responses indicating that grounding segments may not be always relevant. *GPTChat* has with 10.6% the highest proportion of responses with contradictory information possible indicating some hallucinations, while the remaining three systems range between 4% and 5.3% responses ranked as contradicting.

We compute Krippendorff's α (Krippendorff, 2006) to measure the inter-annotator agreement[15]. The agreement on *Appropriateness* is only 0.35. In this subjective task, the annotators may have had different personal criteria for the agent response appropriateness judgment. Agreement on *InfoInGrounding* is 0.46. Judging information in grounding segments was open to the annotators' interpretation. Agreement on *InfoNotInGround* is the highest with $\alpha = 0.69$ indicating that the annotators were generally in agreement whether information outside of the grounding segment was presented.

Examples of model generated responses for a given user question are shown in Table 7. The example shows that none of the models predicts the correct response of *the civil fine is $ 200* but models differ in the amount of the fine and while the two baseline models are very close in their wording to the reference only differing in the amount of the fine, the GPT-generated responses are much longer and either incorrectly retrieve or hallucinate the amount of the fine, but formulate their responses very sovereign and convinced.

	Dialogue context
Agent:	Hi, I'm the service agent from the U.S. Department of Motor Vehicles and can help you with any questions related to vehicle registration, drivers licenses or driving fines - how can I help you?
User:	"Hi, I need to pay a civil penalty. How much is the fine if the order was issued between July 1, 1990 and October 31, 1996?" Relevant information from documents: *"if the order was issued between January 1 , 1981 and June 30 , 1990 , the civil penalty is $100 if the order was issued between July 1 , 1990 and October 31 , 1996 , the civil penalty is $200 if the order was issued between November 1 , 1996 and October 31 , 2006 , the civil penalty is $300 "*
Source	**Response**
Reference	"the civil fine is $ 200"
RAGbase	"the fine is $ 250"
CPII-NLP	"The civil fine is $ 500"
GPTChat	"If the fine was issued between July 1, 1990, and October 31, 1996, the amount of the civil penalty for an uninsured operation is $300, and the amount for an uninsured accident is $500."
GPTLama	The civil penalty for orders issued between July 1, 1990 and October 31, 1996 is $500.

Table 7: Examples of model generated responses compared to hand-written reference response from corpus.

We used the sandbox of the Amazon Mturk interface to set up the evaluation interface. With 25 examples and four variants of the agent responses, the experiment included 100 unique human intelligence tasks (HITs). Each HIT was assigned to three annotators and a total of nine raters participated in the experiment. Given the complexity of

[15] computed between the three annotation instances

the task, all recruited annotators were colleagues from our lab (including the authors) and were not paid for this task.

6 Conclusion

This paper presented a study of document-grounded response generation methods for information-seeking dialogue modeling particularly including two methods utilizing one of the *ChatGPT* large language models for this task. The comparison was conducted by both objective metrics as well as by human evaluation. Objective evaluation results showed that typical word-overlap based metrics are not suitable to fully assess the performance of these methods and human evaluation indicated that *ChatGPT*-based models have strong potential in this domain, even exceeding appropriateness-scores for the human-authored reference responses. Just providing dialogue context and a system-prompt specifying the domain and the role of the response generator was sufficient to outperform human generated responses on the subjective appropriateness metric. While the system which additionally utilized document-retrieved information showed the highest appropriateness score, it was only marginally better than the system without additional retrieval step. However, the potential to access specific external information (or private information) not seen by the LLM during training is essential in specific domains. Especially when information from external documents can be reliably retrieved and methodically inserted into the LLM prompts can help reduce hallucinations. However, human evaluation also visualized the challenges in assessing the accuracy and veracity of *ChatGPT*-generated responses which can simultaneously appear very well formulated but factually wrong. In future work we plan to apply fact verification methods for assessing reliability of generated responses.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*.

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.

Song Feng, Siva Patel, and Hui Wan. 2022. DialDoc 2022 shared task: Open-book document-grounded dialogue modeling. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 155–160, Dublin, Ireland. Association for Computational Linguistics.

Song Feng, Siva Sankalp Patel, Hui Wan, and Sachindra Joshi. 2021. Multidoc2dial: Modeling dialogues grounded in multiple documents. *CoRR*, abs/2109.12595.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Klaus Krippendorff. 2006. Reliability in content analysis: Some common misconceptions and recommendations. *Human Communication Research*, 30(3):411–433.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.

Kun Li, Tianhua Zhang, Liping Tang, Junan Li, Hongyuan Lu, Xixin Wu, and Helen Meng. 2022. Grounded dialogue generation with cross-encoding re-ranker, grounding span prediction, and passage dropout. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 123–129, Dublin, Ireland. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jerry Liu. 2022. LlamaIndex.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

OpenAI. 2022. ChatGPT: Optimizing language models for dialogue.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

A Appendix

A.1 *GPTChat*: Example of input format

{"role": "system", "content":	"Hello, welcome to the Department of Motor Vehicle information service agent - how can I help you?"},
{"role": "user", "content":	"Hello, I forgot to update my address, can you help me with that?"},
{"role": "assistant", "content":	"hi, you have to report any change of address to DMV within 10 days after moving. You should ... vehicles."},
{"role": "user", "content":	"Can I do my DMV transactions online?"}

Table 8: Example of ChatCompletion input format of dialogues.

A.2 *GPTChat*: Initial system prompts for each domain

DMV "Hello, welcome to the U.S. Department of Motor Vehicle information service agent - how can I help you?"

SSA "Hello, I'm the service agent from the U.S. Government Social Security Administration department and can help you with any questions about retirement, disability benefits, how to get or replace your Social Security card, and more."

STA "Hi, I'm the service agent for the U.S. department for Federal Student Aid, which offers grants, loans, work-study, and more to help you pay for college or career school. I can answer your questions related to the Free Application for Federal Student Aid (FAFSA), types of student aid and the many ways to get help paying for college or career school."

VA "Hello, I'm the service agent for the U.S. Department of Veterans Affairs (VA) where service members, veterans, and their beneficiaries can apply for benefits services. I'm also linked with the Federal Benefits Unit (FBU) and can answer your questions about our benefits and services."

A.3 *GPTLama*: Initial domain-specific prompts

DMV "Question for the U.S. department for Veterans Affairs (VA) service agent: "

SSA "Question for the U.S. Government Social Security Administration (SSA) service agent: ”

STA "Question for the U.S. department for Federal Student Aid service agent: "

VA "Question for the U.S. department for Veterans Affairs (VA) service agent: "

A.4 *GPTLama*: Domain-specific prompts before dialogue context

DMV "<system> Hello, I'm the service agent from the U.S. Department of Motor Vehicles (DMV) Here is the conversation I had with the user before I received the question to be answered by you:</system>."

SSA <system> "Hello, I'm the service agent from the U.S. Government Social Security Administration (SSA) department and can help you with any questions about retirement, disability benefits, how to get or replace your Social Security card, and more. Here is ... </system>.”

STA "<system> Hi, I'm the service agent for the U.S. department for Federal Student Aid, which offers grants, loans, work-study, and more to help you pay for college or career school. I can answer your questions related to the Free Application for Federal Student Aid (FAFSA), types of student aid and the many ways to get help paying for college or career school. Here is ...</system>."

VA "Hello, I'm the service agent for the U.S. Department of Veterans Affairs (VA) where service members, veterans, and their beneficiaries can apply for benefits services. I'm also linked with the Federal Benefits Unit (FBU) and can answer your questions about our benefits and services. Here is ...</system>."

A.5 *GPTLama*: Example of prompt

query_str = "Question for the U.S. Department of Motor Vehicles (DMV) service agent: <user> My driver license has been suspended and I need help fixing this."

QA_PROMPT_TMPL = ("We have provided context information below: \n\" "—\n"\"<system> Hello, I'm the service agent from the U.S. Department of Motor Vehicles (DMV)\Here is the conversation I had with the user before I received the question to be answered by you:</system>. \n\" "And here is some context information I've extracted from my database:" "{context_str}" "\n—\n" "Given this information, please answer the question as service agent from the US Department of Motor Vehicles: {query_str}\n")