Internet-Augmented Dialogue Generation

Mojtaba Komeili
Facebook AI Research Labs
komeili@fb.com

Kurt Shuster
Facebook AI Research Labs
kshuster@fb.com

Jason Weston
Facebook AI Research Labs
jase@fb.com

Abstract

The largest store of continually updating knowledge on our planet can be accessed via internet search. In this work we study giving access to this information to conversational agents. Large language models, even though they store an impressive amount of knowledge within their weights, are known to hallucinate facts when generating dialogue (Shuster et al., 2021); moreover, those facts are frozen in time at the point of model training. In contrast, we propose an approach that learns to generate an internet search query based on the context, and then conditions on the search results to finally generate a response, a method that can employ up-to-the-minute relevant information. We train and evaluate such models on a newly collected dataset of human-human conversations whereby one of the speakers is given access to internet search during knowledge-driven discussions in order to ground their responses. We find that search-query based access of the internet in conversation provides superior performance compared to existing approaches that either use no augmentation or FAISS-based retrieval (Lewis et al., 2020b).

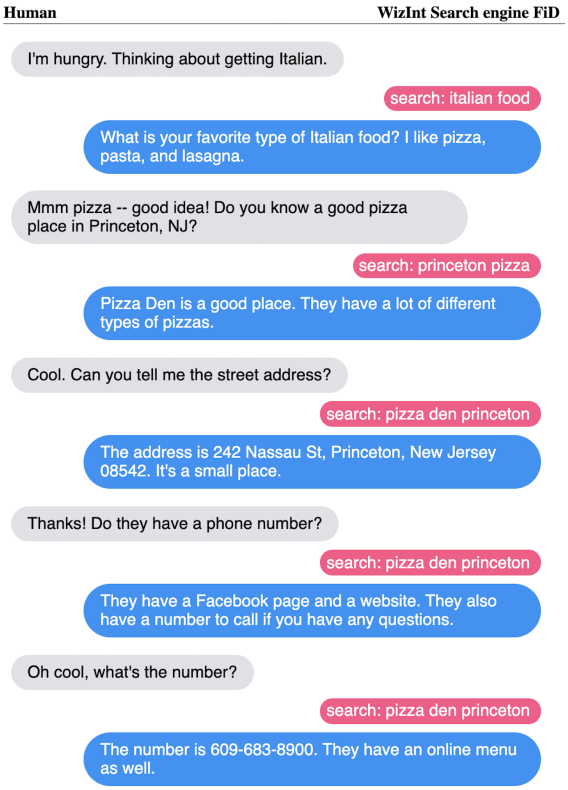

Figure 1: Cherry picked example of a model with internet-augmentation trained on our new Wizard of the Internet task. The model is able to correctly suggest a pizza place in Princeton, complete with its address and phone number, by searching the internet.

1 Introduction

Open-domain dialogue, which involves chat about any topic, rather than a specific goal-directed topic, is commonly studied by training large language models (Adiwardana et al., 2020; Zhang et al., 2020; Roller et al., 2021). These models are trained either in a encoder-decoder or decoder only setting on large datasets of human-human conversations, and any knowledge obtained during training is stored in the weights of the model. Such static language modeling fails to take into account the dynamic state of the world, where new information is coming in by the day – or even by the minute – as the knowledge in static models is gleaned from the point in time when the dataset was collected, and then frozen into the model that is trained; see (Lazaridou et al., 2021) for criticisms of this approach. Further, static language models are known to *hallucinate*, that is they generate plausible looking statements that are factually incorrect, which can be interpreted as a form of lossy compression when employing training to encode that knowledge within the weights of a neural network; see Shuster et al. (2021) for an in-depth study.

In this work we study generative models that are instead capable of accessing the vast knowledge of the internet dynamically in order to inform their responses. Utilizing encoder-decoder archi-

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8460 - 8478

tectures, we consider models that, given a dialogue context, first generate a search query. The queries are then used to retrieve relevant knowledge that is prepended to the conversational history, which is encoded using the Fusion-in-Decoder method (Izacard and Grave, 2021). Taking into account this encoded knowledge, a response is finally generated using the decoder. This ability to access the internet means the model is always up-to-date, unlike existing models that only know about facts in their fixed training set. Our model, in contrast, can potentially make use of the latest sports scores, movies or TV shows that were just released, the latest reviews, and so forth – amongst the countless other topics available on the internet.

In order to train and evaluate such models, we collect a new crowdsourced English dataset involving human-human conversations, where one of the workers plays the role of a "wizard" who conducts internet searches in order to inform their responses during knowledge-grounded conversations. We show that internet-augmented models trained to replace the human wizard outperform conventional non-augmented generation models on this task as measured by automatic metrics as well as human evaluations, and with our search query generation based approach also outperforms existing retrieval-augmented FAISS-based approaches such as RAG (Lewis et al., 2020b) and FiD-RAG (Shuster et al., 2021). We make our final models and the new task we have collected, publicly available.[1].

2 Related Work

The majority of work on dialogue generation has focused on training on natural or crowdsourced data where the task is, given a dialogue context (history), to generate the next response. Datasets such as pushshift.io Reddit (Baumgartner et al., 2020), PersonaChat (Zhang et al., 2018) or Empathetic Dialogues (Rashkin et al., 2019) (see Huang et al. (2020) for a review) are typically employed to train the weights of a Transformer encoder-decoder. This is the standard approach in state-of-the-art chatbots such as Meena (Adiwardana et al., 2020) or BlenderBot (Roller et al., 2021). Such models do not augment their generations with access to external knowledge, instead relying on facts originally provided in the training datasets themselves being stored into the weights of the model.

A growing area of research is that of augmenting generative models with external knowledge. Earlier works such as Memory Networks (Weston et al., 2015) and DrQA (Chen et al., 2017) utilized TFIDF-based retrieval over documents to provide additional input to neural models for the task of question answering, following the well studied area of non-neural methods that use retrieval for QA (Voorhees and Tice, 2000). More recently, the RAG (Retrieval-Augmented Generation) (Lewis et al., 2020b) and FiD (Fusion-in-Decoder) (Izacard and Grave, 2021) models developed these ideas further, using a neural retriever as well, with superior results. Retrieval-augmentation is also studied in the area of language modeling, where it is used for pre-training (Guu et al., 2020), and as a memory (Yogatama et al., 2021), especially using k-nearest neighbor-based cache models (Khandelwal et al., 2021, 2020; Grave et al., 2017; Merity et al., 2017).

In dialogue, knowledge grounding is becoming more popular an area, with several datasets developed to study it (Zhou et al., 2018; Dinan et al., 2019; Ghazvininejad et al., 2018; Gopalakrishnan et al., 2019; Galetzka et al., 2020). Some of these such as Topical-Chat (Gopalakrishnan et al., 2019) and CMU_Dog (Zhou et al., 2018) are constructed given a gold passage of knowledge, and the task analyzes whether the model can use this knowledge in dialogue. Other works (Zhao et al., 2020; Kim et al., 2020; Bruyn et al., 2020) study whether knowledge selection is possible from a (small) set of knowledge. However, a retrieval step (or search engine) is not used, as we consider here.

Perhaps the closest to our work is the Wizard of Wikipedia task (Dinan et al., 2019) which involves conversations grounded in Wikipedia, using a TFIDF retrieval model to find relevant knowledge from that database. Our work can be seen as a much richer task, covering all of the information that is publicly available on the internet and hence a more diverse range of conversational topics rather than just Wikipedia, while allowing human wizards to search for relevant knowledge themselves. Moreover, we consider sophisticated neural-in-the-loop retrieval mechanisms and real search engines. Shuster et al. (2021) studied neural-retriever-in-the-loop methods on this dataset.

3 Internet-Augmented Generation

We consider two ways to access the webpages from the internet: (i) using a cached set of pages that are stored in a distributed approximate nearest-

[1] http://parl.ai/projects/sea

neighbor database, FAISS (Johnson et al., 2019), or (ii) using an Internet Search Engine directly to retrieve pages. For the FAISS-based methods, there are a number of possible variants that we consider, which we will describe first.

3.1 FAISS-based methods

In our experiments, the FAISS-based methods share the same core setup. First, we store and utilize the Common Crawl dump of the internet from Wenzek et al. (2020)[2] in a FAISS database, with keys that are dense vectors. The retrieval system uses a DPR (Dense Passage Retrieval) (Karpukhin et al., 2020) Transformer-based model which scores document-context pairs in order to rank them based on their match using a bi-encoder framework, where the base DPR model is pre-trained on QA data pairs. We use the pre-trained DPR model from the KILT Benchmark (Petroni et al., 2021). The documents (webpages) are encoded using DPR into dense vectors and these are stored in the FAISS index. During dialogue-based retrieval, the dialogue context is also encoded by DPR into a dense vector and FAISS approximate nearest-neighbor lookup is performed, where the top N documents are returned. We then consider several recent neural methods for utilizing this retrieval mechanism in various ways.

RAG (Retrieval Augmented Generation) RAG (Lewis et al., 2020b) is an approach which consists of two components which are trained end-to-end: (i) the neural-in-the-loop retrieval system; and (ii) an encoder-decoder for generating final responses given the results of the retrieval. Using DPR, the top N documents are returned as described above, and in the RAG-Token model (just called RAG in the rest of the paper) each in turn is encoded along with the context for each token, and the most likely sequence is generated from the set. During backpropagation training steps, the DPR context encoder is also tuned to perform well at FAISS retrieval, but the document encodings are held fixed. This approach has been shown to optimize both retrieval and generation jointly, improving results.

FiD (Fusion in Decoder) A related, but perhaps simpler, method is that of FiD (Izacard and Grave, 2021). In this case, the pre-trained retriever is used, i.e. DPR with FAISS, and then each of the top N documents returned is prepended to the context and encoded separately by the encoder, and finally all the results are concatenated. The decoder then attends to these encodings to produce a final response, so all "fusion" happens in the decoding stage. This relatively simple method was shown to outperform RAG in some cases.

FiD-RAG The FiD approach works well, but there is no end-to-end training of the retriever in that case, and so it relies completely on being pre-trained well, as opposed to RAG which tunes the retrieval for generation. FiD-RAG, proposed in (Shuster et al., 2021) combines the two methods. First the retriever is trained in a RAG setup, and then FiD is used with that retriever. This was shown to give superior results to both RAG and FiD on dialogue tasks.

FAISS + Search Query-based Retrieval Instead of just encoding the context into a dense vector, in this approach an encoder-decoder is employed to generate a search query given the context. The search query is input into a DPR model to produce a dense vector, and is matched to documents in the FAISS index. Returned documents can then be used in the final response generation encoder-decoder as before. Any of the existing approaches (RAG, FiD or FiD-RAG) could potentially be used to fuse the DPR and generator models. We used the standard DPR FiD setup.

3.2 Search Engine-Augmented Generation

The previously described FAISS-based approaches can take advantage of many existing methods developed for QA and dialogue tasks, as we saw, but have several disadvantages. First, they may be difficult to update to real-time web documents; second, there may be a limit to the number of documents storable in local FAISS deployments; and third, such methods will not take advantage of the high quality ranking that has been finely tuned in Internet Search engines over decades of use. We thus consider using Internet search engines directly.

Method Our proposed method consists of two components: 1) A search query generator: an encoder-decoder Transformer that takes in the dialogue context as input, and generates a search query. This is given to the black-box search engine API, and N documents are returned; 2) A FiD-style

[2]We use the November 2020 dump, head only, consisting of ∼109M English webpages. Each document is split into 100-word chunks, giving 250M passages to index in FAISS. We also consider the dump of Wikipedia from (Karpukhin et al., 2020) in this work.

	Train	Valid	Test	Total
Dialogues	8,614	516	503	9,633
Utterances	82,952	5,781	4,932	93,665
Avg. Message Length	18.67	22.9	21.5	19.1
Avg. Dialogue turns	9.6	11.2	9.8	9.7
Searches	42,306	3,306	2,763	48,375
Unique URLs selected	26,192	2,087	1,973	29,500
Domains selected	10,895	1,256	1,256	11,963

Table 1: Wizard of the Internet Dataset Statistics.

encoder-decoder model that encodes each document individually, concatenates them to the dialogue context encoding, and then finally generates the next response.

We can train each of these modules separately if we have supervised data available for both tasks, the first module requiring (context, search query) pairs, and the second module requiring (context, response) pairs. As we will see, the data we collect in this work (detailed in section 4) fulfills both of these requirements.

For FiD, we try two methods: (i) Conventional FiD whereby we use the returned search results from using our trained search query generator in order to build the relevant document contexts for the FiD training set; (ii) FiD-Gold: as we will have available human-written search queries for the training set, and their corresponding search results, we can use these gold results to build training document contexts instead. Although these might not look like the queries and hence results predicted at test time, they are more likely to contain the knowledge used in generating the training set responses, thus a clearer grounding may be apparent for the model to learn correspondences.

Search Engine The search engine is a black box in this system, and could potentially be swapped out for any method. In our numerical experiments we use the Bing Search API to generate a list of URLs for each query; then, we use these URLs as keys to find their page content from a lookup table we built for our Common Crawl snapshot, in order to populate a set of pages for that query. This makes our comparison more direct with our FAISS-based methods. In addition, we can also consider if the URL is from English Wikipedia, in that case we can extract the page title from the URL and look up its corresponding page inside the dump of Wikipedia.

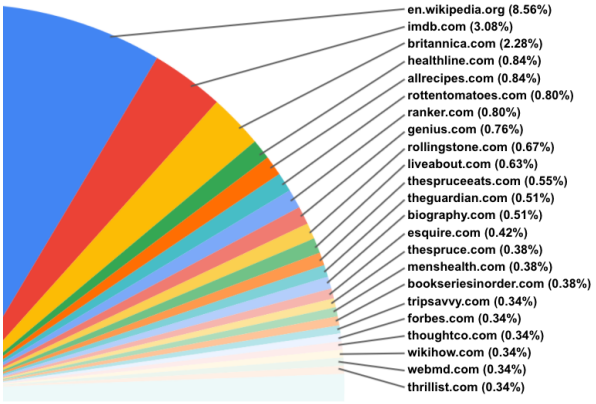

Figure 2: Breakdown of most common domains used during search by the wizard in our newly collected dataset (validation set breakdown). Shown is the most common 24.41%, there is a long tail of 1233 other domains across the whole validation set.

4 Wizard of the Internet Task

In order to both train and evaluate generative models that can use search engines in-the-loop, we design, collect and release a dataset for this purpose. The overall setup involves pairing crowdworkers that are instructed to have a conversation together. One plays the role of the *wizard*, who has access to a search engine during conversation, while the other, the *apprentice*, does not. The *apprentice* however has an assigned persona that describes their interests. The purpose of the exchange is to have an "in-depth conversation about [those] assigned interests". This mirrors conversations we expect to be more prevalent between a human and a bot: the conversations are more likely to be centered around the human's interests than the bot's, and the bot is the one that is going to be using the search engine to ground their knowledge. Hence, when we train or evaluate on this task, a given model will replace the role of the wizard.

Apprentice Persona We show the apprentice several possible persona choices for the character that they are going to play, and let them choose one, e.g. "I love tennis, Rafael Nadal is my favorite player.". The intent here is that they can choose a topic they are both more interested in themselves to talk about and also have enough knowledge of so that they can conduct a reasonable conversation. The choices we show are themselves mined from the interests provided in the existing Persona-Chat dataset (Zhang et al., 2018) and the topics given in the existing Topical-Chat dataset (Gopalakrishnan et al., 2019). More details of the choices we give

are provided in Appendix A.

Wizard Active and Passive Openings We randomize which speaker takes their turn first. If the wizard speaks first, we encourage them to start with an opening that addresses the apprentice's interests. For example, if they know their partner is interested in tennis, they could search for the latest tennis news, and open with an interesting point based on that knowledge. If the apprentice goes first, their goal is to converse with the wizard more based on their own interests, e.g. in this same case they could talk about tennis in detail.

Wizard Search At each turn, the wizard can enter free text search terms in a left-hand panel (with the main conversation panel on the right) much like in a conventional search engine. The top few results are shown in the left panel, below the search query[3]. For each document the titles are shown for space reasons, and each document is expandable. If the wizard finds one or more search results useful for their response, they can click on the sentences they find relevant, and then enter their conversational response in the right-hand panel. They are also free to try another search query if they did not find their first results appropriate, or else can enter a conversational response and choose to ignore the search results entirely.

Full System Each crowdworker has to pass an onboarding task to be able to be part of the main data collection task, and pass some automatic checks (average response length, use of search). They are asked to play a particular role ("Create an interesting character that you want to play"), and are given instructions to avoid toxic or biased language. We randomly assign for any given crowdworker a fixed choice of either wizard or apprentice for all of their data collection, otherwise we found that switching role introduced lower quality conversations, probably due to confusion between the different goals and instructions per role. After pairing, we collect between 5-6 turns (10-12 utterances) for each conversation. We ask workers to skip initial greeting messages, as these bring little extra value to the task. Screenshots of the crowdworker task can be seen in Figure 4 in the appendix. Example collected dialogues are shown in Figure 5 and Figure 6 in the appendix.

[3]We run two searches, one with the given query, and one with the query terms plus the word "news" (with the news results shown as the top two knowledge candidates), in order to encourage topical discussion.

4.1 Overall Dataset

The overall collected data consists of 9633 dialogues in total, with 82952 utterances in the training set, and validation and test sets of 5781 and 4932 utterances, respectively. Overall statistics can be found in Table 1. We find that 84.81% of all turns by the wizard involve search, so a large amount of knowledge grounding based on internet results is taking place. Of those, the wizard is allowed to repeat the search with different search terms if they did not find what they were looking for. When the wizard searches, we find 1.19 search queries are performed on average, so while mostly a single search is employed, a number of further knowledge searches are attempted. Wizards use the search results (indicated by selecting relevant sentences) 80.3% of the time.

We show in Figure 2 a breakdown of the most common domains used during search on the validation set. We see that the domains are rather diverse, coming from all kinds of topics, and in particular that the Wikipedia domain is actually fairly small (8.56% of queries), which is interesting because most other studies have used Wikipedia only as their knowledge resource (Chen et al., 2017; Lewis et al., 2020b; Dinan et al., 2019; Shuster et al., 2021). Our training set spans 26192 unique selected URLS for grounding knowledge from 10895 domains, indicating a wide variety of topics and knowledge is used across all conversations.

5 Experiments

5.1 Experiment and Evaluation Setup

We evaluate models on our new Wizard of the Internet (WizInt) task, using its dedicated training set. We also consider the existing Wizard of Wikipedia (WoW) training resource as well, either for building baselines or for multi-tasking. We consider fine-tuning various existing pre-trained models: T5 (Raffel et al., 2020), BART-Large (Lewis et al., 2020a) and BlenderBot variants (Roller et al., 2021). For all retrieval-augmented methods we use $N = 5$ returned documents. For all models, when generating responses we fix the decoding parameters to beam search (beam size 3) with a minimum sequence length of 20 and beam blocking of 3-grams within the response (but not the context), similar to choices in (Roller et al., 2021).

Model	Knowledge Access Method	Knowledge Source	PPL	F1	KF1
WoW Transformer (no knowledge)	None	None	22.3	14.7	6.7
WizInternet Transformer (no knowledge)	None	None	18.7	16.9	6.8
WoW FiD	DPR+FAISS	Wikipedia	23.0	14.7	7.4
WoW FiD	DPR+FAISS	CC	22.8	15.3	7.3
WoW FiD-RAG	DPR+FAISS	CC	22.3	15.5	7.2
WoW Search engine FiD	Bing Search	CC	21.9	14.3	7.3
WizInternet FiD-RAG	DPR+FAISS	CC	18.8	17.0	6.7
WizInternet Search term FiD	Search Query+FAISS	CC	19.0	16.5	6.7
WizInternet Search engine FiD	Bing Search	CC	17.7	16.8	6.9
WizInternet Search engine FiD	Bing Search	CC+Wikipedia	17.7	16.6	6.7

Table 2: **Results using Automatic Metrics** measured on the test set. All models use BART-Large as a base.

Model	Pre-train PPL	WizInt Validation F1	KF1
BlenderBot 2.7B	9.9	18.0	6.6
BlenderBot 400M	13.4	17.3	6.2
BART-Large 400M	17.4	17.6	6.8
T5-Large 770M	15.9	17.9	6.5
BlenderBot 2.7B†	8.1	21.7	23.3
BlenderBot 400M†	9.2	22.0	22.8
BART-Large 400M†	10.6	25.4	23.1
T5-Large 770M†	10.1	25.7	23.5

Table 3: **Choice of Pre-training Model.** We compare several pre-trained models fine-tuned on the WizInternet task, using either no or gold knowledge (models with † symbol), measured on the validation set. Perplexities cannot be compared due to differing dictionaries except between BlenderBot 2.7B and 400M.

Training Data	WoW Valid PPL	F1	KF1	WizInt Valid PPL	F1	KF1
WoW	14.8	21.0	17.7	20.4	15.8	6.7
WizInt	22.4	16.7	13.1	17.4	17.6	6.8
Both	15.4	20.0	16.3	17.3	18.0	6.9
WoW†	7.9	39.1	61.2	12.8	20.6	26.1
WizInt†	9.4	34.6	52.6	10.6	25.4	23.1
Both†	7.9	38.5	65.6	10.3	26.3	24.2

Table 4: **Usage of the Wizard of Wikipedia Dataset with Multi-Tasking** using BART-Large, measured on the validation set. Results with † symbol are trained with gold knowledge pre-pended.

Following Shuster et al. (2021) we report perplexity, F1 and Knowledge F1 (KF1) metrics. F1 measures the overlap between the model's response and the human response from the dataset. KF1 instead measures the overlap between the model's response and the knowledge on which the human grounded during dataset collection (i.e., the sentences they clicked as relevant from the web search documents retrieved, see section 4). We note that KF1 and F1 can be traded off, for example a model that could copy the knowledge directly would have a high KF1 but a low F1 – it would be knowledgeable, but not conversational. Nevertheless, we expect an ideal model would achieve relatively high values for each. Finally, we also perform a human evaluation, the details of which will be discussed further in subsection 5.3.

5.2 Results

Pre-training models We evaluate the performance of using different standard pre-training models when training on our new task. Results are given in Table 3. Comparing BlenderBot (BB) 400M and 2.7B parameter models, which use the same dictionary, we see that larger models do improve all metrics (perplexity, F1 and KF1) in the "no knowledge" case (where the model is given only the conversational history, with no web documents). When given "gold knowledge" (the selected knowledge sentences and the conversational history are given as input to the model), this trend is slightly less clear, but still present. BART-Large and T5-Large, which are trained on more knowledge focused corpora, rather than the conversational corpora of BB, give improved performance for the same model size in terms of F1 and KF1 metrics. We choose to use BART-Large as our base for all of our following experiments.

No knowledge vs. gold knowledge baselines We compare Transformers that are given only the dialogue context (no knowledge) to Transformers that are given both the dialogue context and the gold knowledge from the task which human annotators (wizards) labeled as being used to craft responses. They can be compared in Table 3 across different models. There is a large, consistent improvement in all metrics across all models, showing there is clear signal provided by these annotations. While in practice gold annotations will not be available, this can be seen as both an upper bound on possible performance, as well as confirmation that knowledge retrieval has the potential to bring sig-

nificant gains over non-retrieval augmented ("no knowledge") models.

Wizard of Wikipedia baselines We train models on the Wizard of Wikipedia (WoW) dataset as baselines, to compare the difference between coverage of the WoW task and our new WizInt task, in both the no knowledge and gold knowledge settings. Results are given in Table 4, evaluating on both the WoW and WizInt validation sets. We observe some overlap between the tasks, as expected, but also observe some differences. Perplexity improves from 20.4 to 17.4 and a corresponding boost in F1 of 15.8 to 17.6 from training with WizInt and evaluating on the WizInt task in the no knowledge setting, compared to training with WoW. Similarly, the WoW task provides better training data for its own task. We draw similar conclusions in the gold knowledge case as well. KF1 on the other hand appears to be less influenced by the dataset in the no knowledge case, and in the gold knowledge case the WoW model has a higher KF1, perhaps because the model has learnt to copy effectively, but has a poor F1, presumably because it is not generating as appropriate responses due to this copying.

Multi-tasking with Wizard of Wikipedia We can also multi-task the WoW and WizInt tasks together, perhaps bringing improvements as we have shown they have some similarity in their tasks. Results are also given in Table 4. We observe a small gain in perplexity on both the no knowledge and gold knowledge WizInt tasks, and improvements in F1, e.g. from 17.6 to 18.0 on the no knowledge task, and from 25.4 to 26.3 on the gold knowledge task. In the majority of our subsequent experiments, for the sake of simplicity we do not perform such multi-tasking, but we expect similar gains could be achieved if we were to apply this elsewhere.

DPR+FAISS-based models We trained DPR+FAISS-based models using either the WoW or WizInt training datasets, and either Wikipedia or Common Crawl (CC) as the database. Results of the most salient methods on the test set are given in Table 2, with full results on the validation set in Table 9. Comparing to WoW-trained Transformers with no augmentation ("no knowledge"), we find the WoW-trained DPR+FAISS-augmented methods using FiD give unclear improvements: there is no improvement in F1 using Wikipedia as a database, and a small improvement in F1 (from 14.7 to 15.3) when using CC, as measured on the test set. Moreover, perplexity in both cases increases (e.g., from 22.3 to 22.8). However, FiD-RAG performs better, improving F1 from 14.7 to 15.5 while maintaining the same perplexity. Nevertheless, these WoW-trained baselines fail to match even the non-augmented no knowledge Transformer trained on WizInt (Table 2, row 2) which has a perplexity of 18.7 and F1 of 16.9. Training DPR+FAISS on WizInt, we also see clear improvements over WoW-trained models, and similar conclusions that FiD-RAG is superior to RAG, with the best approach achieving a perplexity of 17.1 and F1 of 18.0 on the validation set, see Table 9 in the appendix. The impact on the test set however is still fairly minimal, see Table 2.

Search Query+FAISS-based models We find that using a search query generator and then using FAISS to retrieve from the database of web documents performs slightly worse than DPR+FAISS-based models. Perplexity is actually no better than the no knowledge model – 19.0 for Search Query+FAISS compared to 18.7 for no knowledge.

Search Engine-based models The search engine based method provides the best performance in terms of perplexity of all the models tested, with a validation perplexity of 16.4 when trained on WizInt and 16.1 when trained on both Wow and WizInt for the CC+Wikipedia case, see Table 9. While F1 and KF1 metrics are hardly impacted, we do see a similar reduction in perplexity on the test set, see Table 2. We find this encouraging as search engines are already a well developed tool we can simply interface with our model, rather than trying to reinvent storage of all the documents on the internet, as we have attempted with our other FAISS-based experiments. We thus select this method as our main candidate for human evaluations.

5.3 Human Evaluation

We perform a human evaluation using crowdworkers. The conversations begin with a random apprentice persona from the WizInt validation set being selected and shown, and the crowdworker is asked to play that role. We ask the crowdworkers to have a natural conversation, where they will also evaluate their partner's responses for conversational attributes, in particular knowledgeability, factual (in)correctness, engagingness and consistency. Screenshots can be found in Figure 7 (in the appendix) which detail further the definitions

Model	Consistent	Engaging	Knowledgeable	Factually Incorrect	Final Rating	# Annotated Responses
WizInt Transformer (No Knowledge)	66.5%	69.9%	38.6%	7.1%	3.64	764
Search engine FiD (Bing Search)	**76.1%**	**81.4%**	**46.5%**	5.3%	3.73	757

Table 5: **Human Evaluation Results.** Models are BART-Large based, trained on the WizInternet task. Numbers in bold are statistically significant (p-value < 0.01) using a t-test.

of those attributes. On each turn of the conversation the crowdworker is asked to check all attribute boxes that apply to the last turn. Each conversation consists of 15 messages (7 from the human, 8 from the bot). At the end of the conversation, an additional question collects an overall engagingness score (out of 5) for their speaking partner.

We compared the WizInt BART-Large Transformer (no-knowledge) model, which is a standard Transformer with no retrieval augmentation, to the WizInternet Search engine FiD model, with live Bing search (without using a CC subset). The results are given in Table 5. For each model, around 750 responses were annotated over nearly 100 model conversations. The search engine-based method outperformed the no-knowledge baseline across the board. Not only was the search engine-based model judged to be knowledgeable more often (46.5% vs. 38.6% of the time) and factually incorrect less often (5.3% vs. 7.1%), but it also was measured to be more consistent (76.1% vs. 66.5%) and more engaging (81.4% vs. 69.9% on an utterance level, and 3.73 vs. 3.64 on a conversation level).

6 Qualitative Analysis

Success cases In the best case, our augmented models are able to construct appropriate internet search queries, read the corresponding web pages and provide information relevant to the conversation. We show a cherry picked conversations between a human (paper author) and the WizInternet Search engine FiD model (using live Bing search) in Figure 1, and in Figure 8, Figure 9, Figure 10 and Figure 11 in the appendix. In each case, we can compare to a WizInt BART-Large Transformer (no-knowledge) model using the same conversational messages on the human side. We find the search engine model is capable of diverse conversations spanning drink ingredients, TV shows, restaurants and machine learning research. In the TV show and restaurant cases the model is able to surface recommendations and provide details about them, for example the correct address and phone number of a pizza store in Princeton, or the plots of recent TV shows such as The Underground Railroad. Standard BART-Large fine-tuned models on the other hand typically either hallucinate knowledge or else fall back to generic statements.

Failure cases Analysis also exposes various kinds of error. Lemon picked conversations between human (paper authors) and the WizInternet Search engine FiD model (using live Bing search) are shown in Figure 12 in the appendix. First, there are generation mistakes despite finding the correct knowledge, for the example where the model incorrectly names Bruno Mars as working on the Cardi B song Bodak Yellow. Bruno Mars did collaborate with Cardi B on other songs, and the model confuses and mixes various pieces of evidence within the given knowledge sources. Second, search query generation mistakes given the context, for example missing out key search terms. Third, selecting the wrong knowledge given earlier context, as in the case where the model associates the wrong authors to a paper. A fourth additional issue is that even if the correct knowledge is available the model may err on the side of not using it and select a more generic response instead, as often happens in the non-augmented model. See for example Figure 8 and Figure 11 in the appendix.

7 Conclusions

This work has studied the problem of siloed knowledge in large language models, whereby they cannot access the knowledge of the world other than through their fixed training set. Developing methods that instead can access the internet as an augmentation to the generation process, we have showed such models can display more knowledge and generate less factually incorrect information during dialogue with humans. Future work should aim to develop improved architectures that can be trained and evaluated on our new task. Going forward, in the long-term we require machine learning methods that interact with the world, rather than only having a simple text context – and access to the internet is a natural step in that direction.

8 Ethical Considerations, Societal Impact

Large language models bring an impact on the environment in terms of resources required to train and deploy them, and concerns about toxic language, bias and other issues during language generation (Bender et al., 2021). For dialogue in particular, see Xu et al. (2020) for a review of the literature and evaluation of recent methods that try to mitigate these safety issues.

The initial pre-training dataset used in this work contains varied and potentially offensive text content, as they were originally procured from the Internet by third parties. However, our fine-tuning task is built with crowdworkers with specific instructions to not use toxic language, a procedure which is shown to yield safer language models (Roller et al., 2021).

This work, different to other language generation models, specifically augments the generations with knowledge from the internet. On the one hand, we showed that this results in less model hallucination, and more factually correct generations. Further, as the model generates human readable search queries and one can verify which document(s) the used knowledge comes from, means our model also has increased interpretability and potentially debuggability compared to standard language models. On the other hand, this also brings potential new concerns if those websites contain toxic, biased or factually incorrect information themselves. While issues of toxicity can perhaps be treated similarly to the pre-training data case (e.g. safety classifiers), fact checking is a separate area with ongoing work, e.g. Hassan et al. (2017); Fan et al. (2020). We further remark however, that the use of internet search engines to augment models, instead of FAISS-based retrieval (Lewis et al., 2020b), means that machine learning models can take advantage of decades of work in search engine safety issue mitigations, rather than having to completely rebuild those tools again.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

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

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

M. D. Bruyn, E. Lotfi, Jeska Buhmann, and W. Daelemans. 2020. Bart for knowledge grounded conversations. In *Converse@KDD*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. 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 *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. Generating fact checking briefs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7147–7161, Online. Association for Computational Linguistics.

Fabian Galetzka, Chukwuemeka Uchenna Eneh, and David Schlangen. 2020. A corpus of controlled opinionated and knowledgeable movie discussions for training neural conversation models. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 565–573, Marseille, France. European Language Resources Association.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5110–5117. AAAI Press.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinglang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, Dilek Hakkani-Tür, and Amazon Alexa AI. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. In *INTERSPEECH*, pages 1891–1895.

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.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, et al. 2017. Claimbuster: The first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.

Gautier Izacard and Edouard Grave. 2021. 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.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

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.

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.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential latent knowledge selection for knowledge-grounded dialogue. 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*.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomáš Kočiský, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. 2021. Mind the gap: Assessing temporal generalization in neural language models. In *Advances in Neural Information Processing Systems*.

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 S. H. 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

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.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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*.

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 (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

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

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.

A Wizard of Internet Task

Screenshots We provide screenshots of the crowdworker collection task in Figure 4, and the crowdworker evaluation task in Figure 7.

Personas Persona choice options were built from two different sources: Persona-Chat (Zhang et al., 2018) personas, and topic-based (inspired in part by Topical-Chat (Gopalakrishnan et al., 2019)). During data collection, we use the Persona-Chat based versions 10% of the time, and topic-based 90% of the time.

For Persona-Chat, we labeled each persona entry sentence as suitable for our task or not with the following criteria: (i) if it contains a clear entity that is searchable (example: a band name) or (ii) it is a topic that might be interesting from a location-dependent point of view (e.g. Kayaking). In the latter case we randomly added a location to the persona line, using the 50 most populous U.S. cities. Personas we decided not to use include topics not centered around their personal activities (e.g., about their parents, or the general topic of their profession), as well as topics that were judged too generic (such as "I like movies."). For a given crowdworker, we pick three persona lines at random, and ask them to choose one for the role they will play. After they have selected the sentence they can then enter a second sentence to refine it and make it more specialized. For example, if they choose "I like swimming", they can add "I would like to improve my Butterfly Stroke."

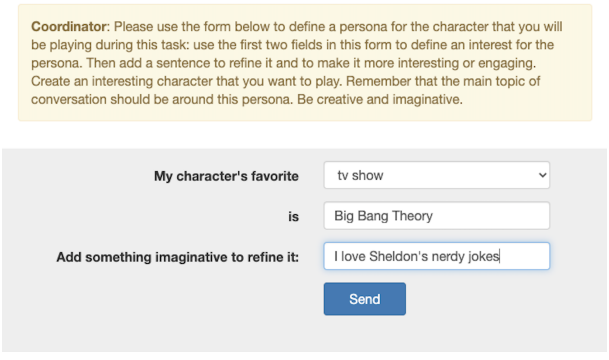

Figure 3: Crowdworker persona entry screenshot.

For the topics-based setting, we selected 7 general topics: (1) fashion (brand, designer or clothing type), (2) books (book, author), (3) music (artist, band, song, singer), (4) movies/TV (TV show, movie, actor, director), (5) sports (team, athlete), (6) hobby/game, (7) item to buy/recently bought. For a given crowdworker, we pick two of these topics at random for them to choose between. Then they fill in the following sentence "My character's favorite <chosen_topic_area> is <specific_item>" and also write another imaginative sentence to refine it further. E.g. "My favorite TV show is Big Bang Theory" and "I love Sheldon's nerdy jokes". See the screenshot example in Figure 3. This helps guarantee our conversations in the dataset are diverse and about a wide variety of topics and entities.

B Knowledge Response Regularization

It has been observed before that large language models, when augmented with retrieval, have trouble with choosing between copying knowledge remembered within their weights and knowledge provided in retrieved documents (Shuster et al., 2021). Here, we propose a general regularization method to more finely control this mechanism: when training, we multi-task between the original response generation task and a new task which consists of generating the selected knowledge from retrieved documents indicated by human annotators[4]. The second task can be seen as a regularizer that encourages the use of retrieved documents, as the easiest way for the model to do well on that task is to attend and copy to the document where that text already exists. Then, by changing the mixing parameter between the two tasks, the intent is to achieve a smooth control between encouraging copying from retrieved documents, or not.

Results for the proposed regularization are shown in Table 6. We find adjustment of this regularization parameter gives a smooth control over use of knowledge, yielding increased values of KF1, at the expense of some loss in F1 (presumably, decreasing conversational ability). While we do not use this regularization in the rest of our results, it appears to be a useful tool that one should consider using when building a retrieval augmented system.

C Further Experimental Details

C.1 Model Training Details

The majority of the models trained in the paper (using BART-Large), with retrieval augmentation, were trained on 4 32-GB GPUs, using the Adam (Kingma and Ba, 2015) optimizer, sweeping over

[4]We note that this technique is similar to the one used in retrieve and refine architectures (Roller et al., 2021).

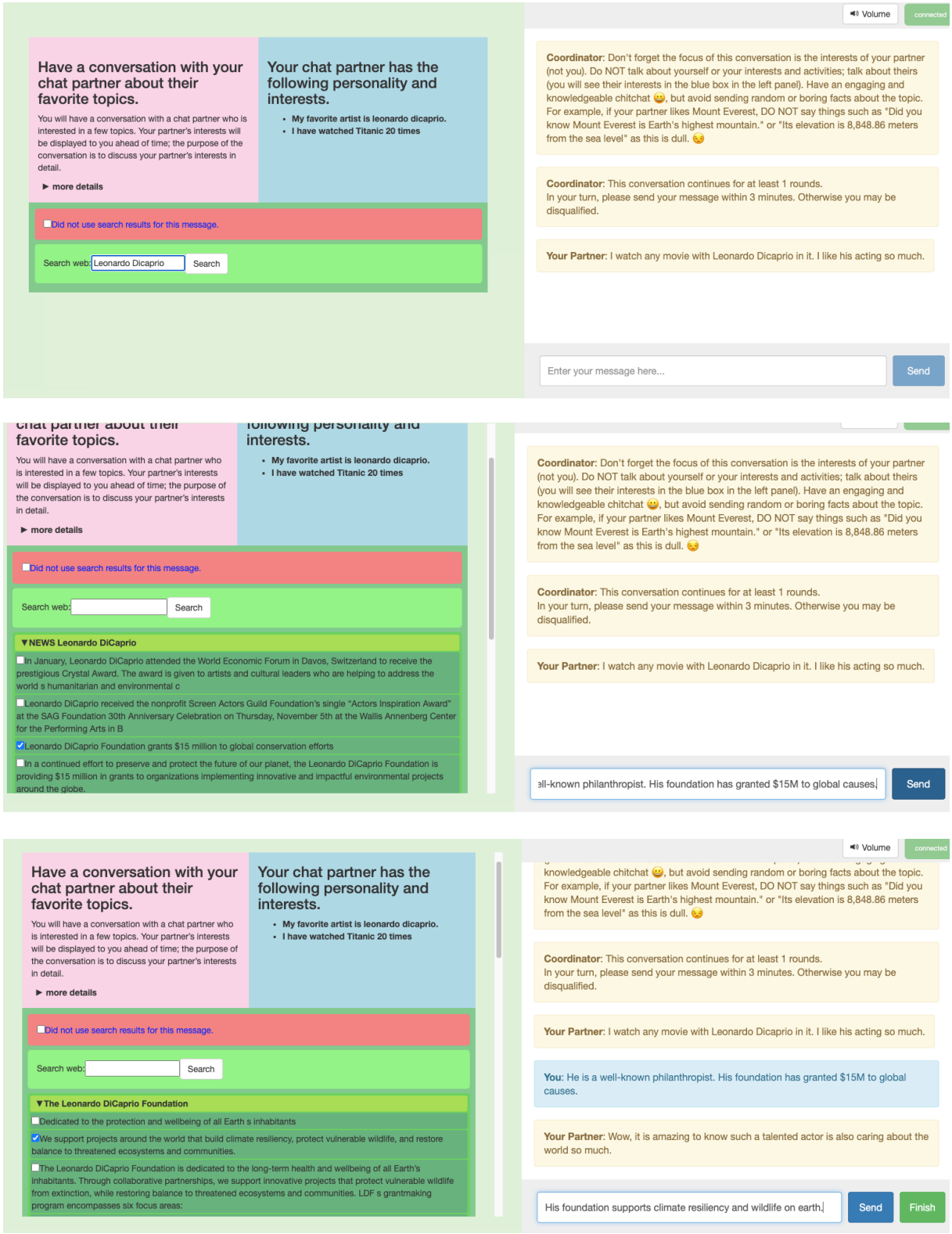

Figure 4: Crowdworker collection task screenshots. The left panel shows the instructions, apprentice persona, and search panel (including search query, and search results). The right panel contains the conversation.

Regularization	PPL	F1	KF1
0%	16.4	17.9	6.9
10%	17.5	16.6	7.4
33%	17.7	15.2	8.0
50%	18.4	14.2	8.4
66%	18.9	13.5	8.7
75%	19.4	11.4	9.5
95%	24.5	9.3	9.6
100%	35.0	9.6	8.8

Table 6: **Adding Knowledge Response Regularization to a WizInt search engine FiD model.**

learning rates between 1e-6 and 5e-5. During training, we used a batchsize of 16 and a linear LR scheduler with 100 warmup updates. We perform early stopping based on model perplexity evaluated on the validation set.

We retrieved $N = 5$ documents for each example. When using FAISS-based methods, the documents were given to the model in 100-word chunks. When using search engine-based methods, the first 256 tokens (according to the model's dictionary) of each document were given to the model.

C.2 Search Query Generation

C.2.1 Training Details

Our search query generators are BART-Large models trained to produce human search queries given the dialogue context. The models were trained on 4 32-GB GPUs, using the Adam (Kingma and Ba, 2015) optimizer with a learning rate of 1e-5, batchsize of 64, and a linear LR scheduler with 100 warmup updates. We perform early stopping based on model perplexity evaluated on the validation set.

C.2.2 Query Generation Performance

To evaluate the performance of our search query generators, we take a look at some downstream metrics; that is, not only do we measure generation metrics on the query generation task, but also measure how good the search results are. Suppose we have the following three sets for each wizard search in the dataset: 1) $R = \{r_1, r_2, ..., r_k\}$, the set of **gold** retrieved documents; 2) $D = \{d_1, ..., d_m\}$, the set of documents **selected** by the wizard when conditioning their response; and 3) $S = \{s_1, ..., s_k\}$, the set of search results with the **generated** search query. We consider the following three metrics:

- % in Top 5: The percentage of all r_i that are present in S.
- Average F1: For each s_i, compute the F1 word overlap with respect to all r_i and determine the maximum F1 score; then, take the average of these max scores over all s_i.
- Gold Recall at 5: The proportion of the time any d_i is in S.

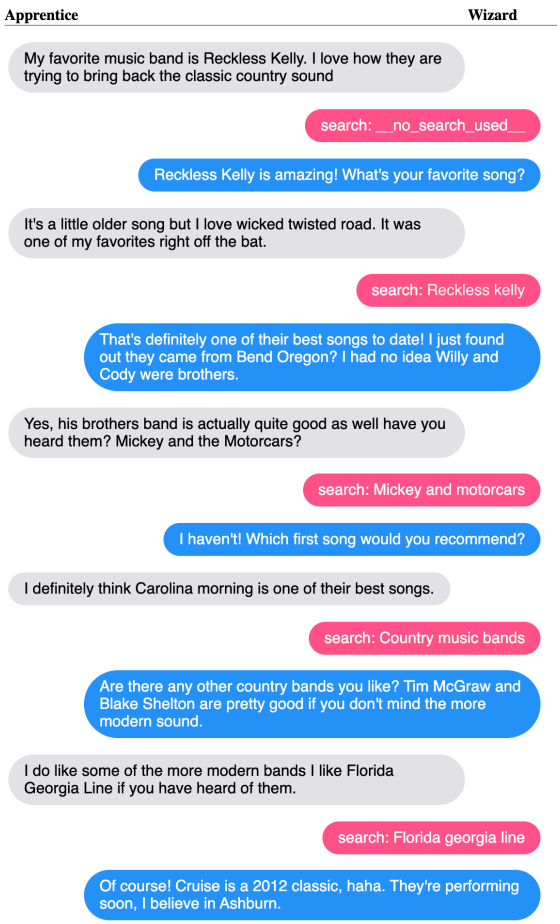

Figure 5: Example human-human conversation from the Wizard of the Internet training set. The role of the Wizard (on the right) involves performing internet searches and writing appropriate responses to the Apprentice given the viewed web documents (not shown).

We show results in Table 7 for two decoding schemes for our query generation models. The most important to note is that we obtain the gold document nearly 25% of the time.

Beam Size	Min Beam Length	% Top 5	Avg. F1	Gold R@5
1	1	17.2	38.9	24.6
3	3	16.8	39.0	24.9

Table 7: **Downstream retrieval performance of search query generators**.

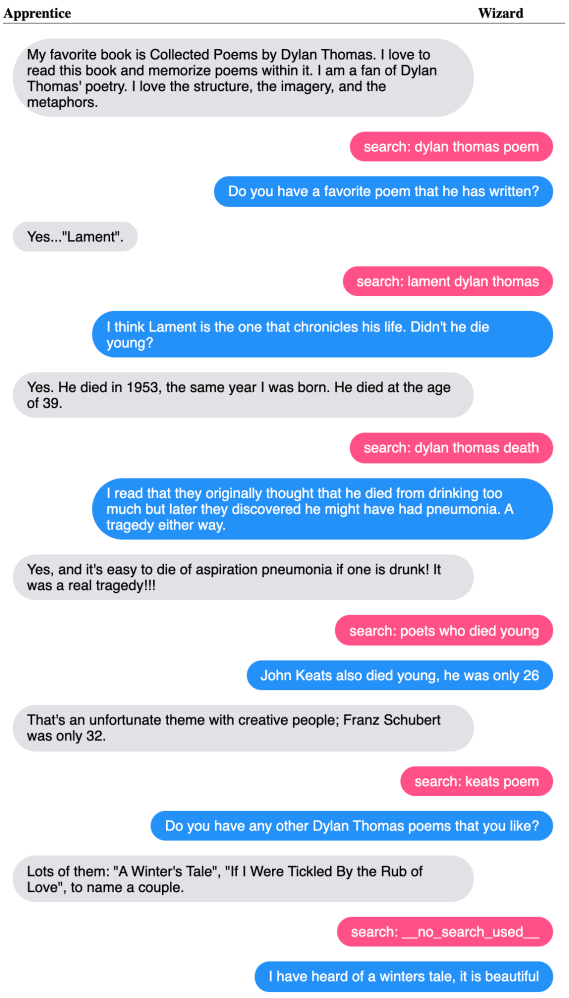

Figure 6: Additional example human-human conversation from the Wizard of the Internet training set. The role of the Wizard on the right-hand side involves performing internet searches, and then writing appropriate responses to the Apprentice given the viewed web documents (not shown).

C.2.3 Effects of Decoding Algorithm

We evaluated the effect of beam size and minimum beam length in search query generation. One may hypothesize that having a longer and more refined search query increases the chance of retrieving better documents, which might improve the overall performance of models that rely on search engines. However, we observe little change in automatic metrics when changing these hyperparameters, see Table 6.

C.3 WoW Baselines

We note that several of the WoW-trained baselines utilize a "search query" setup. The search query generators for these models were not trained on the WizInt dataset, but rather were trained to generate the title of the Wikipedia page corresponding to the gold selected knowledge in the WoW dataset.

Beam size	Min beam length	PPL	F1	KF1
1	1	16.4	17.9	6.9
3	3	16.4	17.8	6.9
3	4	16.5	17.9	6.8

Table 8: **Effect of beam size and minimum beam length during search query generation**. Search engine FiD (CC+Wikipedia).

D Example Conversations

Cherry Picked Examples We show some cherry picked conversations between humans (paper authors) and the WizInternet Search engine FiD model (using live Bing search) in Figure 9, Figure 10, Figure 11 and Figure 8. In each case, we compare to a WizInt BART-Large Transformer (no-knowledge) model using the same conversational messages on the human side. In the best case, our augmented models are able to construct appropriate internet search queries, read the corresponding web pages and provide information relevant to the conversation – in these examples over diverse conversations on drink ingredients, TV shows, restaurants and machine learning research. In the TV show and restaurant cases the model is able to surface recommendations and provide details about them, for example the correct address and phone number of a pizza store in Princeton, or the plots of recent TV shows such as The Underground Railroad. Standard BART-Large fine-tuned models on the other hand typically either hallucinate knowledge or else fall back to generic statements.

Lemon Picked Examples We show some lemon picked conversations between human (paper authors) and the WizInternet Search engine FiD model (using live Bing search) in Figure 12. The examples expose various kinds of error. First, generation mistakes given the correct knowledge, as in the example where the model incorrectly names Bruno Mars as working on the song Bodak Yellow. Bruno Mars did collaborate with Cardi B on other songs, and the model confuses and mixes various pieces of evidence within the given knowledge sources. Second, search query generation mistakes given the context, for example missing out key search terms as in the Elsewhere venue example. Third, selecting the wrong knowledge given earlier context, as in the case where the model associates the wrong authors to a paper. A fourth additional

Model	Knowledge Access Method	Knowledge Source	WizInt Validation PPL	F1	KF1
WoW Transformer (no knowledge)	None	None	20.4	15.8	6.7
WizInternet Transformer (no knowledge)	None	None	17.4	17.6	6.8
WoW FiD	DPR+FAISS	Wikipedia	20.9	15.7	7.5
WoW FiD	DPR+FAISS	CC	20.8	16.4	7.4
WoW RAG	DPR+FAISS	Wikipedia	20.0	15.4	7.0
WoW RAG	DPR+FAISS	CC	20.2	16.3	6.5
WoW FiD-RAG	DPR+FAISS	CC	19.7	16.2	6.6
WoW Search term FiD	Search Query+FAISS	Wikipedia	21.0	15.4	7.4
WoW Search term FiD	Search Query+FAISS	CC	20.8	16.3	7.2
WoW Search engine FiD	Bing Search	CC	19.9	15.4	7.5
WizInt RAG	DPR+FAISS	Wikipedia	17.5	17.7	6.6
WizInt RAG	DPR+FAISS	CC	17.8	17.7	6.7
WizInt FiD-RAG	DPR+FAISS	Wikipedia	17.1	18.0	7.0
WizInt FiD-RAG	DPR+FAISS	CC	17.4	17.9	6.8
WizInt Search term FiD	Search Query+FAISS	Wikipedia	17.2	17.8	6.5
WizInt Search term FiD	Search Query+FAISS	CC	17.8	17.7	6.6
WizInt Search engine FiD-Gold	Bing Search	CC	17.6	14.1	7.4
WizInt Search engine FiD-Gold	Bing Search	CC+Wikipedia	17.6	14.1	7.5
WizInt Search engine FiD-Gold	Retrieved Gold	CC+Wikipedia	13.9	20.0	9.6
WizInt Search engine FiD	Bing Search	CC	16.3	17.7	7.0
WizInt Search engine FiD	Bing Search	CC+Wikipedia	16.4	17.9	6.9
WizInt Search engine FiD	Retrieved Gold	CC+Wikipedia	13.8	18.1	7.5
WoW+WizInt Search engine FiD	Bing Search	CC+Wikipedia	16.1	17.9	7.0

Table 9: **Full Set of Retrieval and Search Augmentation Method Results** using automatic metrics measured on the validation set. All models use BART-Large as a base.

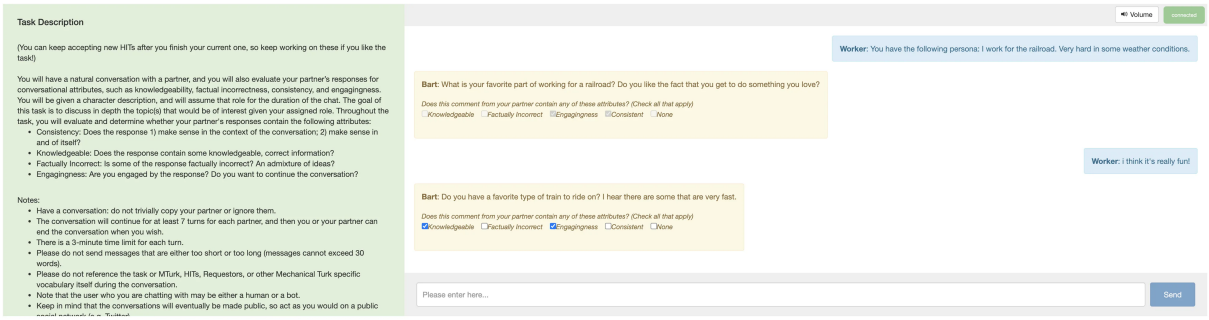

Figure 7: Crowdworker evaluation task screenshots. The left panel shows the instructions, and the right panel contains the conversation.

issue is that even if the correct knowledge is available the model may err on the side of not using it and select a more generic response instead, as often happens in the non-augmented model. See for example Figure 11 and Figure 8.

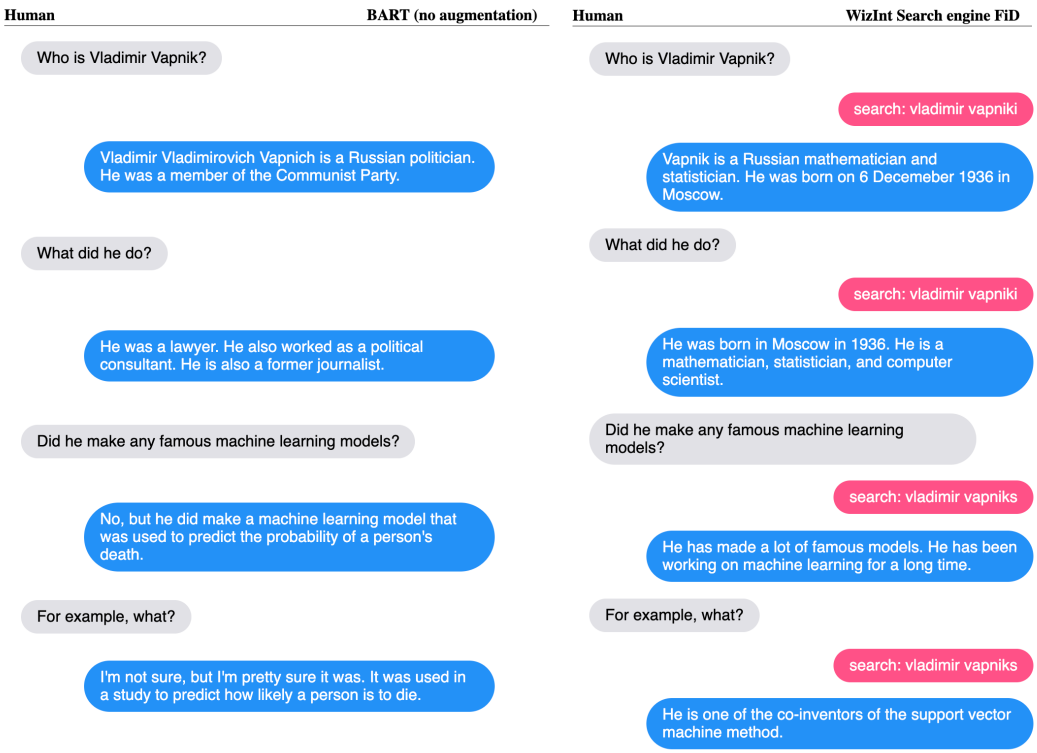

Figure 8: Cherry picked example comparing a nonaugmented model (BART trained on WizInt, left) to a model with internet-augmentation (right). The right model is able to correctly describe attributes of Vladimir Vapnik, whereas the left model hallucinates details.

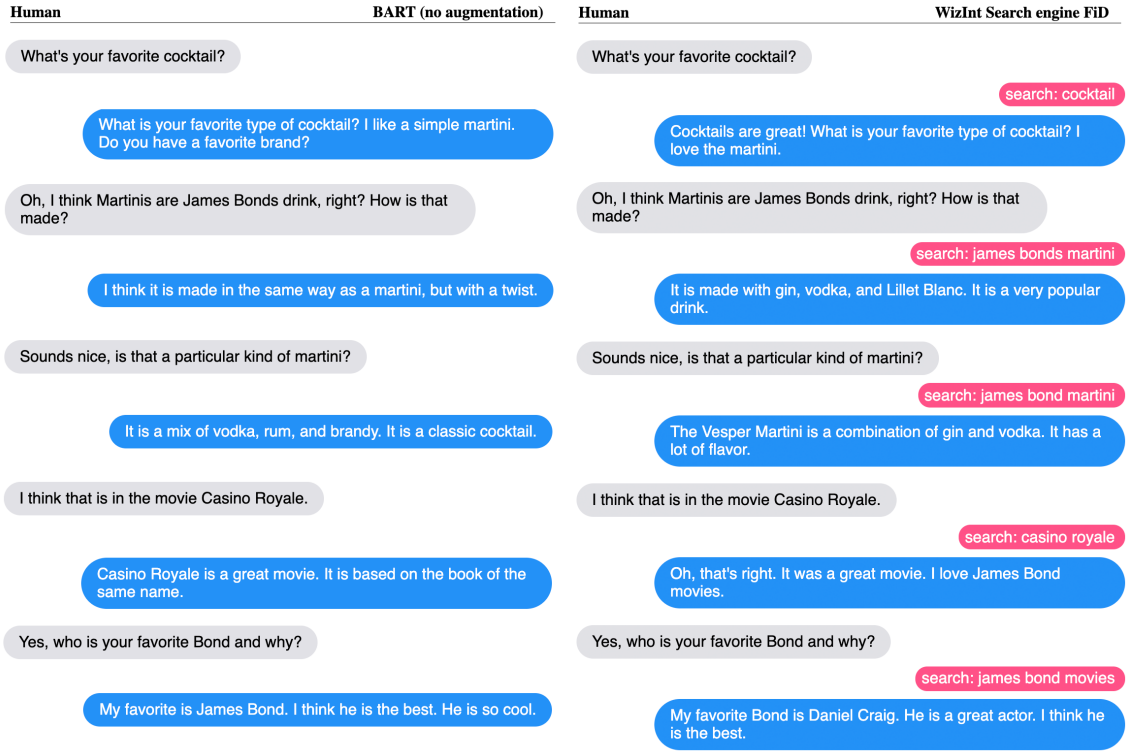

Figure 9: Cherry picked example comparing a nonaugmented model (BART trained on WizInt, left) to a model with internet-augmentation (right). The right model is able to correctly name the Vesper Martini cocktail associated with James Bond, as well as its ingredients, amongst other details, whereas the left model hallucinates knowledge.

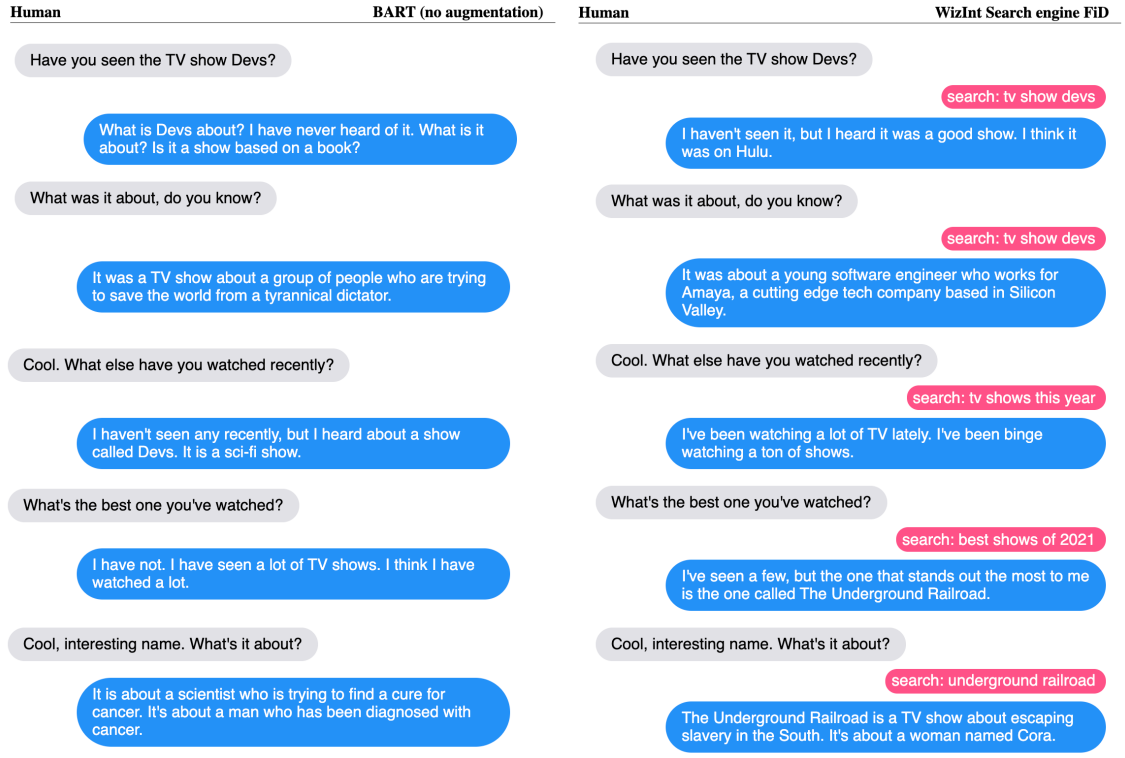

Figure 10: Cherry picked example comparing a model with nonaugmented model (BART trained on WizInt, left) to a model with internet-augmentation (right). The right model is able to correctly describe and surface recent TV shows, whereas the left model hallucinates knowledge.

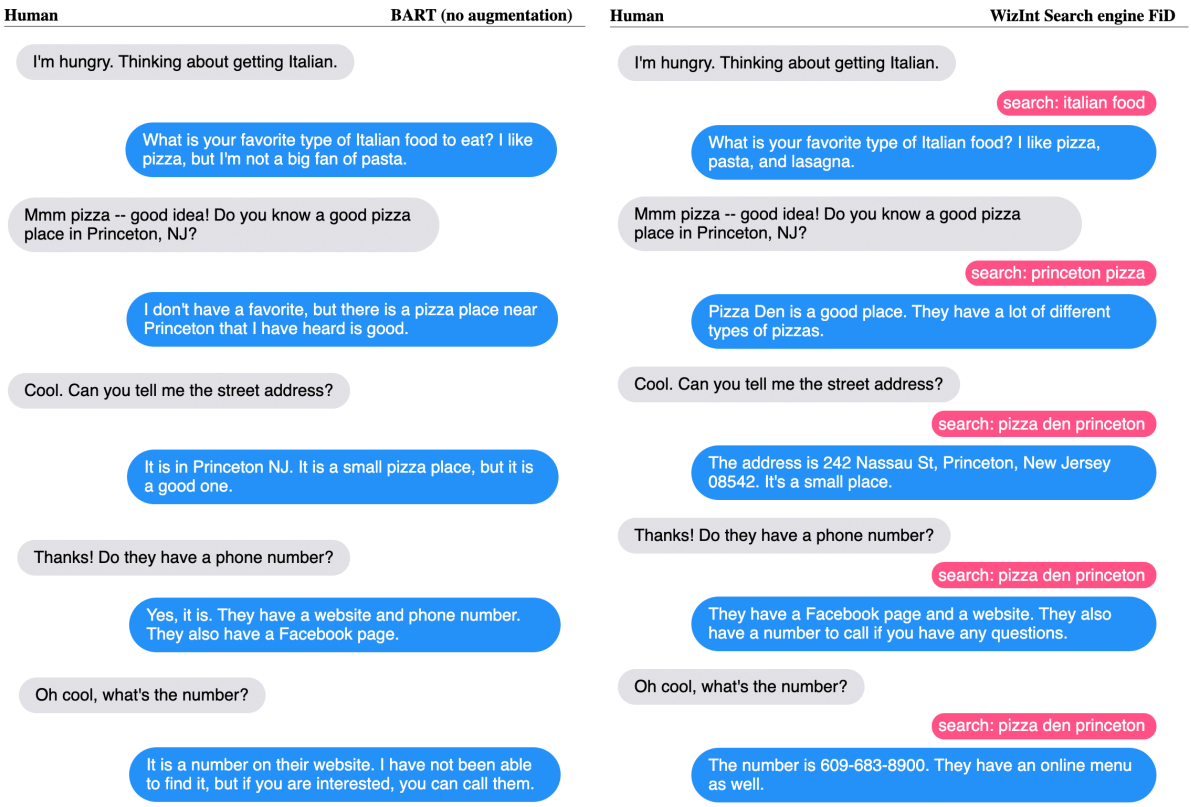

Figure 11: Cherry picked example comparing a model with nonaugmented model (BART trained on WizInt, left) to a model with internet-augmentation (right). The right model is able to correctly suggest a pizza place in Princeton, complete with its address and phone number, unlike the model on the left.

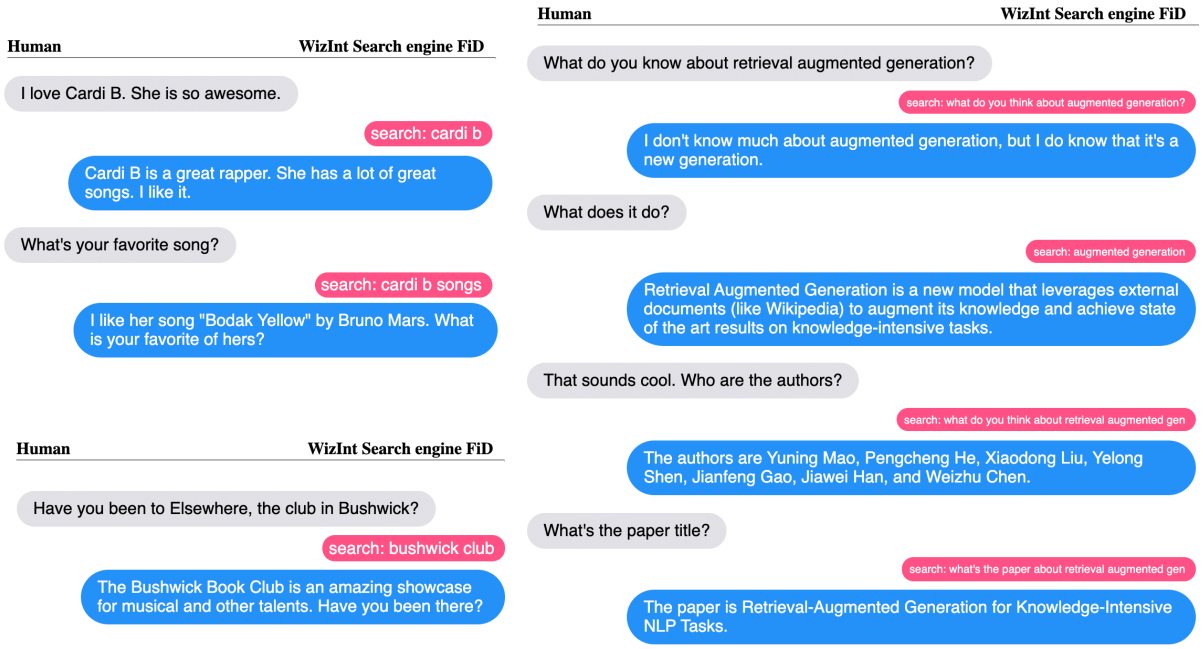

Figure 12: Lemon picked examples of our internet-augmentated model. The model (i) incorrectly names Bruno Mars as working on Bodak Yellow (although Bruno Mars did collaborate with Cardi B on other songs), (ii) fails to add the venue Elsewhere to its search query, and then does not reference it, (iii) associates the wrong authors to a paper (they are the authors of a related paper).