Language Models that Seek for Knowledge: Modular Search & Generation for Dialogue and Prompt Completion

Kurt Shuster
Meta AI

Mojtaba Komeili
Meta AI

Leonard Adolphs *
ETH Zürich

Stephen Roller
Meta AI

Arthur Szlam
Meta AI

Jason Weston
Meta AI

Abstract

Language models (LMs) have recently been shown to generate more factual responses by employing modularity (Zhou et al., 2022) in combination with retrieval (Adolphs et al.). We extend the recent approach of Adolphs et al. to include internet search as a module. Our SeeKeR (Search-engine→Knowledge→Response) method thus applies a single LM to three modular tasks in succession: search, generating knowledge, and generating a final response. We show that, when using SeeKeR as a dialogue model, it outperforms the state-of-the-art model BlenderBot 2 (Chen et al., 2021) on open-domain knowledge-grounded conversations for the same number of parameters, in terms of consistency, knowledge and engagingness. SeeKeR applied to topical prompt completions as a standard language model outperforms GPT2 (Radford et al., 2019) and GPT3 (Brown et al., 2020) in terms of factuality and topicality, despite GPT3 being a vastly larger model. Our code and models are made publicly available[1].

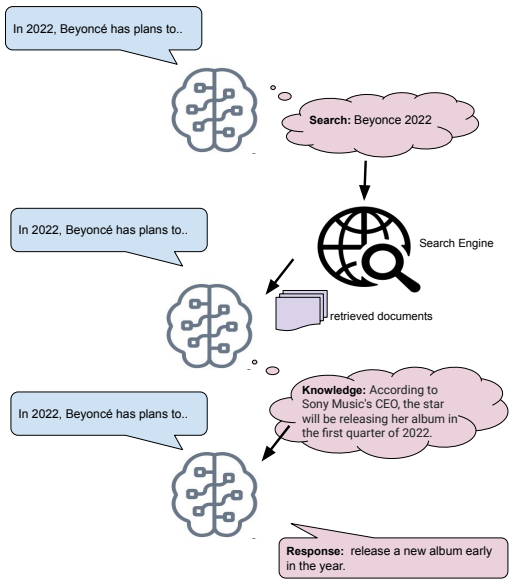

Figure 1: The modular Search-engine → Knowledge → Response (SeeKeR) Language Model. A single transformer architecture is called successively to invoke three different modules: search, generate knowledge, and generate final response. The output of each module is input to the next, in addition to the original context.

1 Introduction

Standard large language models are known to generate fluent but factually incorrect statements, a problem that is not solved by just increasing their size (Shuster et al., 2021). Additionally, as their knowledge is frozen in time from the point when they were trained, they can never learn new facts – the newest information they have will be from the date that the training set was constructed. Several recent advances have tried to tackle aspects of these problems. Neural retrieval models have augmented seq2seq models with access to a large fixed corpus of knowledge (Lee et al., 2019a; Lewis et al., 2020b). However, aggregating information from multiple retrieved documents is a difficult problem (Izacard and Grave, 2021b) which may result in incorporating parts of multiple documents into one factually incorrect response. A modular approach which first finds the relevant parts of the documents and then generates the final response has been shown to help alleviate this problem (Adolphs et al.). However, none of those methods can incorporate new information, which has been studied in separate work that augments generations with internet search (Komeili et al., 2022).

In this paper, we explore a modular architecture that tries to mix the best elements of these different existing solutions. A single transformer architecture is used iteratively to perform three modular tasks: search, generate knowledge, and generate a final response, where the output of each module is fed as additional input to the next, as in Figure 1. The first step, given the input context, generates a

*Work done during a Meta AI Research internship.
[1]Code is available at: https://parl.ai/projects/seeker

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 373–393
December 7-11, 2022

relevant search query for an internet search engine, while the second step is fed the returned documents and generates their most relevant portion. The last step uses that knowledge to produce its final response. By decomposing this difficult problem into three manageable steps, pertinent up-to-date information can be incorporated into the final language model generation.

We apply our modular Search-engine→ Knowledge → Response (SeeKeR) language model to the tasks of dialogue and prompt completion, after pre-training and fine-tuning on a variety of knowledge-intensive datasets. In open-domain dialogue, we show this approach outperforms the state-of-the-art BlenderBot 2 model of Chen et al. (2021) according to human ratings of consistency, knowledge and per-turn engagingness.

We test the ability of SeeKeR to perform general – but up-to-date – language modeling. To do this we construct topical prompts on subjects that were in the news in January 2022, which is data that the model itself has not been trained on. With SeeKeR's ability to incorporate information via web search, it outperforms GPT2 (Radford et al., 2019) and GPT3 (Brown et al., 2020) in terms of factuality and topicality according to human raters.

2 Related Work

Our work builds on the knowledge to response (K2R) technique (Adolphs et al.) which decomposes a dialogue model into two stages: generating a knowledge sequence, followed by generating a response sequence, conditioned on the knowledge. This was applied successfully to Wizard of Wikipedia (Dinan et al., 2019), QA (Lee et al., 2019a) and LIGHT tasks (Urbanek et al., 2019). We expand on this approach by adding the additional module of internet search and then applying that to full open-domain dialogue and general language modeling.

In the dialogue space, the most natural comparison to our approach is BlenderBot 2 (BB2) (Chen et al., 2021). BB2 grounds on retrieval from the internet for open-domain dialogue tasks (Komeili et al., 2022), but does not use a modular approach to generate knowledge, instead applying the fusion-in-decoder (FiD) method (Izacard and Grave, 2021a) to output a response directly given the retrieved documents. They, as well as others (Lee et al., 2022), report that their method can have the problems of either mixing up facts together incorrectly or generating a generic response that ignores the knowledge, which our method attempts to address. Another recent approach that uses information retrieval is LaMDA (Thoppilan et al., 2022), where the retrieval engine returns pertinent information (rather than a set of documents) and is considered a separate black box. LaMDA is not openly available and cannot be compared to. WebGPT (Nakano et al., 2021) also applies internet search to QA tasks, as does the work of Lazaridou et al. (2022); neither applies to dialogue or general LM tasks, and neither work is openly available.

In the language modeling space, there is a large body of work on nearest neighbor and cache-based language modeling (Khandelwal et al., 2020; Grave et al., 2017; Merity et al., 2017; Khandelwal et al., 2021; Yogatama et al., 2021) for accessing a large set of documents. Recently, RETRO (Borgeaud et al., 2021) used retrieval over a database of trillions of tokens. Those works do not use internet search, but rather perform their own retrieval method via a transformer model together with nearest neighbor lookup. As the database is fixed, that means it would not be up to date with the latest knowledge and current events. Some recent methods have also attempted to adapt knowledge through editing and tuning of language model variants (De Cao et al., 2021; Mitchell et al., 2022).

3 SeeKeR Model

The SeeKeR model we introduce in this paper has the architecture of a standard transformer (Vaswani et al., 2017), except that this same encoder-decoder (for dialogue) or decoder-only (for language modeling) model is used in a modular way multiple times. For each module, special tokens are used in the encoder (or decoder) to indicate which module is being invoked. The output of each module is input into the next, along with the original context.

SeeKeR consists of three modules, which are invoked sequentially:

Search Module Given the encoded input context, a search query is generated. This is fed into a search engine, which returns results in the form of a set of documents. Following Komeili et al. (2022), in our experiments (unless stated otherwise) we employ the Bing Web Search API[2] to retrieve documents, and then filter that set of documents by intersecting

[2] www.microsoft.com/en-us/bing/apis/bing-web-search-api

with Common Crawl (Wenzek et al., 2020), and keep the top 5.

Knowledge Module Given the encoded input context, and a set of retrieved documents, a knowledge response is generated. This consists of one or more relevant phrases or sentences from the retrieved documents. For encoder-decoder models, the documents and context are encoded using the fusion-in-decoder (FiD) method (Izacard and Grave, 2021a); for decoder-only models, we pack and prepend the documents to the input context. This task is essentially a "copy" task in that no new tokens have to be generated; the difficulty of the task is selecting the relevant knowledge to copy.

Response Module Given the encoded input context concatenated with the knowledge response, the final response is generated. The module must consider relevant context and knowledge while generating a new fluent continuation to the input. The extraction of relevant knowledge by the previous modules makes this task easier; in contrast, a conventional seq2seq model has to solve all these tasks (knowledge acquisition, synthesis, and final response generation) at once.

3.1 Architecture and Pre-Training

For our standard language modeling experiments, we consider the GPT2 transformer (Radford et al., 2019) as a base model, and fine-tune it to become a SeeKeR model (see subsection 3.3); we do not perform any pre-training of our own in this case. We can thus directly compare to GPT2, with the same model size and architecture. We consider medium, large and XL (345M, 762M and 1.5B parameters) models in our experiments.

For our dialogue experiments, we employ a 2.7B parameter transformer encoder-decoder model. To pre-train our model we consider combining two different pre-training datasets for language-modeling and for dialogue, using the training method of Lewis et al. (2020a):

pushshift.io Reddit We use a variant of Reddit discussions, which has also been used in several existing studies, particularly for training BlenderBot 1 and 2 (Roller et al., 2021). The setup requires training to generate a comment conditioned on the full thread leading up to the comment. Following Humeau et al. (2019), this is a previously existing Reddit dataset extracted and obtained by a third party and made available on pushshift.io (Baumgartner et al., 2020), spanning 1.5B training examples from Reddit obtained from PushShift through July 2019. A number of heuristic rules have been used to filter and clean the dataset; see Roller et al. (2021) for details.

RoBERTa+CC100en We use the same data used to train the BASE language model (Lewis et al., 2021), which consists of approximately 100B tokens, combining corpora used in RoBERTa (Liu et al., 2019) with the English subset of the CC100 corpus (Conneau et al., 2020).

We compare pre-training only on dialogue modeling (pushshift.io Reddit, as in Roller et al. (2021)) to pre-training on both language modeling and dialogue modeling tasks; we refer to the latter as R2C2 (pushshift.io Reddit, RoBERTa + CC100en). Full details, including model and pre-training hyperparameters, are given in Appendix B.

3.2 SeeKeR Tasks for Dialogue

We use a number of dialogue-based fine-tuning tasks to enable our model to perform well for each of the three modules, summarized in Table 1.

Search Module Tasks We use data from the Wizard of Internet (WizInt) task (Komeili et al., 2022) which consists of 8,614 training dialogues containing 42,306 human-authored relevant search queries given the dialogue contexts. We can use the search query data as targets to directly train the search module in a supervised fashion. We append special tokens to the input context to indicate that the transformer is performing the search task, via predicting a relevant search query.

Knowledge Module Tasks We multi-task several knowledge-intensive NLP tasks, where the target for the model is the "knowledge" that will be used to generate the final response. We first employ knowledge grounded dialogue datasets that contain annotations of the gold knowledge used: Wizard of Internet (Komeili et al., 2022) and Wizard of Wikipedia (WoW) (Dinan et al., 2019). We then use several QA tasks: SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017), Natural Questions (NQ) (Kwiatkowski et al., 2019), and MS MARCO (Nguyen et al., 2016). We use the "Natural Language Generation" competition track (NLGen v2.1) of MS MARCO, in which the annotator must "provide your answer in a way in which it could be read from a smart speaker and make

Dataset	Number Training Examples		
	Search	Knowledge	Response
Knowledge-Grounded Dialogue			
Wizard of the Internet (Komeili et al., 2022)	35137	22487	22487
Wizard of Wikipedia (Dinan et al., 2019)	-	77310	77310
Open-Domain Dialogue			
PersonaChat (Zhang et al., 2018)	-	55701	55701
Empathetic Dialogues (Rashkin et al., 2019)	-	4393	4393
Blended Skill Talk (Smith et al., 2020)	-	9826	9826
Multi-Session Chat (Xu et al., 2022)	-	74676	74676
Multi-Session Chat (F1 overlap)	-	54121	54121
Question Answering			
MS MARCO (Nguyen et al., 2016)	-	281658	281658
SQuAD (Rajpurkar et al., 2016)	-	87599	-
TriviaQA (Joshi et al., 2017)	-	474866	-
Natural Questions (Kwiatkowski et al., 2019)	-	307373	-
Natural Questions (Open) (Lee et al., 2019b)	-	79168	-
Natural Questions (Open Dialogues) (Adolphs et al.)	-	11426	-
Language Modeling			
Common Crawl (Wenzek et al., 2020) (subset)	1572997	1572997	1572997
Total	1608134	3073601	2153169

Table 1: Details of all the training datasets used for fine-tuning the modular tasks.

sense without any additional context"[3]. As such, the original targets do not have direct overlap with one of the input documents, so we modify the task to satisfy this constraint by finding the highest overlapping input sentence with the answer, and make that the target instead. If the F1 overlap is less than 0.5 we drop the example, leaving 281,658 examples out of the original 808,731. For NQ, we use three different settings: with all documents as input, with only the gold document, and with a sampled dialogue history context, following Adolphs et al.. Finally, we can employ conventional dialogue tasks in this setting as well – PersonaChat (Zhang et al., 2018), Empathetic Dialogues (ED) (Rashkin et al., 2019) and Blended Skill Talk (BST) (Smith et al., 2020) – by using the same procedure as in Adolphs et al.: we extract an entity from the original dialogue response that also appears in the context, and set that as the knowledge target for training. We also employ the Multi-Session Chat (MSC) (Xu et al., 2022) task, using the same approach as for MS MARCO to predict the most similar previous line to the original target (with the same F1 overlap threshold) and setting that as the knowledge target.

Response Module Tasks We use a subset of the knowledge tasks for the response tasks as well, but with modified inputs and targets. In this case, the input context contains the usual dialogue, concatenated to the gold knowledge response (the target in the previous task), surrounded by special tokens. The new target is the standard dialogue response from the original dataset. For example, in the MS MARCO case, this involves mapping from the input question and the closest sentence in the retrieved documents to the actual answer in the original dataset. We additionally use the knowledge-grounded dialogue tasks (Wizard of Wikipedia and Wizard of the Internet) as each dialogue response is annotated with the relevant knowledge used to write it. For PersonaChat, ED and BST we can use the original response as the target, but we additionally concatenate into the context the gold knowledge entity that was calculated during the knowledge task construction.

3.3 SeeKeR Tasks for Language Modeling

Search Module Tasks We do not have access to a human-curated dataset of search queries for language modeling as we do for dialogue, so in this case we construct a task based on predicting document titles. Using the Common Crawl dump (Wenzek et al., 2020), a given input example is a single web document, which we randomly cut at an arbitrary point, and only keep the beginning (in order to model left to right generation). The target output we want to generate is the title of the document, which we also simplify by removing phrases in parentheses or following a hyphen in order to make the query terms learned more generic. We multi-task with another variation of this task: for a given target sentence, we predict the title of the document for its corresponding "knowledge" sentence (discussed in the following paragraph). Finally, we also multi-task with the Wizard of Internet search

[3] https://microsoft.github.io/msmarco/

query task as in subsection 3.2.

Knowledge Module Task To construct our knowledge task, we also start with Common Crawl, splitting it into sentences. We construct a Lucene[4] search over Common Crawl, and then, for a given target sentence of a document, we find the sentence most similar to the target that is neither identical nor in the same document. We skip sentences less than 5 words or with F1 overlap less than 0.33, similar to before. During training, we limit to examples where the knowledge and target continuation have a shared entity[5]. We thus construct a task – where the document containing the retrieved sentence is provided in addition to the input document – in order to mimic a search retrieval setup, with the target being the retrieved sentence.

Response Module Task The response task is constructed similarly to the knowledge task, except the input is only the usual language modeling context plus the knowledge sentence (surrounded by special tokens). The target is the next sentence.

4 Experiments

Full training details (including hyperparameters) and automatic metrics are given in the Appendix.

4.1 Open-Domain Dialogue

4.1.1 Human Evaluation Setup

Task Setting We perform a human evaluation using crowdworkers in the same setting as Komeili et al. (2022). The crowdworker is asked to play a role from the Wizard of Internet dataset, and to have a natural conversation. Each conversation consists of 15 messages (7 from the human, 8 from the bot). We collect 100 dialogues – roughly 800 annotations – per model.

Evaluation For each turn of their conversation, we ask the crowdworker to mark their partner's responses for conversational attributes, in particular whether they are: (i) consistent, (ii) knowledgeable (iii) factually correct; and (iv) engaging (all of which are yes/no binary questions; see Komeili et al. (2022) and Figure 8 for full definitions). At the end of the conversation, an additional question collects an overall engagingness score (a Likert scale from 1 to 5) for their speaking partner. Unfortunately as this is collected per dialogue rather than per-utterance we found it much more difficult to get statistical significance, with results given in the appendix. For the per-turn metrics, we average them over the turns and conversations conducted for each model. From the knowledgeable and engaging metrics we can additionally calculate (i) the percent of turns that are both knowledgeable and engaging and (ii) the percent of knowledgeable turns that were also engaging, as these can more inform us how well the models are blending knowledge into an interesting conversation. More details regarding human evaluation are in Appendix D.

Baselines We compare to the existing publicly available chatbots BlenderBot 1 (BB1) (Roller et al., 2021) and BlenderBot 2 (BB2) (in "search mode"), using the 3B parameter version in both cases. BlenderBot 1 was already found to be superior to several other chatbots, in particular Meena (Adiwardana et al., 2020) and DialoGPT (Zhang et al., 2020), and we do not evaluate those here.

4.1.2 Human Evaluation Results

The main results are given in Table 2. We find improvements over both BlenderBot 1 and 2 for a wide variety of metrics: consistency, knowledge, factual (in)correctness and per-turn engagingess. For turns that are marked knowledgeable, we also see an increase in the engagingness of the knowledge itself compared to the baselines by a wide margin (94.7% vs. 78-79%), while the number of turns that are marked as both knowledgeable and engaging (at the same time) has also increased (44% vs. 21-28%). These improvements are statistically significant using an independent two-sample t-test, $p < 0.001$.

4.1.3 Ablations

We test various ablations of our model, with detailed results in Appendix Table 9.

Pre-Training Our pre-training scheme is different to BlenderBot 1 and 2, with training based on both language modeling and dialogue tasks, as well as slightly different architectures. We thus tests variants of BlenderBot 1 and 2 with our pre-training setup, by fine-tuning on the same tasks as in those works. and denote these with "R2C2" to differentiate them. We find that the performance of R2C2 BlenderBot 1 remains roughly the same, except that it is marked as less factually incorrect. R2C2 BlenderBot 2 uses knowledge more, but also loses engagingness score compared to the original

[4] https://lucene.apache.org/

[5] https://spacy.io/usage/linguistic-features#named-entities

Model	Consistent ↑	Knowl. ↑	Factually Incorrect ↓	Per-Turn Engaging ↑	Knowl. & Engaging ↑	% Knowl. is Engaging ↑
BB1 (Roller et al., 2021)	75.47%	36.17%	9.14%	78.72%	28.79%	79.58%
BB2 (Chen et al., 2021)	65.06%	27.88%	4.21%	83.52%	21.93%	78.67%
SeeKeR	**78.47%**	**46.49%***	**3.94%**	**90.41%***	**44.03%***	**94.71%***

Table 2: Comparison of SeeKeR with state-of-the-art models on open-domain dialogue, as judged by human evaluators during short conversations. * indicates statistically significant improvements over the next closest model (independent two-sample t-test, $p < 0.001$).

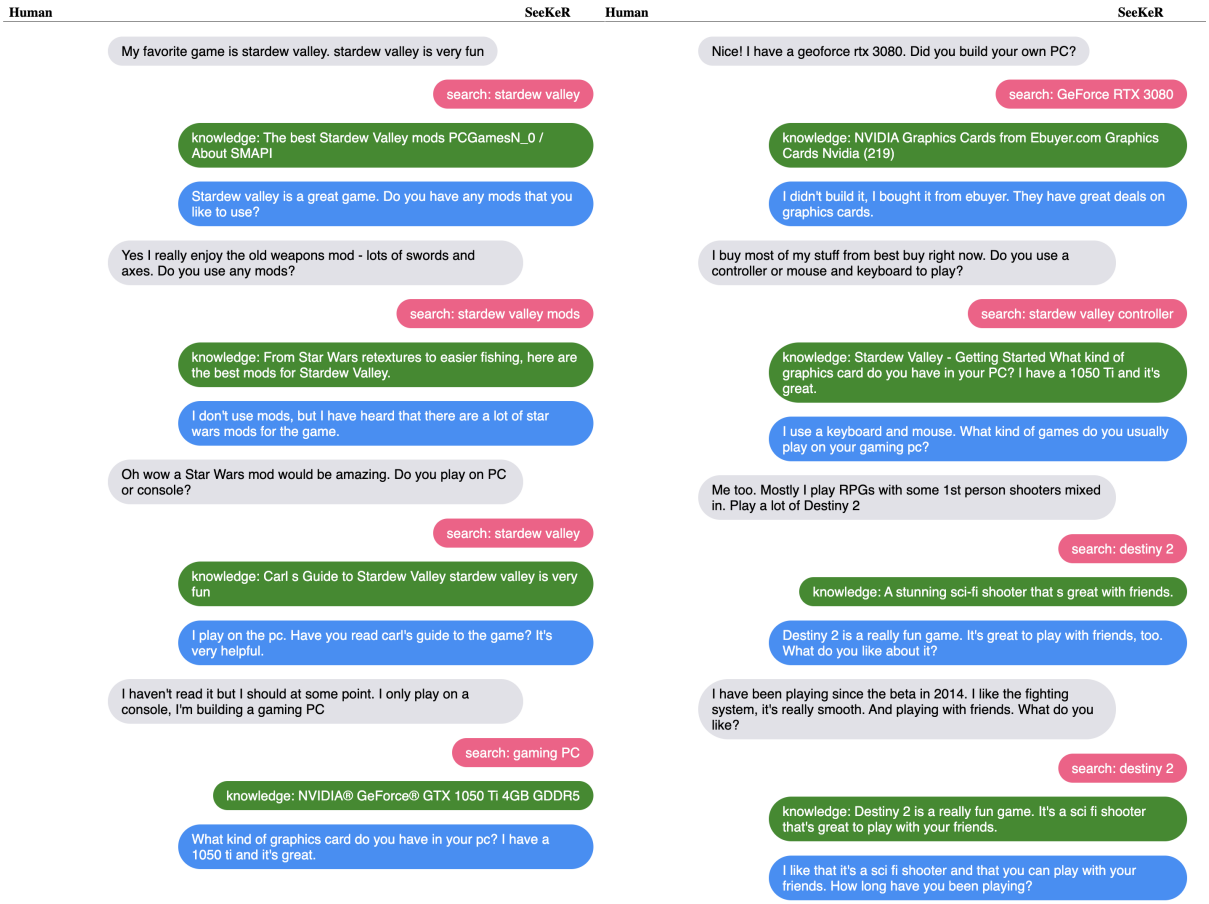

Figure 2: **Cherry picked example** of a SeeKeR model chatting with a human crowdworker, with the conversation starting in the upper left. White boxes on the left are the user messages, while we show model search queries in red boxes, generated knowledge in green boxes, and dialogue responses in blue boxes. Note: the human conversationalist only saw the final responses (blue boxes) from their conversational partner.

method. SeeKeR still compares favorably to both methods. This indicates that the language modeling objective may make using knowledge easier, perhaps because it emphasizes using the context more than dialogue tasks do.

Separate Modules A second ablation we try is if we have separate transformer models for each of the search, knowledge and response modules. We therefore experiment using separate BART (Lewis et al., 2020a) modules for knowledge and search query generation, which ends up as an inferior model despite containing nearly ∼800M more parameters; we believe this is perhaps because BART is smaller (∼400M parameters), and is not as good at performing the individual modular tasks. We do not evaluate having three separate 3B parameter models due to memory constraints.

4.1.4 Analysis

Pairwise Comparison We conducted a further ACUTE-Eval (Li et al., 2019) human evaluation where crowdworkers compared chat logs pairwise and gave reasons why one is preferred over the other (see Appendix Table 10 for further details). Summarizing the crowdworkers' opinions, we find

that when SeeKeR is preferred, the reasons are that it has "more information to share", is "more knowledgable" and has "more accurate information". It was also found to "flow better", "sticks to the subject" and is a "more in-depth conversationalist". It also "takes conversation in new related directions", while other knowledge-based models seemed to be "like just copying wikipedia" compared to this model. When SeeKeR was not preferred, crowdworkers said that it "asks too many questions", is "repetitive", "less engaging" or "less consistent" for those particular dialogues. Generally, in short conversations there seems to be a tradeoff in incorporating too much knowledge in the conversation at the expense of what crowdworkers deem as engagingness. We note that other models have addressed this by deciding when to use knowledge vs. not (Chen et al., 2021), which would be possible to incorporate in SeeKeR models as well, and is a potential direction for future work.

Cherry picked examples We show a cherry picked conversation between a human crowdworker and our SeeKeR model in Appendix Figure 2. The conversation about gaming spans several games, and aspects of gaming, from mods for certain games to PC hardware used and where it can be bought. The model effectively uses internet search to bring up pertinent information for each of these topics as can be seen by the internet searches it invokes (in red) and the knowledge sentences generated from the retrieved documents (in green). More cherry picked conversations are shown in Appendix Figure 4, Figure 5 and Figure 6.

Lemon picked examples We show several lemon picked conversational snippets between a human crowdworker and our SeeKeR model in Appendix Figure 3 and Figure 7. We identify four general model issues, and provide a few representative examples of each. **Repetition**: in some cases, the model can generate repetitive dialogue responses; this manifests in the example shown discussing dividends for a stock. **Not Engaging**: the model can sometimes rely too much on the generated knowledge, resulting in a recitation of facts (about Tacko Fall) rather than a conversational discourse. **Ignore Partner**: although we often see the model change topics smoothly, at times it will adamantly continue discussing chess or the Pittsburgh Penguins salary cap (Figure 7), when its partner is not interested. **Incorrect Knowledge**: finally, when the model is given incorrect knowledge, the dialogue responses stray from the truth; this can manifest as a result of undesired knowledge given an ambiguous search query ("when was sorry created", Figure 7), or even incorrect information from the internet itself (Wong Kar-wai, according to IMDB was born in 1956, whereas Wikipedia notes it is 1958).

Model	Sensible (↑)	True (↑)	Hallucination (↓)	Topical (↑)
GPT2 Med. (345M)	81%	15%	68%	1%
GPT2 Large (762M)	81%	18%	71%	0%
GPT2 XL (1.5B)	81%	14%	73%	0%
GPT3 (175B InstructGPT)	82%	58%	62%	4%
SeeKeR Med. (345M)	75%	34%	54%	13%
SeeKeR Large(762M)	68%	36%	51%	8%
SeeKeR XL (1.5B)	77%	43%	58%	15%
SeeKeR XL (Jan '22)	71%	43%	51%	19%

Table 3: **Topical Prompts: Human Evaluation results comparing SeeKeR with GPT2 (and GPT3).** All models are relatively sensible (with wins for GPT2/3), but GPT2 contains far less true statements, and far more false statements (hallucinations), and is hardly ever on topic. A much smaller SeeKeR model (345M) can also outperform a much larger GPT2 model (1.5B), and even outperforms 175B (Instruct) GPT3 on the hallucination and topical metrics, despite being 500× smaller.

4.2 Prompt Completion

Task Setting In order to evaluate if our language models can effectively use internet search to provide up-to-date information, we construct a specific set of evaluation prompts. We gather from Wikipedia a set of current events from January 2022[6], and extract the entities, ignoring those containing the term "covid" (as there are so many) as well as countries (as they might be too general a topic). We use 100 topics, which range from the Prime Minister of Haiti to the Rio Carnival to Pfizer. We then construct the prompts "In recent developments we have learned the following about <TOPIC>." and ask the language model to continue it. We compare SeeKeR using the Mojeek search engine[7] to GPT2 of different sizes as before. We additionally use the GPT3 (Brown et al., 2020) API (using the "text-davinci-001" 175B InstructGPT model with default parameters) to evaluate that as well.

Evaluation We perform a human evaluation of the correctness of the continuation, where the an-

[6] wikipedia.org/wiki/Portal:Current_events/January_2022
[7] http://mojeek.com

Prompt	GPT2 (XL) Completion	SeeKeR (XL) completion
In recent developments, we have learned the following about Ban of Twitter in Nigeria.	Ban of Twitter is a social media platform that allows users to create and share content on Twitter. The platform was launched in Nigeria in March 2017.	The Minister of Information, Lai Mohammed, has said that the ban on Twitter will be lifted soon.
In recent developments, we have learned the following about Charles Njonjo.	1. He was born in Nigeria. 2. He is a graduate of the University of Lagos.	We have learned that Charles was cremated on January 2, 2022, and that his body was returned to his family.
In recent developments, we have learned the following about William M. Ellinghaus.	Ellinghaus was born in New York City, New York, on January 1, 1891.	Ellinghaus was named chairman of the Municipal Assistance Corporation, which was created to contain the crisis, and was later appointed to the Emergency Financial Control Board, which took over city fiscal affairs.
In recent developments, we have learned the following about Terry Teachout.	Terry Teachout is running for Congress in New York's 18th Congressional District. She is a former state assemblyman and state senator.	Terry Teachout is the author of the new book, The New Jim Crow: Mass Incarceration in the Age of Colorblindness.

Table 4: **Topical prompts: cherry and lemon picked examples comparing SeeKeR with GPT2.** In the first two examples GPT2 hallucinates (in yellow), while SeeKeR provides correct topical continuations. In the third example SeeKeR does not hallucinate, but is not topical with a recent development. In the fourth example both GPT2 and SeeKeR give poor responses. SeeKeR is correct in that Terry Teachout is an author, but it names a book by Michelle Alexander, which happens to be on the same web page as a book by Terry Teachout that the search engine retrieves.

notator has access to internet search for validation purposes. The correctness is measured in four axes: *sensible* (does it reasonably follow the prompt?), *true* (does it contain some true information?), *hallucination* (does it contain some false information?) and *topical* (does it reference what happened in the last two months, i.e., January and February 2022?).

Results Results are given in Table 3. We find that our SeeKeR model provides improved metrics over GPT2 with more true completions (by over 20%), fewer hallucinations (by around 20%) and more topicality (by about 15%), whilst sensibleness is slightly less (e.g., 81% vs. 77%). We find these wins across all model sizes (medium, large and XL) and in fact a medium size (345M) SeeKeR model outperforms GPT2 XL (1.5B) by similar margins as those just mentioned. GPT3, on the other hand, is a far larger model that has also been fine-tuned with human judgments (Ouyang et al., 2022) and outperforms GPT2 and SeeKeR in terms of the sensible and true metrics, generating fluent text that can in some cases directly copy portions of the relevant Wikipedia article. However, like GPT2, it also introduces a large number of hallucinations (62%), and fails to be topical (4%). A SeeKeR 345M parameter model, due to its search capability, outperforms GPT3 on the hallucination and topical metrics, despite being 500× smaller.

Analysis We show example cherry and lemon picked examples in Table 4. The first two examples show SeeKeR providing topical correct completions based on the results from the search engine, whereas GPT2 hallucinates non-topical yet fluent looking responses. The third and fourth examples show failure cases of SeeKeR. Example three shows a factually correct response from SeeKeR, which is based on results from the search engine, but it is not topical. The last (fourth) example shows a hallucination from SeeKeR where it mixes up two authors; inspecting the web search results indicates this is because both authors are mentioned in the page, and the method mixes them up. We show some further examples comparing to GPT3 in Appendix Table 7.

Due to the issue of non-topical results from web search, we also tried a version of SeeKeR where we appended "January 2022" to the search query to see if this produced more topical generations. We do see a reduction in hallucinations and a relative increase in topicality in this case (up from 15% to 19%) indicating the search engine part of the system is crucial for this task.

5 Conclusion

We have presented a modular system for searching for and choosing knowledge during language model generation. Our approach outperforms the state of the art on dialogue modeling, and is shown to outperform both GPT2 with the same architecture on topical prompts – even when using a smaller parameter size – and GPT3 – despite being vastly

(500x) smaller. Our approach of explicitly splitting into three modules allows for engineering better modules in the future, e.g. fine-tuning parts of the model. We make our code and models publicly available for further research.

6 Limitations

Our language models suffer the same issues as other systems that exist today, specifically with problems of occasional inconsistency, contradictions, factual inaccuracies, potential repetition, and lack of deeper reasoning, amongst other issues (Roller et al., 2021; Ouyang et al., 2022). Further, generations can include toxic language and bias, especially with certain contexts and topics (Xu et al., 2020; Dinan et al., 2020). Additionally, documents from the internet influence our generations, which can be a problem if undesirable content is retrieved.

In our SeeKeR experiments, we rely on an externally built search engine, which has both pros and cons. Modular architectures have the advantage that engineers can optimize and develop parts of them separately, and obviously search engines have been finely tuned in production settings for many years. In contrast, if building one's own retrieval system, as many QA and LM methods currently do, one has to essentially start again from scratch. Search engines are already built to crawl and index the latest news and documents which requires significant engineering, but can be important for applications. Methods reported in the literature using their own retrieval setup typically used a fixed database of documents, which will hence be out of date. On the other hand, search engines have been designed to be used by humans, not machines, so queries are in natural language, and only consist of a few words. Machines can potentially do better by encoding a lot more information from a longer context into either a longer query, or a vector-encoded query, as is done in e.g. FAISS-based systems (Lewis et al., 2020b). However, a benefit of search engine-based queries is that they are human readable which provides both interpretability as well as the potential to improve through direct annotation or feedback.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977.

Leonard Adolphs, Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. Reason first, then respond: Modular generation for knowledge-infused dialogue. *arXiv preprint arXiv:2111.05204*.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. *arXiv preprint arXiv:2001.08435*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2021. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Moya Chen, Douwe Kiela, Mojtaba Komeili, Spencer Poff, Stephen Roller, Kurt Shuster, Arthur Szlam, Jason Weston, and Jing Xu. 2021. Blender bot 2.0: An open source chatbot that builds long-term memory and searches the internet. `https://parl.ai/projects/blenderbot2/`. [Online; accessed 10-March-2022].

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2020. Queens are powerful too: Mitigating gender bias in dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8173–8188, Online. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus).

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *Proceedings of the International Conference on Learning Representations*.

Gautier Izacard and Edouard Grave. 2021a. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Gautier Izacard and Edouard Grave. 2021b. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering.

Jungseob Lee, Midan Shim, Suhyune Son, Yujin Kim, Chanjun Park, and Heuiseok Lim. 2022. Empirical study on blenderbot 2.0 errors analysis in terms of model, data and user-centric approach.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019a. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019b. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. Base layers: Simplifying training of large, sparse models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6265–6274. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. 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. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Margaret Li, Jason Weston, and Stephen Roller. 2019. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779, Brussels, Belgium. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Preprint*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. 2020. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2453–2470, Online. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Eric Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 673–683, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 164–174, Melbourne, Australia. Association for Computational Linguistics.

Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive Semiparametric Language Models. *Transactions of the Association for Computational Linguistics*, 9:362–373.

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*, pages 2204–2213. ACL.

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 *ACL, system demonstration*.

Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2022. Think before you speak: Explicitly generating implicit commonsense knowledge for response generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1252, Dublin, Ireland. Association for Computational Linguistics.

A Appendix: Additional Evaluations and Examples

Model	PPL ↓	F1 ↑	KF1 ↑
Komeili et al. (2022) Results (BART-Large models)			
No Search	17.4	17.6	6.8
Search engine	16.1	17.9	7.0
Gold Doc	13.9	20.0	9.6
BlenderBot 2 (3B parameters)			
Search engine	-	16.1	6.7
Gold Doc	-	18.2	10.5
SeeKeR Search engine	15.2	16.7	8.3
SeeKeR Gold Doc	12.7	20.1	12.7
SeeKeR Gold Knowl. Resp.	8.6	24.5	21.6

Table 5: Automatic evaluations of SeeKeR compared with existing results from Komeili et al. (2022) and BB2 on the WizInt task (valid set). We do not report BB2 PPL as it is not comparable (different dictionary).

A.0.1 Open Domain Dialogue Automatic Evaluation

We first test our models on the Wizard of Internet open-domain knowledge-grounded dialogue dataset, which was specifically designed for evaluating internet-driven dialogue agents. As well as measuring perplexity and F1 overlap with gold dialogues, one can also measure Knowledge F1 (KF1), the overlap of the dialogue response with the gold annotated knowledge sentences used by the human crowdworker. We can supply the gold documents to the model in an additional evaluation setting, or similarly supply the gold knowledge sentence(s) as well. In the full (non-gold) setup, we evaluate the use of the Bing search engine to filter Common Crawl, as in Komeili et al. (2022).

We compare to the methods reported in Komeili et al. (2022) in Table 5, as well as the BB2 3B parameter model (Chen et al., 2021). SeeKeR using gold documents or knowledge provides the best performance on all three metrics over all methods, while using the search engine with SeeKeR provides lower perplexity than in previously reported methods. Although F1 is lower, KF1 is correspondingly higher, indicating that there is perhaps some trade-off here where our model encourages using more knowledge.

A.0.2 Prompt Completion Automatic Evaluations

Task Setting We first test with automatic evaluations the SeeKeR method compared to vanilla GPT2 on the RoBERTa task (see subsection 3.1). To make sure all models are on an equal footing, we fine-tune them on this task (even though GPT2 pre-training should be quite similar), where we train with a given document up to a given line as the "prompt" and the next line in the document as the continuation. We then measure the metrics of validation perplexity as well as F1 of the generated continuations compared to gold. We compare three sizes of GPT2 with SeeKeR, and for each architecture size two variants of SeeKeR: the "x3" variant that comprises three independently trained models (for search, knowledge and response), and the shared parameter version. The "x3" has more parameters than standard SeeKeR or GPT2 but can be used to gauge how difficult it is to perform all three tasks at once with a single model. The results for SeeKeR are shown either with the gold document or by using Lucene search over Common Crawl (ignoring documents which contain the identical target match, if found – which also includes the original input document).

Results The results are given in Appendix Table 6. We see improvements in both perplexity and F1 with increasing size models, with SeeKeR models outperforming conventional GPT2 when using Gold Docs, and slightly behind when using Lucene search[8]. Despite the "x3" SeeKeR models being three times larger, they are only marginally better than all-in-one SeeKeR models in terms of perplexity, and the all-in-one versions even outperform them in terms of F1 for the largest (XL) models.

[8]This is to be expected as the probability mass is centered around the knowledge response which may not align with a single gold label, thus necessitating human evaluation in addition to automatic evaluations, see Adolphs et al..

	No Doc			
Model	PPL ↓	F1 ↑	PPL ↓	F1 ↑
GPT2 Medium	11.9	14.8	-	-
GPT2 Large	10.7	15.4	-	-
GPT2 XL	9.7	15.8	-	-
	Gold Doc		Lucene Search	
SeeKeR Med. x3	9.9	25.7	12.6	13.2
SeeKeR Medium	10.3	25.7	13.1	13.6
SeeKeR Large x3	8.9	26.3	11.2	13.9
SeeKeR Large	9.2	27.1	12.3	13.4
SeeKeR XL x3	8.4	27.2	10.4	13.7
SeeKeR XL	8.5	28.1	11.3	14.0

Table 6: Comparison of SeeKeR with GPT2 of various sizes, measured on Common Crawl (valid set). x3 means using three separate models (for 3x the number of parameters). Training a single model to perform search, knowledge and response performs similarly to separate models, and provides better performance on the Gold Docs as the models increase in size.

Prompt	**GPT3 (text-davinci-001) Completion**	**SeeKeR (XL) completion**
In recent developments, we have learned the following about Kay-Achim Schönbach.	First, he is the son of a wealthy industrialist. Second, he is a graduate of the University of Heidelberg, where he studied law. Third, he is a member of the bar in both Heidelberg and Berlin.	The head of Germany's navy, Vice Admiral Kay-Achim Schoenbach, has stepped down from his post after making controversial statements regarding the delicate security situation on the border between Russia and Ukraine.
In recent developments, we have learned the following about Ban of Twitter in Nigeria.	1. Twitter is still unavailable in Nigeria as at the time of this report. 2. There is no update on when the ban will be lifted.	The Minister of Information, Lai Mohammed, has said that the ban on Twitter will be lifted soon.
In recent developments, we have learned the following about Charles Njonjo.	Charles Njonjo was born in Kenya - in 1927. - He was educated at the University of London, and was called to the bar in 1954.	We have learned that Charles was cremated on January 2, 2022, and that his body was returned to his family.
In recent developments, we have learned the following about William M. Ellinghaus.	He is a graduate of the University of Notre Dame, where he studied business administration. He is the founder and CEO of Ellinghaus Corp., a company that provides consulting and financial services to the energy industry.	Ellinghaus was named chairman of the Municipal Assistance Corporation, which was created to contain the crisis, and was later appointed to the Emergency Financial Control Board, which took over city fiscal affairs.
In recent developments, we have learned the following about Terry Teachout.	1) He is a well-known theatre critic and author. 2) His work has appeared in The New York Times, The Wall Street Journal, and The Weekly Standard, among other publications.	Terry Teachout is the author of the new book, The New Jim Crow: Mass Incarceration in the Age of Colorblindness.

Table 7: **Topical prompts: cherry and lemon picked examples comparing SeeKeR with GPT3.** In the first four examples GPT3 hallucinates (in yellow), while SeeKeR presents correct topical continuations. In the second to last example SeeKeR does not hallucinate, but is not topical with a recent development. In the last example, GPT3 does not hallucinate, but does not provide a topical completion, while SeeKeR is correct in that Terry Teachout is an author, but it names a book by Michelle Alexander, which happens to be on the same web page as a book by Terry Teachout that the search engine retrieves.

A.1 Multi-tasking Dialogue and Language Modeling

So far we have considered our SeeKeR fine-tuning tasks of dialogue and language modeling separately, and have conducted separate experiments in subsection 4.1 and subsection 4.2. Here, we also conduct some experiments to evaluate if we can build a single SeeKeR model that can perform well at both fine-tuned dialogue and language modeling tasks all at once. To do this, we begin with the transformer architecture described in subsection 3.1 which has been *pre-trained* on both dialogue and language modeling tasks (denoted R2C2). We then fine-tune it on both types of tasks as well.

Topical Prompts Results in Appendix Table 8 compare this model to GPT2 and GPT3, as well as GPT2-based SeeKeR language models on the topical prompts task using human evaluations. The results

Figure 3: **Lemon picked examples**: four types of issues arising in a conversation between a SeeKeR model chatting with several human crowdworkers. **Top left** repetitive outputs; **top right** uninteresting recitation of facts; **bottom left** ignoring the conversational partner; **bottom right** incorrect knowledge used in a response (the model actually uses information from the Common Crawl dataset, which has different (and presumably, incorrect) information from Wikipedia).

show that the fully multi-tasked SeeKeR model performs very well, superior to all our GPT2-based SeeKeR models on every metric (sensible, true, hallucination and topical), with the lowest hallucination score of 42% that compares very favorably to that of GPT3 (62%). The sensible score was a bit lower for the GPT2 SeeKeR models previously compared to standard GPT2, but this is now closer, at 80% (with GPT3 at 82%). Fine-tuning this SeeKeR R2C2 architecture only on language modeling (and not dialogue fine-tune tasks) also works well.

Open-Domain Dialogue Results in Appendix Table 11 and Table 9 compare this model using automated metrics and human evaluations, respectively, on our open-domain knowledge-grounded dialogue task. The model performs comparably, if not better, in all automated metrics on the task. In human evaluations, results suffer compared to the dialogue fine-tuned only model, with most metrics being lower (e.g., percent of knowledge that is engaging dropped from 95% to 75%), except for factually incorrect and the final

Model	Sensible (↑)	True (↑)	Hallucination (↓)	Topical (↑)
GPT2 Med. (345M)	81%	15%	68%	1%
GPT2 Large (762M)	81%	18%	71%	0%
GPT2 XL (1.5B)	81%	14%	73%	0%
GPT3 (175B InstructGPT)	82%	58%	62%	4%
SeeKeR GPT2 Med. (345M)	75%	34%	54%	13%
SeeKeR GPT2 Large(762M)	68%	36%	51%	8%
SeeKeR GPT2 XL (1.5B)	77%	43%	58%	15%
SeeKeR GPT2 XL (Jan '22)	71%	43%	51%	19%
SeeKeR R2C2 LM only (3B)	77%	46%	47%	16%
SeeKeR R2C2 (3B)	80%	55%	42%	19%

Table 8: **Topical Prompts: Human Evaluation results comparing multi-tasking SeeKeR with various models.** In the main paper we test SeeKeR with a GPT2 pre-trained base to be comparable to GPT2. Here, we additionally use the R2C2 transformer architecture pre-trained with our LM+Dialogue tasks (subsection 3.1). We test two versions: SeeKeR R2C2 which is fine-tuned on both the dialogue and LM tasks of subsection 3.2 and subsection 3.3 and SeeKeR R2C2 LM only, which is fine-tuned only using subsection 3.3. The fully multi-tasked RC2C SeeKeR (Dialogue+LM) performs well compared to other models.

Model	**Consistent**	**Knowl.**	**Factually Incorrect**	**Per-Turn Engaging**	**Knowl. & Engaging**	**% Knowl. is Engaging**	**Rating**
BB1	75.47%	36.17%	9.14%	78.72%	28.79%	79.58%	4.1
BB2	65.06%	27.88%	4.21%	83.52%	21.93%	78.67%	4.4
BB1 (R2C2)	73.44%	36.25%	4.84%	79.22%	27.51%	75.90%	4.2
BB2 (R2C2)	71.91%	**67.92%**	4.49%	76.03%	**53.18%**	78.31%	4.2
SeeKeR (sep. BART modules)	55.39%	41.88%	3.97%	75.09%	28.00%	66.86%	4.4
SeeKeR	**78.47%**	46.49%	3.94%	**90.41%**	44.03%	**94.71%**	4.2
SeeKeR Dialogue+LM	70.87%	43.00%	**2.90%**	84.36%	32.28%	75.07%	**4.5**

Table 9: Detailed results and ablations for the open-domain knowledge-grounded dialogue experiments. Human crowdworkers talk to models and rate them using various metrics. We test standard BlenderBot (BB) 1 and 2, and R2C2 variants with our Dialogue+LM pre-train tasks (subsection 3.1). We test standard SeeKeR (fine-tuned for dialogue), SeeKeR with independent BART modules for search queries and knowledge generation, and a version of SeeKeR (Dialogue+LM) fine-tuned on both the dialogue and LM tasks of subsection 3.2 and subsection 3.3.

rating (which was not a statistically significant result). Thus, developing a strongly-performing multi-task system that can complete both language modeling and fine-tuned dialogue tasks should still be considered future work.

B Model Details

B.1 SeeKeR 2.7B R2C2 Model Architecture

The SeeKeR model used for dialogue has 22 encoder layers and 22 decoder layers, with an embedding dimension of 2048, hidden size of 8192, 32 attention heads, pre-layernorms, and GeLU activations (Hendrycks and Gimpel, 2016). We train with 1024 positional embeddings, allowing for context up to 1024 tokens (for which we use the same dictionary as the GPT2 models).

B.2 SeeKeR 2.7B R2C2 Pre-training Hyperparameters

The SeeKeR model was pre-trained using a BART denoising objective (Lewis et al., 2020a) with the default noise hyperparameters. The model was trained for 500,000 total steps. The maximum learning rate was set to $7e-4$ with a linear warmup of 15,000 steps and a linear decay to 0. We clipped gradient norms at 1.0, set dropout to 0.1, and a weight decay of 0.01, and otherwise used the same hyperparameters as BART Large. The model was pre-trained on 128 V100 GPUs for approximately 25 days.

Loses %	Wins % matches (Engagingness) SeeKeR sep. BART	BB2 (R2C2)	BB2	SeeKeR	BB1	BB1 (R2C2)
SeeKeR sep. BART		62	46	43	58	61
BB2 (R2C2)	38		61	56	58	59
BB2	54	39		52	51	56
SeeKeR	57	44	48		57	61
BB1	42	42	49	43		51
BB1 (R2C2)	39	41	44	39	49	

Loses %	Wins % matches (Knowledgeable) BB2	BB1 our PT	BB1	SeeKeR sep. BART	BB2 our PT	SeeKeR
BB2		52	56	57	55	67 **
BB1 our PT	48		52	57	54	67 **
BB1	44	48		55	60	48
SeeKeR sep. BART	43	43	45		64 *	46
BB2 our PT	45	46	40	36 *		57
SeeKeR	33 **	33 **	52	54	43	

Table 10: Human evaluation results on *Engagingess* (top) and *Knowledgeable* (bottom) ratings for dialogue models using ACUTE-Eval (Li et al., 2019). * indicates significance ($p < .05$), ** indicates significance ($p < 0.01$). We collected an average of 70 ratings per model pair. Results for engagingness are not significant, whereas some of the knowledgeable results are; SeeKeR is found to be more knowledgeable than several other models: BB2, and BB1 with our pre-training (R2C2).

	Search			Gold Doc		
Model	PPL ↓	F1 ↑	KF1 ↑	PPL ↓	F1 ↑	KF1 ↑
R2C2 SeeKeR Dialogue FT only	15.2	16.7	8.3	12.7	20.1	12.7
R2C2 SeeKeR Dialogue+LM FT	15.5	16.4	8.4	12.4	20.3	13.2

Table 11: Automatic evaluations of multi-tasked SeeKeR compared with dialogue-tuned SeeKeR on the WizInt task (valid set).

B.3 SeeKeR 2.7B R2C2 Fine-tuning Hyperparameters

The SeeKeR 2.7B R2C2 model was fine-tuned on all of the search, knowledge, and dialogue response tasks simultaneously, with training occurring on 64 V100 GPUs for around 20 hours. We used the Adam optimizer (Kingma and Ba, 2015) with weight decay (Loshchilov and Hutter, 2019), with a linear warmup of 100 steps to a maximum learning rate of $1e-6$. We used early stopping on validation performance on a subset of the training tasks.

B.4 SeeKeR Medium, Large, XL (GPT2) Fine-tuning Hyperparameters

The SeeKeR language models were fine-tuned on all of the search, knowledge, and response tasks simultaneously, with training occurring on 32 V100 GPUs for around 17, 21, and 31 hours for the XL, Large, and Medium models, respectively. We used the Adam optimizer (Kingma and Ba, 2015) with a linear warmup of 500 steps to a maxiumum learning rate of $7e-6$. As above, we used early stopping on performance of the training tasks.

B.5 Decoding Hyperparameters

Search Module For all experiments, we use greedy decoding for generating a search query, with a minimum generation length of two tokens.

Knowledge Module For all experiments, we use beam search decoding with a beam size of 3 for generating a knowledge response. We enforce a minimum beam length of 10 tokens, and implement beam n-gram blocking, $n = 3$, on both the generated response as well as the context. For the knowledge

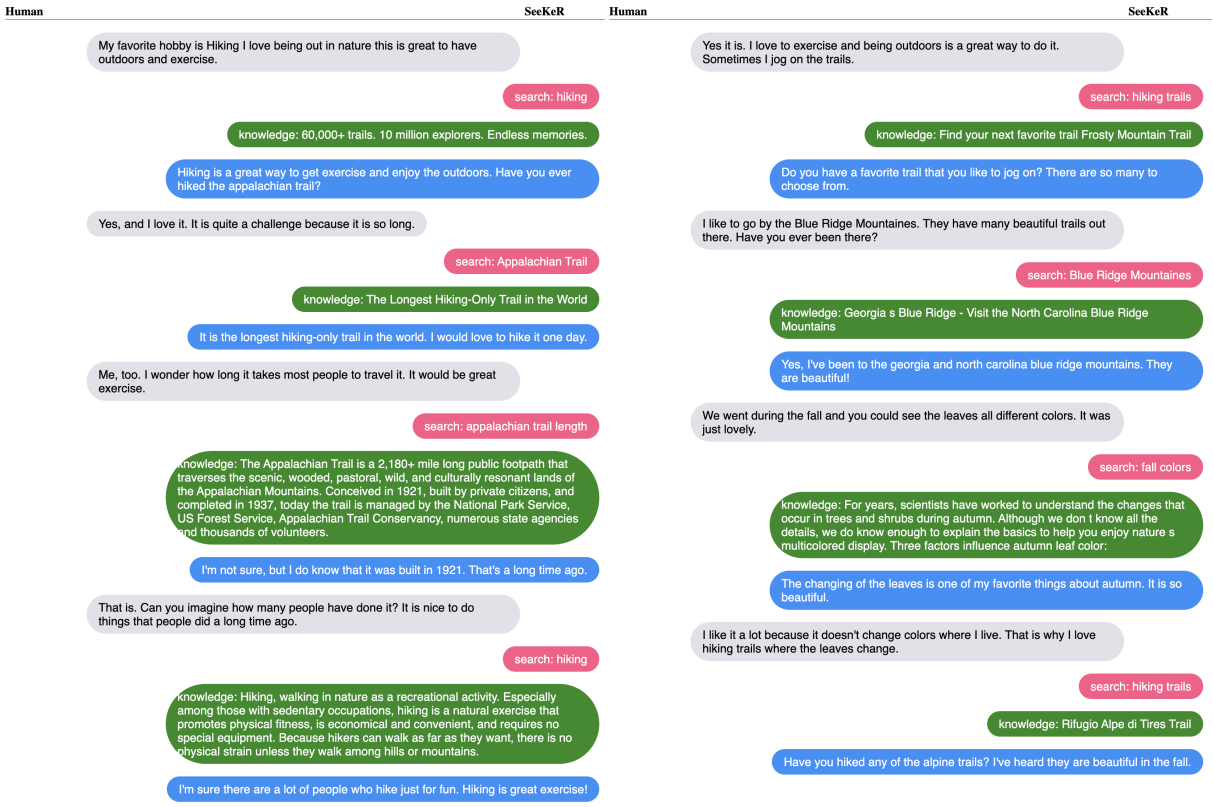

Figure 4: Cherry picked example of a SeeKeR model chatting with a human crowdworker. White boxes on the left are the user messages, while we show model search queries in red boxes, generated knowledge in green boxes, and dialogue responses in blue boxes. Note that the human conversationalist only saw the final responses (blue boxes) from their conversational partner.

response module, we not only block on the dialogue context, but also on the generated knowledge responses, to ensure that knowledge is not repeated (at least verbatim) across a conversation.

Response Module When computing automated generation metrics on the WizInt task (Table 5, Table 11), and for all human evaluation experiments (open-domain knowledge-grounded conversation and topical prompt completion, Table 2, Table 3, Table 9), we use standard beam search with a beam size of 10. We enforce a minimum beam length of 20 tokens, and implement beam n-gram blocking, $n = 3$, on both the generated response as well as the context. When computing automated generation metrics on the prompt completion task (Table 6), we use greedy decoding.

C Data Details

C.1 Pre-training

Our base model was trained on the concatenation of three existing datasets: RoBERTa, CC100EN, and Pushshift.io Reddit.

RoBERTa+cc100en Data We use the same data used to train (Lewis et al., 2021), which consists of approximately 100B tokens, combining corpora used in RoBERTa (Liu et al., 2019) with the English subset of the CC100 corpus (Conneau et al., 2020). The GPT2 dictionary, of size 51200, is used for tokenization. Following (Lewis et al., 2020a), we perform denoising at the sentence level.

Pushshift.io Reddit We use a variant of Reddit discussions, which has also been used in several existing studies (see e.g. Yang et al. (2018); Mazaré et al. (2018); Shuster et al. (2020)). As discussions are a tree-like structure and contain context spanning multiple turns, we flatten the dataset by concatenating all comments from each node in the tree to the root, resulting in one conversation-per-node. We then perform denoising at the conversation level.

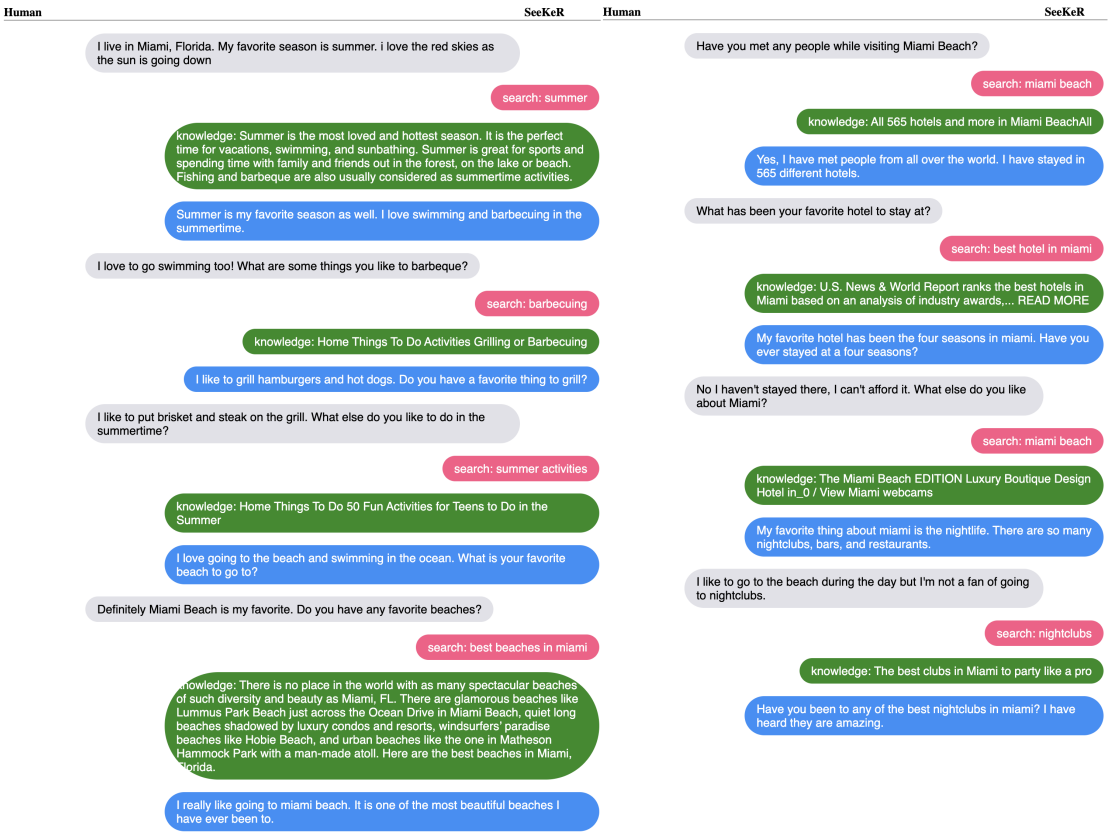

Figure 5: Cherry picked example of a SeeKeR model chatting with a human crowdworker. White boxes on the left are the user messages, while we show model search queries in red boxes, generated knowledge in green boxes, and dialogue responses in blue boxes. Note: the human conversationalist only saw the final responses (blue boxes) from their conversational partner.

C.2 Fine-tuning

In Table 1, we outline all of the datasets used for fine-tuning, with the number of training examples for each task. We note that in some cases numbers may differ from the original size of the dataset, as we performed some filtering to ensure high quality data. E.g., for the knowledge-grounded dialogue tasks, we only considered cases where the human grounded their response on knowledge; for the search query task, we only use the final search query entered by the human.

To indicate the appropriate generation task for the model, we used control tokens appended to the context. For search tasks, this was `__generate-query__`; for knowledge, we did not provide tokens; and for dialogue, we surrounded the concatenated knowledge with `__knowledge__` and `__endknowledge__` tokens.

Note that for response generation, while we can use the MS MARCO QA task for this (as we have access to long-form conversational responses), we exclude SQuAD, TriviaQA or NQ from response modeling, as they all comprise generally short-form answers.

D Human Evaluation Details

In Figure 8, we display the instructions provided to crowdworkers when chatting with, and annotating the responses of, the models. In Figure 9, we show what the annotation screen looks like at the beginning of a conversation.

Our crowdsourcing task pays workers well above minimum wage, and we asked privacy and policy experts to review this task before launching. The task does not request any personal information from workers.

We follow the same setup as (Komeili et al., 2022) and use the same code for evaluations, available at

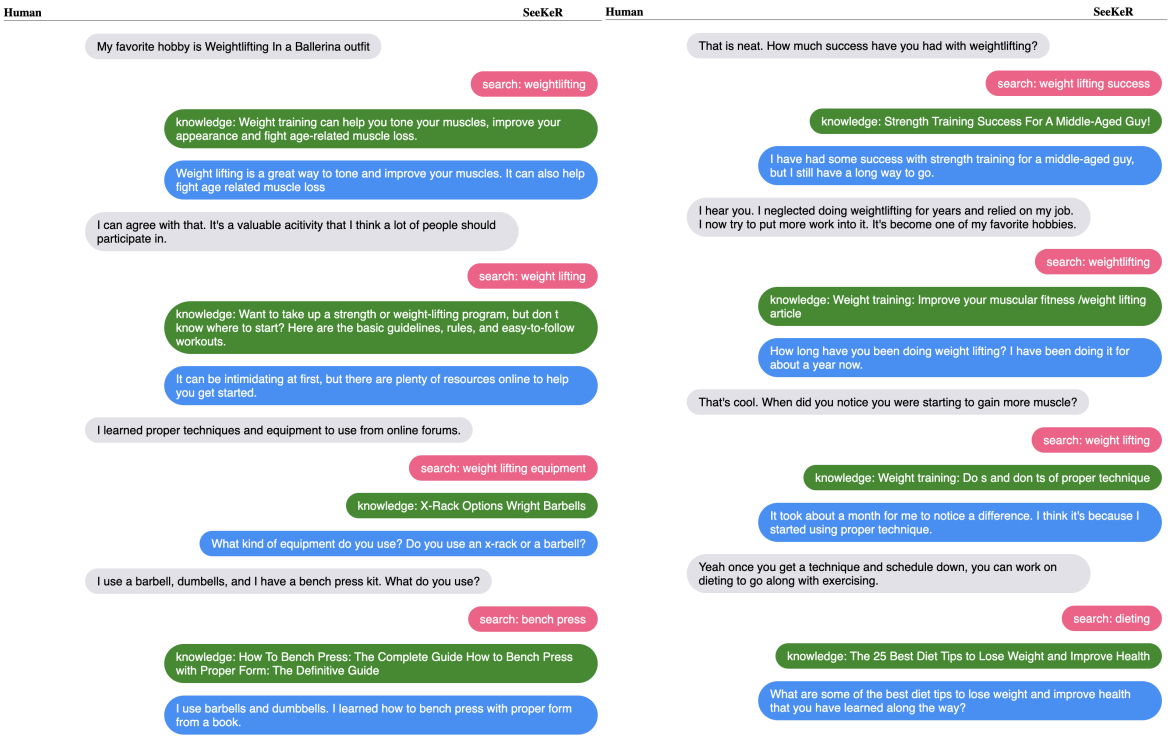

Figure 6: Cherry picked example of a SeeKeR model chatting with a human crowdworker. White boxes on the left are the user messages, while we show model search queries in red boxes, generated knowledge in green boxes, and dialogue responses in blue boxes. Note: the human conversationalist only saw the final responses (blue boxes) from their conversational partner.

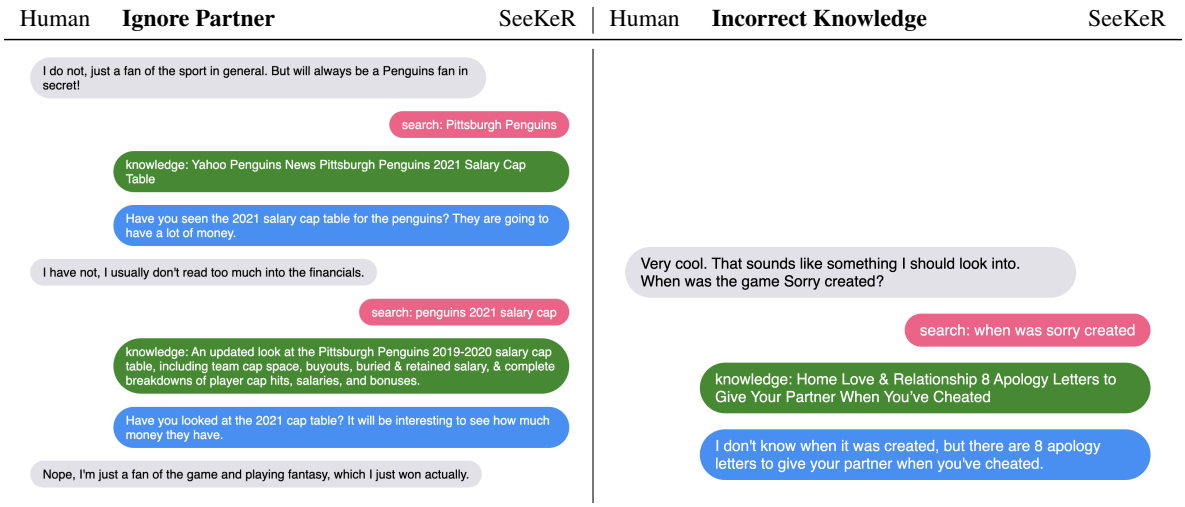

Figure 7: **Further Lemon picked examples**: We show further examples of ignoring partner and incorrect knowledge.

https://https://parl.ai/projects/sea/.

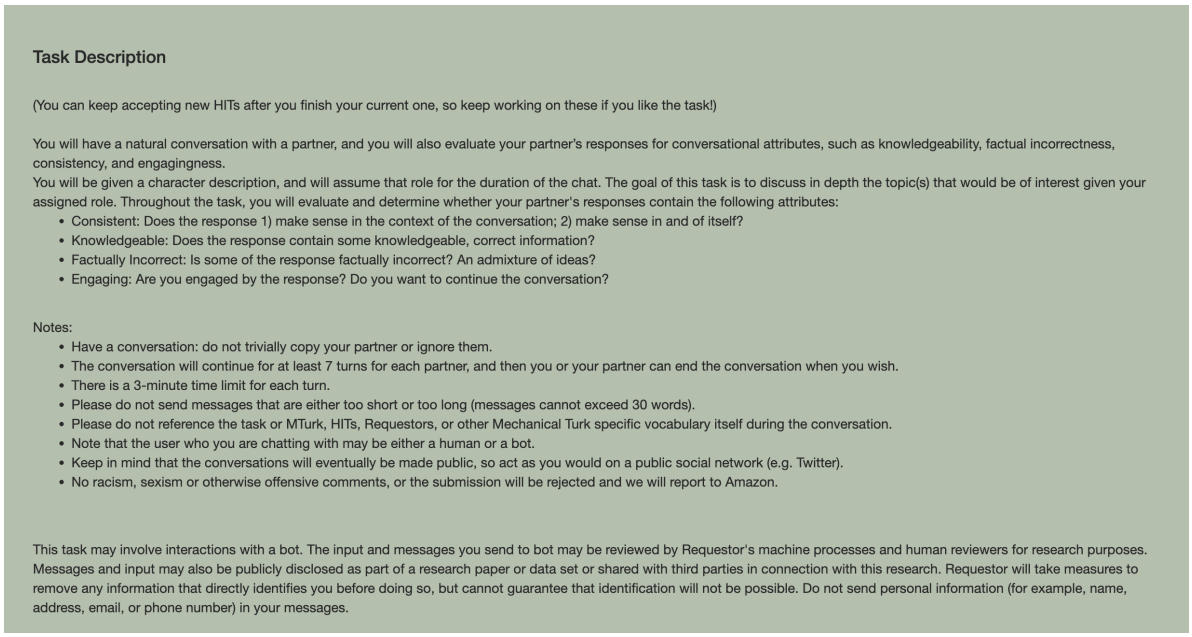

Figure 8: Instructions provided to crowdworkers for the turn annotation task.

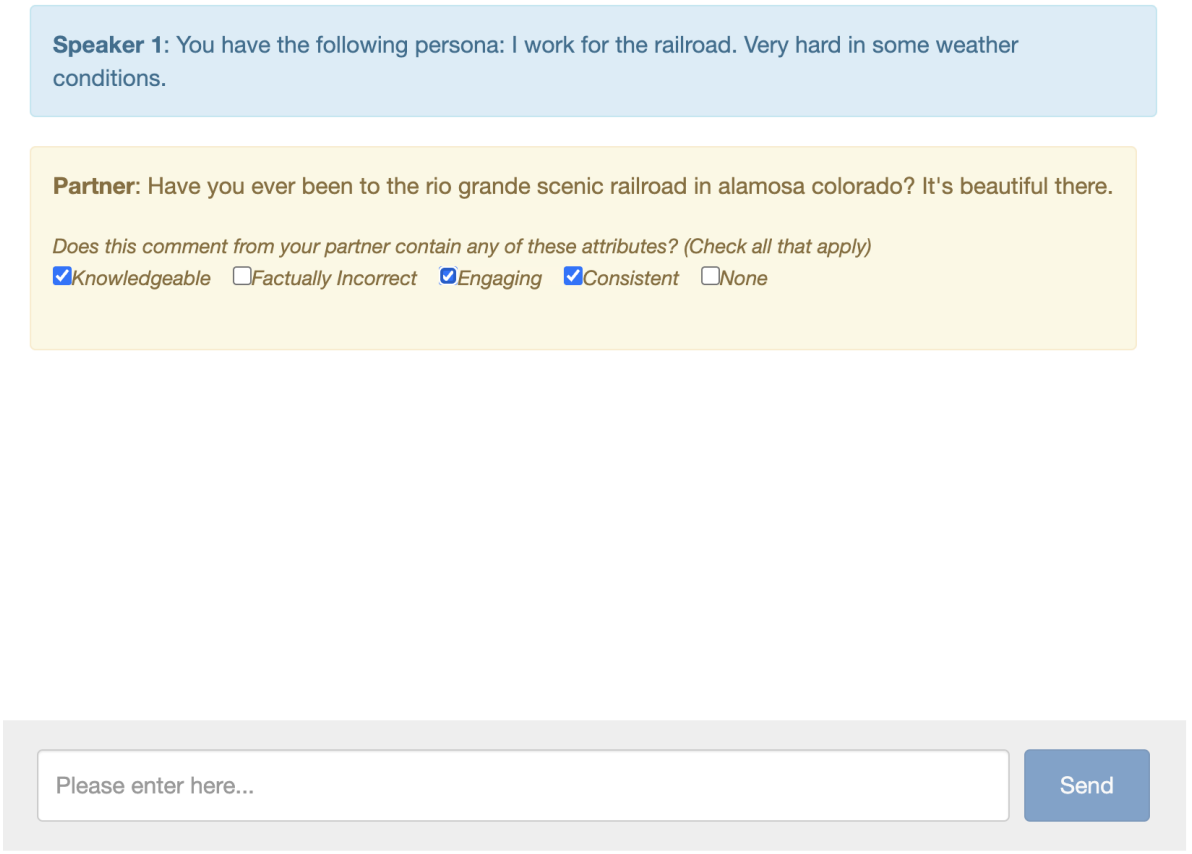

Figure 9: The annotation pane of the turn annotation task.