

MemPrompt: Memory-assisted Prompt Editing with User Feedback

Aman Madaan^{*}, Niket Tandon^{*†}, Peter Clark[†], Yiming Yang

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

[†] Allen Institute for Artificial Intelligence, Seattle, WA, USA

{amadaan, yiming}@cs.cmu.edu

{nikett, peterc}@allenai.org

Abstract

Large LMs such as GPT-3 are powerful, but can commit mistakes that are obvious to humans. For example, GPT-3 would mistakenly interpret "What word is similar to *good*?" to mean a homophone, while the user intended a synonym. Our goal is to effectively correct such errors via user interactions with the system but without retraining, which will be prohibitively costly. We pair GPT-3 with a growing memory of recorded cases where the model misunderstood the user's intents, along with user feedback for clarification. Such a memory allows our system to produce enhanced prompts for any new query based on the user feedback for error correction on similar cases in the past. On four tasks (two lexical tasks, two advanced ethical reasoning tasks), we show how a (simulated) user can interactively teach a deployed GPT-3, substantially increasing its accuracy over the queries with different kinds of misunderstandings by the GPT-3. Our approach is a step towards the low-cost utility enhancement for very large pre-trained LMs.¹

1 Introduction

Language models are now better than ever before at generating realistic content, but still lack commonsense (Bender and Koller, 2020; Marcus, 2021). One failure mode due to a lack of commonsense is in misunderstanding a user's *intent*. The typical remedy of retraining with more data is prohibitive due to the cost and infrastructure requirements. In such cases, even if users repeatedly observe the model making a mistake, there are no avenues to provide feedback to the model to make it more accurate and personalized over time.

Our goal is to allow users to correct such errors directly through interaction, and without retraining

Our memory enhanced GPT-3 implementation.

User: What word is similar to *good*?

GPT-3: The homophone of good is: wood.

User: "Similar to" means "with similar meaning".

GPT-3: Noted [*writes to memory*]

User: What word is similar to *surprised*?

GPT-3: The synonym of surprised is: amazed.

[*Retrieves and adds to prompt "Similar to" means "with similar meaning"*].

Figure 1: This paper enhances GPT-3 performance by looking up questions with a similar intent that received any user feedback. Our approach is simple because only the question in the prompt needs to be updated with relevant feedback, and no retraining is necessary.

by injecting the knowledge required to correct the model's misunderstanding. Building upon the recent success of injecting commonsense in the input (Lewis et al., 2020; Talmor et al., 2020), we propose a novel approach of injecting knowledge in the input via interactive feedback from an end-user.

Our approach, MemPrompt, pairs GPT-3 with a growing memory of cases where the model misunderstood user's intent and was provided with corrective feedback. This feedback is question dependent, and thus the prompt for each sample is *edited* to adapt to the input. In this sense, our work can be seen as an instance of prompt engineering (Liu et al., 2021b) which involves editing the prompts. Our work adds interactivity to prompt engineering as it involves dynamically updating the prompt for every instance.

Figure 1 presents a sample interaction between a user and GPT-3 that our setup enables. The model was asked for a similar word. However, the model's (incorrect) task understanding *u* was "The homophone of good is". The user can detect such discrepancy between the intended and interpreted task instruction, and can provide feedback *fb* as "*similar to means with a similar meaning*", clarifying

^{*}Equal Contribution

¹Code, data, and instructions to implement MemPrompt for a new task at <https://www.memprompt.com/>

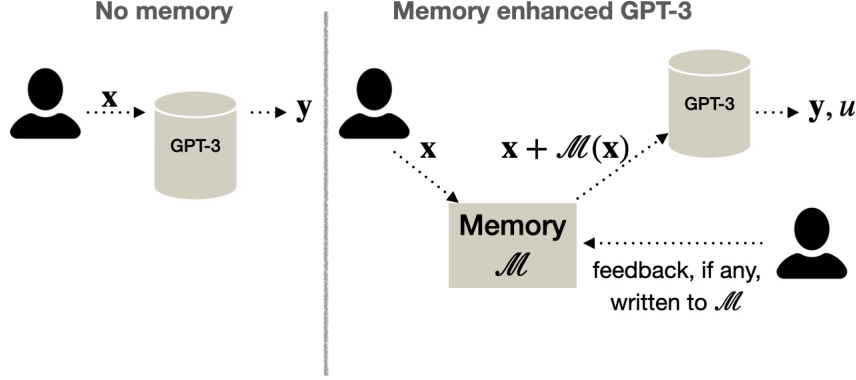


Figure 2: Proposed architecture: (left) GPT-3 does not account for user feedback. (right) MemPrompt maintains a memory \mathcal{M} of corrective feedback, and searches for feedback from prior queries with a similar intent as x using a retrieval function $\mathcal{M}(x)$. x is then concatenated to the retrieved feedback and appended to the prompt for querying GPT-3. Users can also give new feedback on the model’s task understanding u , then added to \mathcal{M} .

that they actually wanted a synonym. Crucially, note that such instructional correction is feasible *even if the user does not know the correct answer to their question*, as they are critiquing the model’s understanding of their intent, rather than the answers themselves. Thus, our setup **does not** require the users to be experts at tasks being solved, another advantage of our approach.

Further, it is desirable to have a system that can leverage past feedback on new, unseen examples for prompt-editing. We maintain a memory \mathcal{M} of such feedback as a set of key-value pairs, where the key is a misunderstood question, and the value is the user’s feedback to correct that misunderstanding. Given a new question, we check if the model has made a mistake on a similar question earlier, by querying the memory for a similar question. If found, append the corresponding feedback to the question prompt. This mechanism aims to prevent the model from making the same type of mistake twice. This failure-driven reminding mechanism draws inspiration from the theory of recursive reminding in psychology (Jacoby and Wahlheim, 2013), which suggests humans index error corrections in the context in which those errors occurred.

This paper presents the general architecture for the system and provides representative implementations for each component. We then demonstrate the system on four tasks, using simulated user feedback: (1) lexical relations (e.g., antonyms, Figure 1), (2) word scrambling (e.g., anagrams), (3) ethical reasoning with user feedback being the appropriate *class* of ethical consideration, e.g., “it is about

cheating”, using a small set of categories, and (4) ethics reasoning with user feedback being natural language. We find that in all cases, GPT-3’s accuracy significantly increases with time, without retraining, as our approach enables it to use corrective feedback from earlier examples to avoid similar misunderstandings on future examples. In summary, our **contributions** are:

- We show that a large model like GPT-3 can be improved after deployment, without retraining, through a memory-assisted architecture.
- Our implementation, MemPrompt, is the first demonstration that this is possible - this is an important step forward for real use of LMs, and the paper sets out a general architecture that others can build on, a specific implementation, and detailed evaluation on multiple tasks.

2 Related work

In Tandon et al. (2022), we show that using a memory of user feedback can be used to repair erroneous model in a supervised setting. In this work, we build upon the recent advances in few-shot prompting to modify GPT-3’s behavior by adding user feedback to the query (prompt). Like others, we use GPT-3 with *few-shot prompting*, where the prompt consists of a **prefix** *prefix* containing a few input-output “training” examples of the task, followed by the **input** x , e.g., a question, to operate on. However, while prior work has focused on constructing better prefixes, e.g., dynamically selecting good “training” examples based on the question (Le Scao and Rush, 2021; Liu et al., 2021a), or even rep-

representing the prefix latently (Li and Liang, 2021), our work elaborates the input x itself to clarify the intended task, by adding user feedback fb from previous misunderstandings.

Similarly, our work can be seen as a form of retrieval-augmented QA. Extensive prior work has used retrievals from a text corpus to aid QA, e.g., Pan et al. (2019); Guu et al. (2020), or retrievals of prior QA pairs for nearest-neighbor QA (Khandelwal et al., 2020). In contrast, we retrieve from a dynamic memory of user feedback.

The idea of failure-driven reminding and dynamic memory date back several decades, e.g., (Schank, 1983; Riesbeck, 1981). Our work resurrects these ideas in a modern context.

Learning from instruction has become important for large LMs that can perform a task based on direct instruction rather than examples (Wei et al., 2021; Mishra et al., 2021). Our work extends this by adding an adaptive component when those instructions are misinterpreted. While it may not be possible for a user to provide meaningful feedback on the output itself, giving feedback on the understanding of the instruction is more feasible.

Our approach aims to modify the model’s behavior through prompting, given a wrong answer. An alternative, recently explored approach is “model editing” - updating the model itself by modifying its parameters to fix incorrect answers (Mitchell et al., 2021; De Cao et al., 2021; Hase et al., 2021). Model editing approaches have to date been limited due to uncontrollable out-of-scope changes (Mitchell et al., 2021). In contrast, our goal is not just to correct a prediction, but to generalize that correction for new problems by collecting feedback to clarify the misunderstanding without damaging the model’s basic problem-solving acumen.

Finally, our work is a simple example of debugging and learning via dialog. While system debugging through dialogue has been explored in many contexts (Hixon et al., 2015; Wang et al., 2016; Davis, 1977), our contribution is a dialogue about the model’s understanding of the user’s intent.

3 Approach

3.1 Memory enhanced GPT-3 architecture

In our setup, given an input x , a model generates an output y and a sentence u expressing its understanding of the task, a skill learned through few-shot examples in the prompt (Appendix D). The user can then critique u by providing natural lan-

guage feedback fb . This is feasible even if the user does not know the correctness of y because they are critiquing the *model’s understanding of their intent* rather the answers themselves.

Given a new query, MemPrompt uses fb from similar, prior queries to enrich the (few-shot) prompt p . We use the principle that if two inputs x_i and x_j are similar (i.e., $x_i \sim x_j$), then their feedback fb_i and fb_j should be exchangeable ($x_i \sim x_j \Leftrightarrow fb_i \sim fb_j$). The underlying assumption here is that for a fixed model, similar inputs will incur similar errors, and thus can use the same feedback for correction. Fig. 2 gives an overview of MemPrompt, with the following components:

Memory \mathcal{M} : \mathcal{M} is a growing table of key (x_i) - value (fb_i) pairs that supports read, write, and lookup operations. The write operation is used whenever a user gives new feedback.

Lookup $\mathcal{M}(x)$: The memory allows lookup operations, denoted as $\mathcal{M}(x)$, that matches the query= x against all the keys of \mathcal{M} .

Combiner $\mathcal{C}(x, \mathcal{M}(x))$: A gating function allowing irrelevant, retrieved feedback to be ignored.

Few-shot prompting Let us briefly recap few-shot prompting with GPT-3. Consider a general setup where given an input x , a model is expected to generate an output y . In a few-shot prompting mode (Brown et al., 2020), a prompt p consists of k (x, y) “in-context” examples, i.e., $p = x_1.y_1\#x_2.y_2\ldots\#x_k.y_k$, where $\#$ is a token separating examples and $.$ indicates concatenation. During inference, the user inputs a question x_i , and the model is fed $p \# x_i$ (i.e., the question suffixed to the prompt) and is expected to generate the answer y_i as a continuation.

MemPrompt setup As mentioned, given an input x , we prompt the model to generate an output y and a sentence u expressing its understanding of the task. Thus, the in-context examples for MemPrompt are of the form $x \rightarrow u, y$. In addition to the input x , MemPrompt retrieves a fb if a question similar to x has been asked before. To enable the model to react to such feedback, we also include examples of the form $(x, fb \rightarrow u, y)$ in the prompt, which are aimed to teach the model to react to fb (Appendix D).

Task (fb type)	(x \rightarrow y)	u and fb
Lexical relations (INS)	x : What sounds like good? y : wood	u : Question is asking for a synonym. fb : No, I want a homophone.
Word scrambling (INS)	x : Find the right word given this cycled word: elylarg y : largely	u : The question is about anagram. fb : No, its about uncycling a word.
Ethical reasoning (CAT)	x : Turning my blender on at 3AM y : It's bad.	u : Question is about authority. fb : No, it is about harm.
Ethical reasoning (NL)	x : John has started using again after his mother passed y : It's bad.	u : Question is about spending money. fb : No, it is about drug use.

Table 1: Feedback types and demonstration of understanding: our system leverages user feedback to prevent failures caused due to a misunderstanding of the task (INS) or semantics of the input (CAT and NL). We achieve this by having the model articulate an understanding **u**, on which a user can provide feedback using **fb**.

3.2 Verbalizing Task Understanding

Existing methods for receiving user feedback typically assume the user knows the correct answer **y** (Elgohary et al., 2021). This assumption is paradoxical: if the user knew the answer, why would they be using the model? Further, allowing only “oracle” users (who know correct **y**) might lead to sampling biases. In real-world settings, it is common for users to not have the exact answer, but rather, a general understanding of what they are searching for. Thus, we propose eliciting a verbalization of task understanding **u** from the model in addition to the answer. End users can thus critique **u**.

We operationalize this idea by including task verbalization in the prompt (Fig. 3). Given a question *What sounds like < sighted > ?*, a vanilla prompting approach will generate the answer *cited*. In contrast, we include a **u** *the homophone for* in the prompt. Large-scale language models, such as GPT-3, have been shown to excel at reasoning with a limited number of examples, making them well-suited to mimic the prompt and generate not only the answer, but also an understanding of the task at hand. Given a test question *What sounds similar to < sighted > ?*, if the model generates *the word that has the same meaning* as **u**, the user has a reason to believe that the answer is wrong. Our experiments demonstrate that GPT-3 models are able to generate this additional information in all tasks presented.

Our approach is not foolproof—the model may spell out a wrong **u** while giving out the correct answer, misleading the user into believing that there is an error (or vice-versa). Hallucinating remains a critical limitation of generative models (Cao et al., 2022), therefore additional heuristics and model calibration might be necessary to make our approach foolproof. In practice, however, we found such cases to be rare for the tasks in this paper.

(Word reasoning task)

Ques: What sounds like < sighted > ?

Ans: *the homophone for* sighted is cited.

(Ethical judgment task)

Situation: i heard that if you cringe about your past it just means you’ve grown as a person, for anyone who needs to hear it.

Morality Judgment: *This question is about: regretting poor decisions from your past.* The answer is it’s okay.

Figure 3: MemPrompt is tuned to generate *task understanding* + answer. This allows the users to provide feedback on the task understanding even without knowing the actual answer.

3.3 Allowing GPT-3 to react to feedback

Once the feedback is received from the user, can the model successfully utilize it? By adding a few examples of the form **x**, **fb** \rightarrow **u**, **y** in the prompt and setting **fb** = **u**, we force the model to use the task understanding present in the input when generating the output (Figure 4). Recently, it has been shown that such repetition plays a crucial role in the success of few-shot prompting models (Madaan and Yazdanbakhsh, 2022).

Ques: What is similar to popular ? clarification:

when I ask for similar to, I want a synonym.

Ans: *the synonym of* popular is admired.

Figure 4: An in-context example of the form **x**, **fb** \rightarrow **u**, **y**, which encourages **u** to be like **fb**, thereby conditioning the output to react to **fb**.

3.4 Feedback on model’s understanding

Within the setup **x** \rightarrow **u**, **y**, we focus on following two modes of failure:

- Task instruction understanding: this is especially concerning in a multi-tasking setup, where the model may consider the question to be about a different task than the one user intended.
- Task nuanced understanding: when the model understands the task type, but misunderstands the subtle intent in a question.

Our primary goal is to elicit feedback on the model’s understanding of the task, however, we also explore settings where an Oracle is available to provide feedback on the labels (as detailed in Section §4.3). Finally, we note again that the model reacts to the feedback because some in-context samples are of the form: $(x, fb \rightarrow u, y)$. We consider a diverse set of tasks $(x \rightarrow y)$, fb and u , as summarized in Table 1.

3.5 Tasks

We apply our approach to four tasks: (1) lexical relations (e.g., antonyms, Figure 1), (2) word scrambling (e.g., anagrams), (3) ethics (with user feedback being the appropriate *class* of ethical consideration), and (4) ethics (with user feedback being natural language). For all five tasks, the dataset consists of $(x, fb \rightarrow u, y)$ tuples, where fb clarifies the task in x . We have a simulated conversational setting, in which a user can ask the model x (covering any of these five tasks). If the model gives a wrong answer to query x , then fb is used as the simulated corrective feedback. The sources for these datasets are listed in Appendix §E.

3.5.1 Lexical Relations

The lexical relation task is to predict a word with a given lexical relationship to an input word. We use five relationships: synonym (*syn*), antonym (*ant*), homophone (*hom*), definition (*defn*), and sentence usage generation (*sent*).

3.5.2 Word Scrambling

For this task, given a word with its characters transformed, the model is expected to recover the original characters. There are four transformation operations the user can request: reversal of words (*rev*, *yppup* \rightarrow *puppy*), cycle letters in word (*cyc*, *atc* \rightarrow *cat*), random insertions (*rand*, *c!r ic/ke!t* \rightarrow *cricket*), and anagrams by changing all but the first and last (*anag1*, *eelhpnat* \rightarrow *elephant*) or all but the first and last 2 characters (*anag2*, *elapehnt* \rightarrow *elephant*). We use the original dataset by Brown et al. (2020).²

²word scrambling dataset <https://github.com/openai/gpt-3/tree/master/data>

For both these tasks, each question can be asked in multiple ways (e.g., for synonym generation, the users might ask questions of the form *what is like*, *what has a similar sense*, *what is akin to*, *what is something like*, etc.) Similarly for the lexical relations task, we specify the task description x using different phrasings, e.g., “rearrange the letters” (which the system sometimes misunderstands), and the (simulated) user feedback fb is a clearer task description, e.g., “The anagram is”. The system thus accumulates a set of (x, fb) pairs in memory after each failure, helping it avoid future misunderstandings of x through feedback retrieval.

3.5.3 Ethical Reasoning (2 tasks)

For ethical reasoning, we consider a setup where given a situation (e.g., *cheating on your partner*), the model is expected to provide a judgment on whether the situation is ethical or not (e.g., *it’s not okay*). In addition to providing a judgment on the ethics of the situation, the model also elucidates its understanding of what the question is about (e.g., *being loyal*). While the user may not know the answer, we posit that they would be able to provide feedback on the broader context. For example, if the model generates *being financially savvy* instead of *being loyal* for the situation *cheating on your partner*, a user can still point out this problem and provide feedback.

We use a subset³ of the dataset provided by DELPHI (Jiang et al., 2021). We simulate two different kinds of user feedback, using two of the annotations attached to each example in the Delphi dataset:

- Categorical feedback (ERT-CAT): In this setting, the model generates its understanding u of the situation by selecting one of 10 different possible categories of morality to which the situation might belong: *care*, *loyalty*, *authority*, *fairness*, *sanctity*, *degradation*, *cheating*, *subversion*, *betrayal*, and *harm*. These categories are explicitly provided for each example in the Delphi dataset.
- Natural language feedback (ERT-NL): For this, we use the associated “rule of thumb” (RoT) annotation—a general moral principle—attached to each example in the Delphi dataset. To compile a challenging subset of the data for ERT-NL, we sample by input length, preferring long x , with a short feedback fb . Specifically, we use the top 1% of the inputs by length to create a challenging

³social norms dataset (social-chemistry-101, Forbes et al. (2020)) <https://github.com/mbforbes/social-chemistry-101>

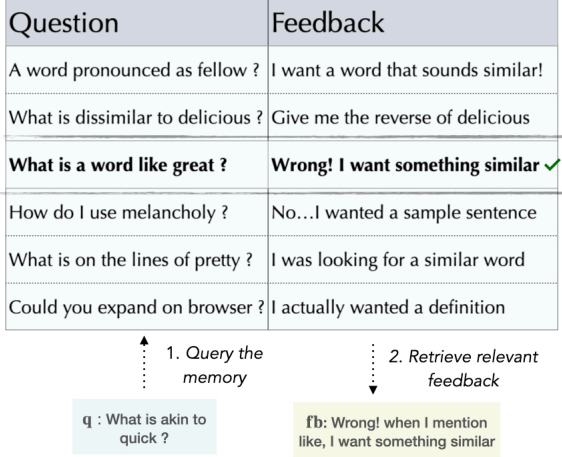


Figure 5: Sample snapshot of memory for lexical QA.

set of input situations (\mathbf{x}). User feedback \mathbf{fb} is a natural language feedback on the understanding \mathbf{u} .

In both the cases, the model is “taught” to generate a category \mathbf{u} (as well as the okay/not-okay answer \mathbf{y} to the ethical question) by being given a few examples in the prompt prefix, thus articulating which moral category (for ERT-CAT) or rule-of-thumb (for ERT-NL) it thinks is applicable. The simulated feedback \mathbf{fb} is the gold category associated with the example in the question, if GPT-3 gets the answer wrong.

We selected these tasks because situations that involve reasoning about similar ethical principles can utilize similar past feedback. For example, *sharing an extra umbrella with your friend if they don’t have one*, and *donating surplus food to the homeless* both involve *compassion*.

3.6 MemPrompt Implementation

Implementation of memory \mathcal{M} \mathcal{M} uses the user input \mathbf{x} as the key and the corresponding feedback \mathbf{fb} as value. Given a question \mathbf{x}_i , if the user detects that the model has misunderstood the question, they may provide a \mathbf{fb}_i with *clarification probability* $Pr(\mathbf{fb}_i)$. The $(\mathbf{x}_i, \mathbf{fb}_i)$ pair is stored in a memory \mathcal{M} , with \mathbf{x}_i as the key and \mathbf{fb}_i as the value. For a subsequent question \mathbf{x}_j , the retriever $\mathcal{M}(\mathbf{x})$ checks if a similar question appears in memory. If yes, then the corresponding feedback is attached with the question and fed to the model for generation.

For example, a question asking for a synonym, such as *what is akin to fast?* might be misinterpreted as a request for antonyms. As mentioned, in our setup, the model generates its understanding of

the task \mathbf{u} , and not just the answer to the question. The user, by inspecting $\mathbf{u} = \textit{The opposite of fast is:}$ might determine that the model has misunderstood them, and give feedback *i wanted a synonym*, which gets stored in \mathcal{M} . If a similar question (e.g., *what is akin to pretty ?*) is asked later by the same or a different user, the corresponding feedback (*i wanted a synonym*) is attached with the question to generate the answer. Figure 5 illustrates a sample memory for this task.

Implementation of retriever $\mathcal{M}(\mathbf{x})$ A retrieved past feedback that is incorrect might cause the model to make a mistake, thus necessitating a good retrieval function. We propose a two-stage method for effective retrieval involving: transforming \mathbf{x} , followed by a similarity lookup of the transformed \mathbf{x} in \mathcal{M} . When the task involves high surface-level similarity among past feedback, such as in lexical word tasks, then a simple heuristic-based transformation is sufficient. However, such simple transformations are insufficient for tasks that involves more complex retrieval e.g., when two lexically dissimilar situations can share the same understanding. For example, consider two situations from ERT-NL: *Filling a false time sheet at work* and *Being at a party, and telling parents I am studying*. These situations look lexically dissimilar but correspond to the same underlying social principle *lying to authority*. In our experiments, off-the-shelf methods failed to address these challenges (see §4 later).

To address these challenges with transformation in complex tasks, we have designed a novel SEQ2SEQ based transformation called GUD-IR. Given \mathbf{x} , GUD-IR generates a *transformed* feedback $\hat{\mathbf{fb}}$ for \mathbf{x} using a *generative* SEQ2SEQ model. Our approach is inspired and supported by the recent success of generate and retrieve (Mao et al., 2021) methods. However, despite the similarity, the methods have different goals: Mao et al. (2021) leverage generative models for query expansion, whereas our goal is explainable input understanding. See Appendix B for more details on GUD-IR.

After the transformation stage, the closest matching entry is then used as the corresponding \mathbf{fb} . Transformation reduces $\mathcal{M}(\mathbf{x})$ to a search over $\mathbf{fb}_1, \mathbf{fb}_2, \dots, \mathbf{fb}_{|\mathcal{M}|}$ with $\hat{\mathbf{fb}}$ as the search query. We compute similarity based on a fine-tuned Sentence transformers (Reimers and Gurevych, 2019).

Implementation of combiner \mathcal{C} \mathcal{C} concatenates \mathbf{x} with relevant \mathbf{fb} retrieved by $\mathcal{M}(\mathbf{x})$. To ensure

that the \mathbf{x} is appended with \mathbf{fb} only if it is relevant, our current implementation of combiner uses a threshold on the similarity score between the \mathbf{x} and the closest feedback \mathbf{fb} retrieved by $\mathcal{M}(\mathbf{x})$. We rely on the model (GPT-3) to pay attention to the relevant parts of the input. Exploring more complex gating mechanisms remains an important future work.

4 Experiments

Baselines We compare MemPrompt (memory-assisted prompt editing) with two baselines:

- **NO-MEM** This is the standard GPT-3⁴ in few-shot prompting mode (hyper-parameters listed in Appendix §C). Input is $\mathbf{p} \# \mathbf{x}_i$ (i.e., question \mathbf{x}_i appended to prompt \mathbf{p}). It generates answer \mathbf{y}_i and its understanding of the user’s intent \mathbf{u}_i .
- **GROW-PROMPT**: Similar to NO-MEM, but the \mathbf{p} is continuously grown with a subset of memory \mathcal{M} that can fit within the prompt (max. 2048 tokens). The most recent subset of \mathcal{M} of memory inserted is inserted in the prompt. The ethical reasoning tasks (ERT) involve long examples, and the initial prompt itself takes close to the max allowed tokens. Thus, the GROW-PROMPT setup is only provided for the lexical relations and word scrambling tasks.

Metrics We use two different metrics:

- $Acc(\mathbf{y})$: % of cases where answer matched the ground truth.
- $Acc(\mathbf{u})$: % of cases where the model’s understanding of user’s intent is correct. $Acc(\mathbf{u})$ is also referred to as instruction accuracy. As discussed in §3.4, depending on the task, the model generates its understanding on either the instruction or semantics of the question.

Clarification probability In real-world cases, we cannot expect a user to provide feedback for all the examples (e.g., the user might not know that the understanding of the model is wrong). To simulate this realistic setting, we experiment with various values of clarification probabilities Pr .

4.1 MemPrompt improves GPT-3 accuracy

Does pairing GPT-3 with MemPrompt help? §4.1.1 empirically validates this on ethical reasoning tasks and §4.1.2 on word reasoning tasks.

⁴We use GPT-3-175B (davinci) for all experiments.

4.1.1 Ethical reasoning tasks

Table 2 presents results on the DELPHI dataset (1,000 points in the test set). Recall from §3.5 that there are two kinds of feedback on DELPHI questions: CAT and NL feedback. MemPrompt gets over 25% relative improvement for both ERT-NL and ERT-CAT. We found that having an efficient retriever was critical for ERT-NL: sentence transformer based retriever scored 38.5, vs. 45.2 using GUD-IR, a 17% improvement.

model	ERT-CAT	ERT-NL
NO-MEM	48.3	34.4
MemPrompt	60.0	45.2

Table 2: MemPrompt outperforms NO-MEM for both the categorical and the more challenging ERT-NL setup having longer, ambiguous inputs.

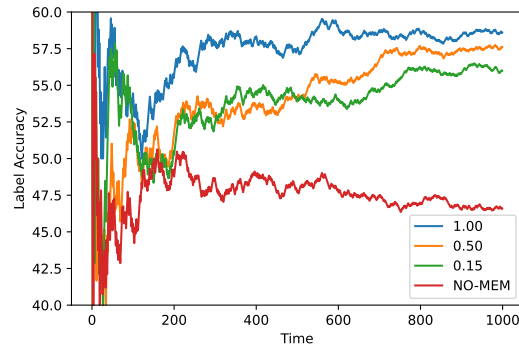


Figure 6: ERT-CAT: Label accuracy increases with time for all values of clarification probabilities $Pr(\mathbf{fb}_i)$.

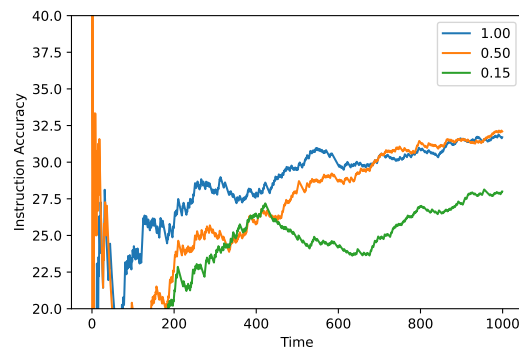


Figure 7: ERT-CAT: Instruction accuracy sharply increases with a larger clarification probability, showing that MemPrompt responds to feedback. With time, lower values of $Pr(\mathbf{fb}_i)$ catch up as memory is gradually filled with error cases and feedback.

Error category	Input	Output (MemPrompt)	Reference
Correct, lexically variant understanding	when I said "i do" to my wife, I am now realizing I only meant "i'll try". does she deserve to know and a chance to back out?	This question is about: telling a spouse your true feelings.	This question is about: loving your partner.
Distracted understanding	i am 20 years old, lately I have been forgetting simple spellings while taking notes in class . I often bump into random objects around hurting myself.	This question is about: wanting to kill yourself. The answer is BAD.	This question is about: seeking medical attention if you feel there is something wrong with you. The answer is GOOD.
Retrieval failures	checked out my high school ex girlfriends instagram for the first time in two years after graduating. I'm happy I did. she's not as pretty as she used to be and I feel great.	[retrieved feedback] This question is about: looking at someone's social media. The answer is GOOD.	This question is about: judging people based on physical attributes. The answer is BAD.

Table 3: ERT NL task error categories

MemPrompt effectively incorporates feedback, improving accuracy over time Figure 7 demonstrates that the instruction accuracy increases over time for different values of clarification probability.

Fig. 6 shows that label accuracy improves over time. Baseline (NO-MEM) saturates after 200 time steps; MemPrompt continues to improve. Continuous improvement is one of our key advantages. These charts show that instruction accuracy and label accuracy are correlated (corr. coeff = 0.36).

We observe that using a higher clarification probability leads to a sharp increase in instruction and label accuracy early on in the training for both ERT-CAT and ERT-NL. This is because a higher clarification probability causes the feedback memory to fill up more quickly, providing more feedback for new questions.

Error analysis: Ethical-NL In ERT NL and CAT tasks, a primary source of label errors is confusion between labels such as OKAY and GOOD due to the nuanced differences e.g., input = teaching your child a musical instrument. MemPrompt predicts GOOD, but the expected answer is OKAY. Jiang et al. (2021) make similar observations.

We randomly sampled examples from the ERT-NL dev set where the model generates an incorrect understanding (i.e., $Acc(\mathbf{u}) = 0$ based on exact match). Our goal is to understand the typical errors made by the model and use the analysis to calibrate the findings in Table 2. We select ERT-NL for the analysis because it involves free-form natural language which is difficult to study quantitatively.

• **Correct, lexically variant understanding (30%):** Exact match underestimates model performance (as the task involves generation). $\sim 30\%$ \mathbf{u} is a lexical variation of the reference gold understanding. E.g., *telling a spouse your true*

feeling vs. *loving your partner*. The generated label in these 30% cases is still correct. (Table 3, row 1)

• **Distracted understanding (50%):** A major source of instruction and label errors is the model getting distracted by an unimportant context. Bad retrieval accounts for 30% errors within this category, e.g., matching a situation in the memory where the expected understanding is only partially applicable to the query. (Table 3, row 2)

• **Retrieval failures (18%):** These errors are caused by an irrelevant retrieved understanding from the memory, when using a state-of-the-art retrieval method (Table 3, row 3). GUD-IR helps to reduce these retrieval failures. See Appendix §B. Table 3 presents canonical examples of these error categories. We also find that over time, more relevant past examples are fetched (see Table 7).

4.1.2 Word Reasoning Tasks

For these tasks, we compare gold \mathbf{u}^* and generated \mathbf{u} based on hard-coded linguistic variations (e.g., *the antonym is matches the opposite is*). While we do not explicitly evaluate task accuracy, we observe a near-perfect correlation between the accuracy of \mathbf{y} and \mathbf{u} (i.e., if the GPT-3 understands the task correctly, the output was almost always correct). This shows improving model’s understanding of a task might lead to an improved performance.

Figure 8 reports the overall performance on the word reasoning tasks. The accuracy improves substantially within 300 examples when using memory (in yellow) vs. no memory (in blue). Note that our approach operates in a few-shot learning regime, where there is no pre-existing training data available. The only examples provided to the model are through the prompt. The performance

of GROW-PROMPT (red) lies in between, showing that non-selective memory is partially helpful, although not as effective as failure-driven retrieval (our model). However, GROW-PROMPT is $\sim 3\times$ more expensive (larger prompts) and cannot scale beyond the 2048 tokens limit. We also found that the retrieved feedback from memory was effective 97% of the time; only in $\approx 3\%$ of cases feedback had no positive effect.

When the memory is used for every example (green line, Fig 8, top), the performance improves quickly vs. the yellow line ($Pr(\mathbf{fb}_i) = 0.5$).

model	syn	ant	hom	sent	defn	all
NO-MEM	0.58	0.43	0.13	0.30	0.39	0.37
GROW-PROMPT	0.71	0.87	0.75	0.92	0.76	0.80
MemPrompt	0.99	0.98	0.98	0.98	0.96	0.98

Table 4: Results on lexical qa: MemPrompt has the best performance across all lexical QA tasks.

model	anag1	anag2	cyc	rand	rev	all
NO-MEM	0.81	0.47	0.95	0.98	0.62	0.77
GROW-PROMPT	0.86	0.89	0.93	0.96	0.90	0.91
MemPrompt	0.81	0.83	0.98	0.95	0.93	0.90

Table 5: GROW-PROMPT and MemPrompt outperform NO-MEM on all word scramble QA tasks.

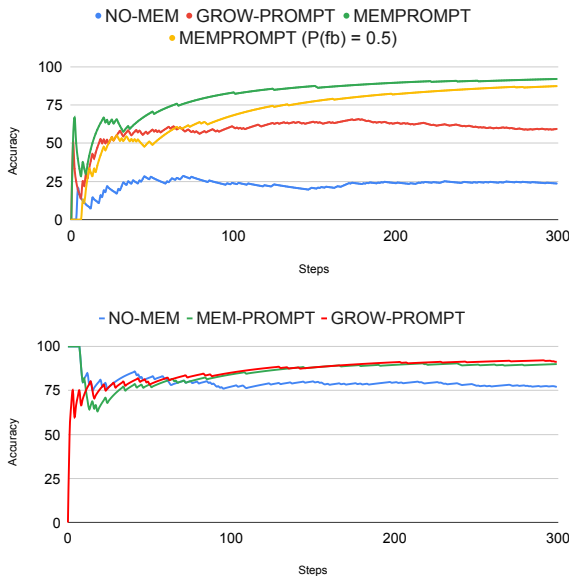


Figure 8: Avg. performance on lexical (top) and word scramble (bottom) tasks with time (x-axis). Accuracy increases with time as memory is filled up with feedback from past errors.

4.2 Using dynamic prefix in prompts

Recent work such as Liu et al. (2021a) investigate using dynamic prompts for better generation. For a given input \mathbf{x} , their method(KATE) relies on retrieving examples from the training set that are similar to \mathbf{x} for dynamically creating the prompt \mathbf{p} . Note that our method edits \mathbf{x} with a feedback \mathbf{fb} , and is thus complementary to KATE. To demonstrate this, we conduct experiments on ERT-CAT and ERT-NL tasks, where dynamic prompts were created using KATE, and MemPrompt was used to attach feedback to the question. Our results show a consistent 10% improvement when using both KATE and MemPrompt, indicating that the improvements are complementary.

4.3 MemPrompt with label feedback

MemPrompt requires the model to verbalize its understanding of the question, on which a user provides feedback. To investigate the efficacy of MemPrompt in settings where generating an understanding is not easy, we experiment with factual question answering on the WEBQA dataset (Berant et al., 2013), and find that MemPrompt is effective even with label feedback (Appendix §F).

4.4 Using MemPrompt for language and dialects based personalization

We demonstrate an application of MemPrompt for personalization with a use-case where user language preferences can be folded in the memory. We simulate a user who does not speak fluent English and uses code-mixed language. The queries posed by the user contain words from two Indian languages: Hindi and Punjabi. GPT-3 predictably misunderstands the task. The user clarifies the meanings of their dialect/language phrases. While initial queries fail, subsequent queries that reuse similar words succeed because their clarifications are present in the memory (details in Appendix §G).

5 Conclusion

We present MemPrompt, a novel, memory-enhanced GPT-3 that allows users to interact and improve the model without retraining. A key insight is to have the model articulate not just its answer but also its understanding of the user’s intent, providing an avenue for feedback. We show that deployed systems with fixed large-language models can still be improved by interacting with

end-users, potentially improving their performance and broadening their utility.

Acknowledgments

We thank Dheeraj Rajagopal and Yannic Kilcher for the insightful and engaging discussions. This material is partly based on research sponsored in part by the Air Force Research Laboratory (agreement number FA8750-19-2-0200). The U.S. Govt. is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

6 Limitations

We have shown how to improve very large models through interaction. Our memory-based enhancement is a low-cost utility enhancement eventually geared towards personalized, correctable models, which is currently an open question in NLP with unresolved issues. While our method is a step toward a promising open direction, it comes with limitations and opportunities when deploying to the real world.

Scaling In practical deployments of the MemPrompt method, the memory can grow to orders of magnitude, introducing scaling challenges. We anticipate using memory as a buffer between cycles of re-training, and these cycles could range from a week to several months. Between cycles of re-training, MemPrompt can serve as a way to avoid repeating mistakes and collect feedback which can be used to fine-tune and improve the next version of the model.

Currently, we operate with *a single user* at a time, but a real-world deployment could encounter multiple users. These users could exhibit characteristics of a user community where some feedback could apply to multiple users in a community cluster, while others differ in interpretation and style. In such a multi-user environment, managing the memory effectively when dealing with incompatible entries would be important. Existing initial ideas towards managing a bank of beliefs could be extended to address these problems, e.g., (Kassner et al., 2021). In addition, when looking up such

a rich and potentially noisy feedback collection, rather than retrieving a single feedback item, it would help to have an adapter over the memory that generates feedback by adapting the existing, diverse, and related past feedback to the current scenario. This increases the diversity of the generated knowledge and reduces the impact of erroneous feedback and noise.

Ethical concerns Extending the discussion on noise in feedback, our setting assumes that users will not provide any *adversarial* feedback. However, in real-world environments, this assumption is unlikely to hold. Additionally, there is a risk in the real-world deployment of our system, wherein an adversarial user might provide harmful feedback, thus maliciously controlling the systems (potentially a home-based robot) where our method is deployed. Thus, robust mechanisms such as GUD-IR and memory adapters will be critical for successful real-world deployments.

Privacy is another ethical concern, as the deployed system collects and records feedback from a user, some of which could contain personal information (*when I look for an interesting movie, I mean something that contains romance*). Therefore, the system needs to win the trust of the users so they would be encouraged to interact closely, and to win this trust, the system needs to demonstrate smartness, receptivity to user feedback, and the ability to maintain the memory without leaking any personal information safely.

Finally, large-language models generate text that might be biased and insensitive to a user’s socio-cultural context (Bordia and Bowman, 2019; Sharma et al., 2021; Hovy and Prabhumoye, 2021). In a multi-user deployment of our system, the memory could contain feedback from user communities of diverse beliefs, gender identities, and cultural backgrounds could lead to conflicts. Thus the system will need checks and balances to ensure that the content produced by the system as a result of the feedback is not harmful.

References

- Emily M. Bender and Alexander Koller. 2020. [Climbing towards NLU: On meaning, form, and understanding in the age of data](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy

- Liang. 2013. [Semantic parsing on Freebase from question-answer pairs](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.
- Shikha Bordia and Samuel R. Bowman. 2019. [Identifying and reducing gender bias in word-level language models](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2022. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354.
- Randall Davis. 1977. Interactive transfer of expertise: Acquisition of new inference rules. *Artif. Intell.*, 12:121–157.
- 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.
- Ahmed Elgohary, Christopher Meek, Matthew Richardson, Adam Fourney, Gonzalo Ramos, and Ahmed Hassan Awadallah. 2021. [NL-EDIT: Correcting semantic parse errors through natural language interaction](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5599–5610, Online. Association for Computational Linguistics.
- Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. [Social chemistry 101: Learning to reason about social and moral norms](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 653–670, Online. Association for Computational Linguistics.
- Kelvin Guu, Kenton Lee, Z. Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.
- Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2021. [Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs](#). *ArXiv preprint*, abs/2111.13654.
- Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. [Learning knowledge graphs for question answering through conversational dialog](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 851–861, Denver, Colorado. Association for Computational Linguistics.
- Dirk Hovy and Shrimai Prabhumoye. 2021. Five sources of bias in natural language processing. *Language and Linguistics Compass*, 15(8):e12432.
- Larry L. Jacoby and Christopher N. Wahlheim. 2013. On the importance of looking back: The role of recursive reminders in recency judgments and cued recall. *Memory & Cognition*, 41:625–637.
- Liwei Jiang, Jena D Hwang, Chandra Bhagavatula, Roman Le Bras, Maxwell Forbes, Jon Borchardt, Jenny Liang, Oren Etzioni, Maarten Sap, and Yejin Choi. 2021. [Delphi: Towards machine ethics and norms](#). *ArXiv preprint*, abs/2110.07574.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.
- Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. Beliefbank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861.
- 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.
- Teven Le Scao and Alexander Rush. 2021. [How many data points is a prompt worth?](#) In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, 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. 2020. [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.*
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. [CommonGen: A constrained text generation challenge for generative commonsense reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021a. [What Makes Good In-Context Examples for GPT-3?](#) *ArXiv preprint*, abs/2101.06804.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *arXiv preprint arXiv:2107.13586*.
- Aman Madaan and Amir Yazdanbakhsh. 2022. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*.
- Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. [Generation-augmented retrieval for open-domain question answering](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.
- Gary Marcus. [Experiments testing gpt-3’s ability at commonsense reasoning: results](#). [online]. 2021.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *ArXiv*, abs/2104.08773.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. [Fast model editing at scale](#). *arXiv preprint arXiv:2110.11309*.
- Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. [Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459, Berlin, Germany. Association for Computational Linguistics.
- Xiaoman Pan, Kai Sun, Dian Yu, Jianshu Chen, Heng Ji, Claire Cardie, and Dong Yu. 2019. [Improving question answering with external knowledge](#). In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 27–37, Hong Kong, China. 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*, 21:1–67.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- C. Riesbeck. 1981. Failure-driven reminding for incremental learning. In *IJCAI*.
- Roger Schank. 1983. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press.
- Shanya Sharma, Manan Dey, and Koustuv Sinha. 2021. [Evaluating gender bias in natural language inference](#). *arXiv preprint arXiv:2105.05541*.
- Douglas Summers-Stay, Claire Bonial, and Clare Voss. 2021. [What can a generative language model answer about a passage?](#) In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 73–81, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. [Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Niket Tandon, Aman Madaan, Peter Clark, and Yiming Yang. 2022. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. *NAACL Findings (to appear)*.
- Sida I. Wang, Percy Liang, and Christopher D. Manning. 2016. [Learning language games through interaction](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2368–2378, Berlin, Germany. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652.

A Inside MemPrompt: Populating and using the memory

MemPrompt maintains a growing memory of recorded cases where a feedback was provided to clarify the user’s misunderstood intent. This flow is presented in Figure 9 that shows a sequence of steps 1-5 on how the memory is populated.

MemPrompt also produces enhanced prompts for any new query based on the user feedback on similar cases recorded previously in the memory. Figure 10 presents the sequence of steps 1-3 involved in retrieving and applying a past feedback on a similar case.

B Generative IR (GUD-IR)

A note on feedback and understanding Feedback \mathbf{fb} and understanding \mathbf{u} are two concepts that we repeatedly use in this work. Briefly, MemPrompt requires a model to spell out its understanding of the instruction (\mathbf{u}). The user can then provide a feedback \mathbf{fb} on the understanding. In the prompt, both \mathbf{fb} and \mathbf{u} are identical. Such examples are of the form $\mathbf{x}, \mathbf{u} \rightarrow \mathbf{u}, \mathbf{y}$ and their main purpose is to reinforce that model the input feedback \mathbf{u} be used to generate the output.

B.1 Introduction

One of the key strengths of MemPrompt is its ability to leverage feedback provided on earlier inputs \mathbf{x} to improve a current input. This is achieved by retrieving a feedback from memory \mathcal{M} using \mathbf{x} as the key. An underlying assumption of this process is that similar inputs will admit similar feedback, allowing us to use the feedback provided for one situation on another. For two input situations s_i and s_j with respective feedback \mathbf{fb}_i and \mathbf{fb}_j , this assumption can be succinctly stated as:

$$s_i \sim s_j \implies \mathbf{fb}_i \sim \mathbf{fb}_j$$

The ethical reasoning dataset with natural language feedback, ERT-NL, provides a unique challenge for this assumption because lexically dissimilar situations might have the same feedback. As a concrete example, consider an input situation s_i : *tom hated skating because he had no sense of balance* – with a feedback \mathbf{fb}_i : *this question is about practicing more when you want to improve your skills*. Suppose that our system has already seen s_i and has received a feedback \mathbf{fb}_i (i.e., there is an entry in \mathcal{M} : $s_i \rightarrow \mathbf{fb}_i$). Next, suppose a user enters

a new situation s_j : *jordyn was trying to improve her soccer skills*. As usual, MemPrompt will try to retrieve feedback for a *similar* situation. However, such retrieval is going to be challenging, because s_i (*tom hated skating because he had no sense of balance*) has little to no overlap with s_j (*jordyn was trying to improve her soccer skills*), although humans can easily tell that both situations are about improving skills. Consequently, MemPrompt may fail to retrieve the relevant feedback \mathbf{fb}_i or worse, may retrieve a misleading feedback.

The fact that two ostensibly dissimilar inputs two inputs ($\mathbf{x}_i, \mathbf{x}_j$) may share the same feedback makes vanilla retrieval non-viable for our setting. We deal with this challenging situation with two different solutions of increasing complexity.

B.2 Initial approach: Learning a feedback similarity function

Since the surface level similarity of input situations is not enough to capture similarity of respective feedback, we attempt to learn a function f_θ that will map similar inputs \mathbf{x}_i and \mathbf{x}_j to similar representations if the corresponding feedback \mathbf{fb}_i and \mathbf{fb}_j are close to each other, and vice-versa. A natural choice is training an embedding function $f : \mathbf{x} \rightarrow \mathbb{R}^d$ supervised by $\cos(\mathbf{fb}_i, \mathbf{fb}_j)$ where \cos is the cosine similarity ($\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$). Thus, the objective function is:

$$\mathcal{L}_\theta = (\cos(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}_j)) - \cos(\mathbf{fb}_i, \mathbf{fb}_j))^2$$

Intuitively, this objective function will encourage the similarity between the inputs ($\cos(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}_j))$) to be high when the corresponding feedback are similar, and vice-versa.

Feedback retrieval proceeds as follows: an input s_i is embedded using f_θ , and $f_\theta(s_i)$ is then used to retrieve a feedback from the memory, with the hope that representations $f_\theta(s_i)$ and $f_\theta(s_j)$ will be similar after the training.

While in principle this objective function should be enough to learn informative representations that bring two inputs with similar feedback close, we found the training to be unstable. We attribute this to the fact that two extremely dissimilar situations can have identical feedback. Given limited training data, it might be unrealistic to train similarity functions that can capture all possible cases where the same feedback applies to two situations. As a way to circumvent this, we also experiment with a generative version of our method, described next.

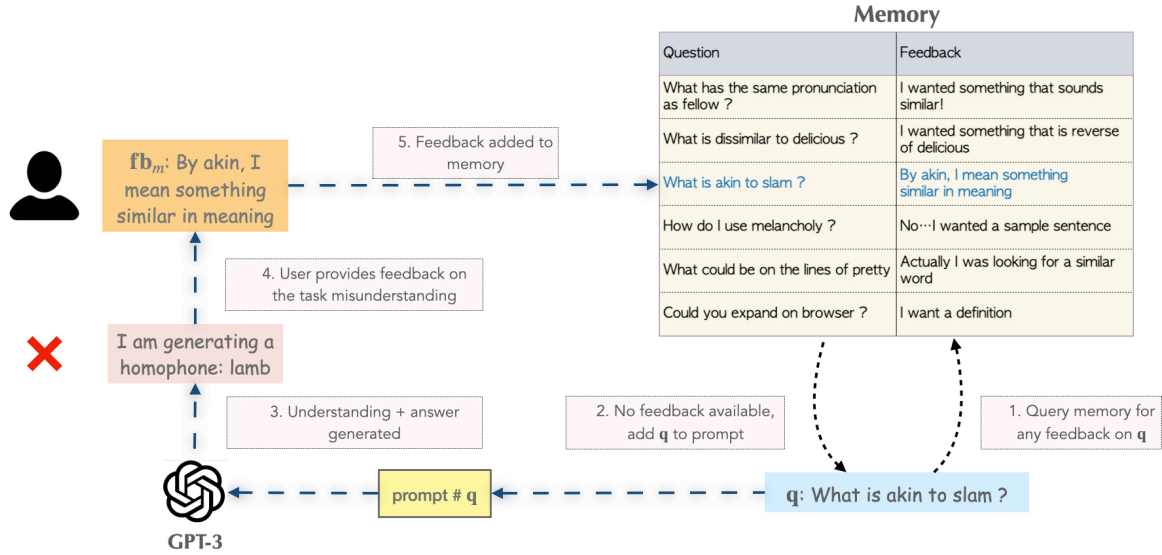


Figure 9: MemPrompt: adding to memory. User enters a question for which no feedback is available (steps 1, 2). Directly prompting GPT-3 with the question leads to incorrect answer and understanding (step 3). User-provides feedback on the incorrect understanding (step 4), which is added to memory (step 5).

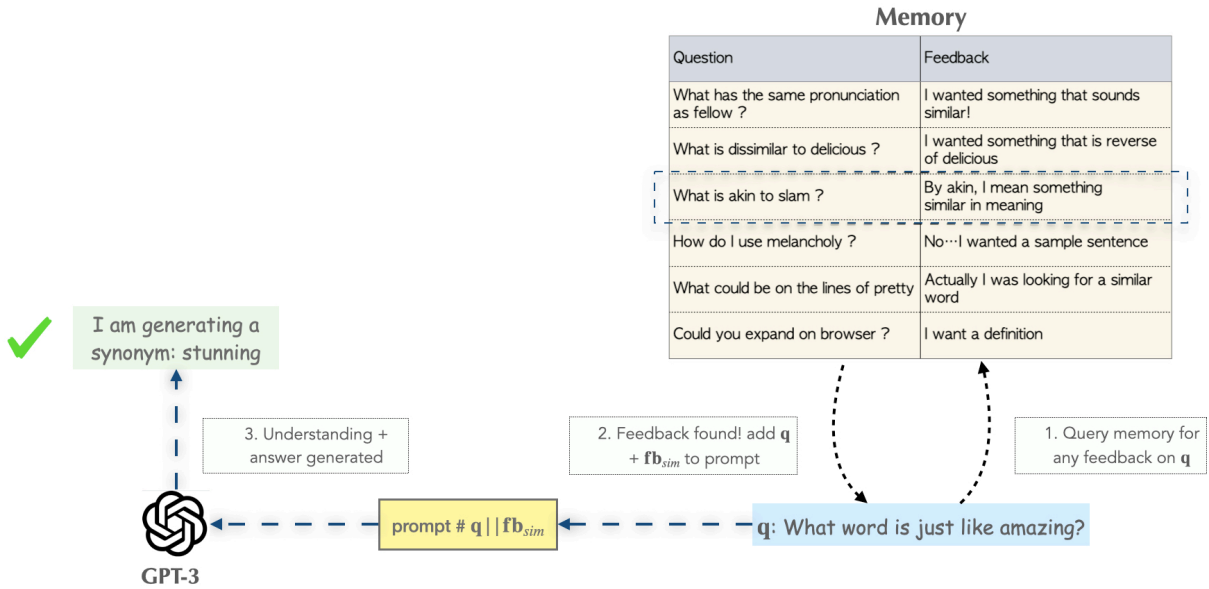


Figure 10: MemPrompt: retrieving feedback from memory. User enters a question which GPT-3 has incorrectly answered in the past, and has received feedback from a user (step 1). The feedback is retrieved from memory (step 2), and both question and feedback are added to the prompt. The prompt contains examples that allow GPT-3 to react to user feedback and generate correct understanding and answer.

B.3 Proposed approach: Training generative model for retrieving similar feedback

To address these retrieval issues, we propose **GUD-IR** (Generated UnDerstanding for explainable IR). The key intuition for our approach relies on substituting $f_\theta : \mathbf{x} \rightarrow \mathbb{R}^d$ (latent space projection) with $f_\theta : \mathbf{x} \rightarrow \mathbf{fb}$ (generated understanding of \mathbf{x}). Concretely, instead of learning a function that maps a question to a d dimensional vector, we train a

generative model that directly maps an input to a rough understanding. The generated rough understanding is then used as a key to retrieve a relevant understanding from the database using any off-the-shelf retrieval method. This two-step *generate-then-retrieve* procedure has benefits: (i) it alleviates sparsity issues that we found latent space projection

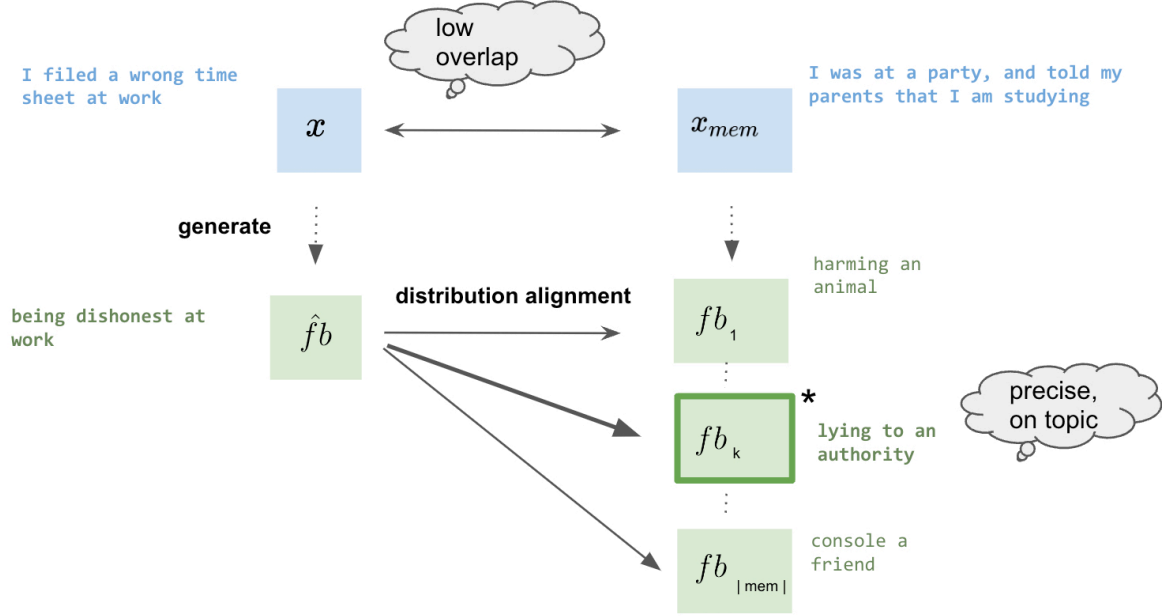


Figure 11: **Overview of GUD-IR.** To retrieve a relevant feedback that applies to x , GUD-IR first generates a feedback \hat{fb} using a generative model. This is then aligned with a corpus of feedbacks $fb_1, fb_2, \dots, fb_{|tr|}$ (e.g., sourced from the train split). The best matching feedback \hat{fb}^* is then used for x . Thus, GUD-IR decomposes the retrieval problem $x \rightarrow fb$ into two sub-problems: (i) generate a rough feedback ($x \rightarrow \hat{fb}$) and (ii) search for the closest feedback in a large store $\hat{fb}^* = \arg \min_{j \in [1, |tr|]} |\hat{fb} - fb_j|$.

methods were unable to deal with⁵ (ii) the overall retrieval becomes explainable and debuggable.

Our approach is inspired and supported by the recent success of generate and retrieve (Mao et al., 2021) methods. However, despite the similarity, the methods have different goals: Mao et al. (2021) leverage generative models for query expansion, whereas our goal is explainable input understanding. Moreover, their implementation is geared towards open-domain QA, while ours is towards explainable input understanding. Thus, it is non-trivial to adapt similar ideas to our tasks effectively.

Specifically, we train a SEQ2SEQ model, (e.g., T5 (Raffel et al., 2020)), that maps each input x to a corresponding output fb . The feedback is now retrieved in a two step process:

1. The generative model f_θ is used to generate a noisy feedback for s_i , \hat{fb} .
2. \hat{fb} is used as a key to *search* over the set of already present feedbacks, to retrieve the nearest one.

⁵e.g., there are only eight popular emotions but can lead to a large number of diverse situations. Hence, many inputs can map to the same principle fb . This mapping becomes increasingly difficult for a model as the specificity of fb increases, because of sparsity issues. This is exacerbated when the input situations are diverse and previously unseen.

Instead of directly using clarification to lookup the nearest feedback, we first transform the input to the space of clarifications, then search over the set of already present clarifications. Figure 11 presents an overview of our *generation then reshape* approach (GUD-IR). As we discuss in Section 4.1.1, GUD-IR was key to achieving good performance for the ERT-NL task.

In addition to the task accuracy, we plot the distribution of $\text{sim}(\hat{u}, \hat{u}^*)$ (similarity of the true and retrieved feedback) over the test set for different retrieval methods. Figure 12 shows this distribution using GUD-IR and using surface-level similarities. The probability mass shifts towards a higher similarity range for GUD-IR.

The lexical reasoning and WEBQA tasks present a simpler setting for retrieval, as similarity of keys indicates a similarity of values. For such cases, we use Sentence transformers (Reimers and Gurevych, 2019) to encode the query, and cosine similarity with a threshold of 0.9 to find a matching key.

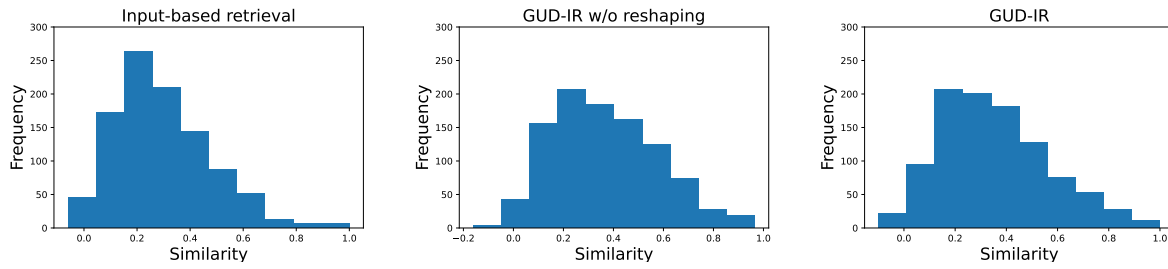


Figure 12: Distribution of similarity scores between expected \mathbf{fb}^* and \hat{u} for retrieval (left) and GUD-IR (right). The similarity scores are higher using GUD-IR.

C Querying GPT-3-175B using OpenAI API

We use the OpenAI API for querying GPT-3-175B.⁶ The python code is listed below. Here, “PROMPT” is set to prompt shown in §D, followed by the input question \mathbf{x} and feedback \mathbf{fb} if applicable.

We used a temperature of 0.0 for factual QA (WEBQA) experiments to select the most likely token at each step, and this setting does not require generating diverse answers, as one would expect for a factual domain. For ERT-CAT and ERT-NL, we found that a higher temperature (~ 0.7) was causing a large divergence in the performance (a difference of $\pm 10\%$ accuracy across runs), making reproducibility challenging – similar observations were made by (Summers-Stay et al., 2021). Thus, we used to a temperature of 0.0 for ERT experiments. A temperature of 0.7 was used for all the other experiments.

```
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")

response = openai.Completion.create(
    engine="davinci",
    prompt="PROMPT",
    temperature=0.7,
    max_tokens=64,
    top_p=1,
    frequency_penalty=0,
    presence_penalty=0
)
```

⁶<https://beta.openai.com/docs/introduction>, we use ‘text-davinci-001’

D Prompt

GPT3 is queried using a prompt \mathbf{p} of example i/o behaviors, followed by the actual question \mathbf{x} and (optionally) retrieved feedback \mathbf{fb} . It then generates the understood intent \mathbf{u} and answer \mathbf{y} as a continuation. \mathbf{u} and \mathbf{y} are expressed a single sentence, e.g., "[The synonym for <word> is] [<word>]" Figure 13 shows this prompt \mathbf{p} , containing a mixture of $(\mathbf{x} \rightarrow \mathbf{u}, \mathbf{y})$ and $(\mathbf{x}, \mathbf{fb} \rightarrow \mathbf{u}, \mathbf{y})$ "training" tuples.

What is the homophone for < wring > ?

#

the homophone for wring is ring END

#

how do you use < highway > in a sentence?

#

a sentence with highway is: soldiers stand guard along the [highway] END

#

can you define < camisole > ?

#

the definition of camisole is a sleeveless undergarment. END

#

What is the antonym for < prohibition > ?

#

the antonym for prohibition is permit END

#

What is the synonym for < surrogate > ?

#

the synonym for surrogate is substitute END

#

how do i use < fog > ? | clarification: when i ask for how do i use, i want a sentence.

#

a sentence with fog is: a rising sun burns the [fog] off a city END

#

What sounds like < sighted > ? | clarification: when I ask for sounds like, I want a homophone.

#

the homophone for sighted is cited END

#

what is like < provident > ? | clarification: when I ask for like, I want a synonym.

#

the synonym for provident is prudent END

#

can you define < rider > ? | clarification: when i ask for define, i want a definition.

#

the definition of rider is a person who is riding something. END

#

What is the opposite of < citation > ? | clarification: when I ask for opposite, I want an antonym.

#

the antonym for citation is award END

Figure 13: The prompt used for our tasks. During inference, an input question x_i , and optionally a feedback fb_i is appended after this prompt, and the model is expected to generate the answer y_i and its understanding of the question intent u_i as a continuation. The prompt contains examples of the form $(x \rightarrow u, y)$, expressed " $x \# u \ y$ END #", and $(x, fb \rightarrow u, y)$, expressed " $x \mid \text{clarification: } fb \# u \ y$ END #". (u and y are expressed together as a single sentence, e.g., "[The synonym for <word> is] [<word>].")

Find the right word after removing random letters from < t!r/e/a/s/u/r.e!s >

the word after removing symbols from t!r/e/a/s/u/r.e!s is treasures END

Find the original word after ignoring the punctuation and spaces in < e >

the word after removing symbols from e is elders END

Find the right word given this cycled word: < lprovisiona > ?

the uncycled version of lprovisiona is provisional END

Make a word while keeping the first and last char < vosiin > ?

the anagram 1 for vosiin is vision END

Find the original word that is interspersed in < f.i.n!e/p.i/x >

the word after removing symbols from f.i.n!e/p.i/x is finepix END

Find the right word given this rotated word: < cturalarchite > ?

the uncycled version of cturalarchite is architectural END

Find the original word after ignoring the punctuation and spaces in < s.e!n.t.i.n/e/l >

the word after removing symbols from s is sentinel END

Find the right word given this rotated word: < ibitioninh > ?

the uncycled version of ibitioninh is inhibition END

Figure out the word which has the same first two and the last two char < watsed > ? | clarification: when I want you to figure out the word which has the same first two and the last two char, I mean anagram 2.

the anagram 2 for watsed is wasted END

Make a word while keeping the first and last char < isucnase > ? | clarification: when I want you to make a word while keeping the first and last char, I mean anagram 1.

the anagram 1 for isucnase is issuance END

Unscramble everything except the first two and the last two char < acotrs > ? | clarification: when I want you to unscramble everything except the first two and the last two char, I mean anagram 2.

the anagram 2 for acotrs is actors END

Figure 14: The prompt used for our word scrambling tasks derived from [Brown et al. \(2020\)](#)

Situation: last day of high school food fight and I nailed a special needs girl square in the face with a heavy muffin.

Morality Judgment: This question is about: getting into food fights with people. The answer is It's bad. END

#

Situation: when phone shopping at stores, I used to write inappropriate stuff and post it to random peoples book of faces accounts who would forget to logoff on the display phones. This question is about: messing with people's social media accounts.

Morality Judgment: This question is about: messing with people's social media accounts. The answer is It's bad. END

#

Situation: i had an uncomfortable talk with my 10yo about puberty and feel like avoiding him for a few days.

Morality Judgment: This question is about: being honest with your kids. The answer is It's good. END

#

Situation: every time I see my three year old nephew being funny or I'm having a good time with him, my mind goes to a school shooting or car crash or kidnapping or some other horrible atrocity happening to him. I have absolutely no idea how to handle this and I'm begging for help.

Morality Judgment: This question is about: admitting you imagine bad things happening to a child and that you need help with this. The answer is It's good. END

#

Situation: i see someone who's not there. he's in my head. he screams sometimes when I'm too stressed and the only way I can get him to leave me alone, is when I cut myself .

Morality Judgment: This question is about: asking someone close to you who is having mental health issues how you can help them. The answer is It's okay. END

#

Situation: i heard once that if you cringe about your past it just means you've grown as a person, for anyone who needs to hear it. This question is about: regretting poor decisions from your past.

Morality Judgment: This question is about: regretting poor decisions from your past. The answer is It's okay. END

#

Situation: my cat hopped up in my lap, cleaned herself and went to sleep. now I'm a captive and I want to move but I can't. This question is about: waking up an animal because you would like to move.

Morality Judgment: This question is about: waking up an animal because you would like to move. The answer is It's okay. END

#

Situation: if it wasn't for the pain it would cause to my family, I would have thrown myself off a bridge a while ago.

Morality Judgment: This question is about: caring about your family's feelings. The answer is It's good. END

#

Figure 15: The prompt used for ERT-NL task. The user provides a situation and the model generates a morality judgement. In addition to the morality judgment, the model also provides a high-level understanding of what the task is about.

Situation: i am selectively racist towards "generic" indian recruiters because they waste my time with absolute zero results.

Morality Judgment: This question is about: trating you differently because of your race. The answer is It's bad. END

#

Situation: i told my husband I didn't feel well so I could stay in bed and play red dead redemption 2 all day and night. This question is about: playing hooky every once in awhile.

Morality Judgment: This question is about: playing hooky every once in awhile. The answer is It's bad. END

#

Situation: I've misunderstood the phrase 'when the going gets tough the tough get going' for my entire life. This question is about: not understanding cliches when young.

Morality Judgment: This question is about: not understanding cliches when young. The answer is It's okay. END

#

Situation: when phone shopping at stores, I used to write inappropriate stuff and post it to random peoples book of faces accounts who would forget to logoff on the display phones. This question is about: messing with people's social media accounts.

Morality Judgment: This question is about: messing with people's social media accounts. The answer is It's bad. END

#

Situation: i had an uncomfortable talk with my 10yo about puberty and feel like avoiding him for a few days.

Morality Judgment: This question is about: being honest with your kids. The answer is It's good. END

#

Situation: every time I see my three year old nephew being funny or I'm having a good time with him, my mind goes to a school shooting or car crash or kidnapping or some other horrible atrocity happening to him. I have absolutely no idea how to handle this and I'm begging for help.

Morality Judgment: This question is about: admitting you imagine bad things happening to a child and that you need help with this. The answer is It's good. END

#

Situation: i heard once that if you cringe about your past it just means you've grown as a person, for anyone who needs to hear ito. This question is about: regretting poor decisions from your past.

Morality Judgment: This question is about: regretting poor decisions from your past. The answer is It's okay. END

#

Situation: my cat hopped up in my lap, cleaned herself and went to sleep. now I'm a captive and I want to move but I can't. This question is about: waking up an animal because you would like to move.

Morality Judgment: This question is about: waking up an animal because you would like to move. The answer is It's okay. END

#

Situation: if it wasn't for the pain it would cause to my family, I would have thrown myself off a bridge a while ago.

Morality Judgment: This question is about: caring about your family's feelings. The answer is It's good. END

Figure 16: The prompt used for ERT-CAT task. The user provides a situation and the model generates a morality judgement. In addition to the morality judgment, the model also provides a high-level understanding of what the task is about.

E Datasets for lexical question-answering tasks

As mentioned in Section §4, we focus on five different linguistic QA tasks. The source of data for each of these tasks is listed below:

1. The synonyms (syn) and antonyms (ant) were obtained from [Nguyen et al. \(2016\)](#).⁷
2. The homophones (hom) were obtained using homz <https://github.com/cameronehrlich/homz>. We use the closest homophone returned by homz for each word in the English dictionary.
3. The definitions (defn) were sourced from *The Online Plain Text English Dictionary* <https://github.com/eddydn/DictionaryDatabase>
4. Examples for usage in a sentence (sent) are from CommonGen ([Lin et al., 2020](#)).

E.1 Templates

We manually created 15 task templates with three variants of phrasing the question for each task. Sample templates are shown in code listing 1. The data (word1, word2) in the code is initialized with the entries in the four sources mentioned above. The complete file is available in the project repository <https://github.com/madaan/memprompt/tree/main/src/templates>.

E.2 Sample questions

Tables 10, 11, and 11 list some sample x-y for settings where the question was asked as a linguistic variation, in Hindi, and in Punjabi, respectively.

F MemPrompt with label feedback

Our current approach requires the model to verbalize its understanding of the question, on which a user provides feedback. Such a setup might not be possible, for instance, due to the nature of questions. Can MemPrompt be effectively used in such settings as well? To investigate this, we experiment with factual question answering on the WEBQA dataset ([Berant et al., 2013](#)), and use the test

set provided by [Berant et al. \(2013\)](#) for all experiments (2032 questions). The WEBQA dataset consists of factual questions (*which language is spoken in Canada?*) with multiple answers (*English, French*), and is a popular dataset for benchmarking the performance of GPT-3 on question answering in a few-context prompting setup.

Inference Let k be the number of examples (i.e., question-answer pairs) in the prompt. For a given question q , We keep half ($k/2$) examples fixed in the prompt, whereas the other half $k/2$ examples are retrieved from a memory of feedback M . As before, on receiving a question q , consults a memory M to see if a similar question has been asked before. However, different from earlier setups, in this case, we retrieve $k/2$ most similar questions from the memory M on **which the system has been wrong earlier**. The corresponding true answers are also retrieved. These $k/2$ retrieved question-answer pairs are combined with the $k/2$ fixed questions to create a prompt, and query GPT-3. Let a' be the generated answer.

Growing memory of errors M In our setup, we assume an expert user (or a teacher) that knows the true answer a for a given query q . The expert user compares the GPT-3 generated answer a' with a . If the generated answer is correct ($a' = a$), no further action is taken. If not, the entry $((q, a))$ is added to the memory M . As time passes, M is populated with an increasing number of challenging examples that the model has been wrong on. Thus, the retrieved $k/2$ examples get more relevant with time, aiding the accuracy. In the experiments, we set $k = 16$ due to budget constraints (note that the setups used in [Liu et al. \(2021a\)](#) and [Brown et al. \(2020\)](#) set $k = 64$, but their results are comparable to our baseline with $k = 16$).

Results Similar to ERT and word reasoning tasks, a memory of errors helps in increasing accuracy with time over 3,000 points in the test split of the WEBQA dataset (Figure 17). This is expected, as M gathers more examples on which GPT-3-175B has been wrong before. Adding these examples in the prompt avoids the model in repeating these mistakes.

To check if examples that belong to a similar domain improve with time, we cluster the questions in the test set of WEBQA, and randomly select three clusters for our analysis. Table 8 shows the top three of the 8 ($k = 16/2$) examples retrieved

⁷<https://www.ims.uni-stuttgart.de/en/research/resources/experiment-data/lexical-contrast-dataset/>

%	Question (q)	Example 1	Example 2	Example 3
1	what highschool did harper lee go to?	what did st augustine do?	who is keyshia cole dad?	when did charles goodyear invented rubber?
4	what college did albert einstein go to?	what highschool did harper lee go to?	who did tim tebow play college football for?	what timezone is utah in?
40	where did john mayer go to college?	what school did michael jackson go to high school?	where did derek fisher go to college?	what style of music does john mayer play?
90	where did scott fitzgerald go to college?	who was f. scott fitzgerald?	where did otto frank go to college?	where did derek fisher go to college?

Table 6: Relevant examples fetched with time: as time proceeds, the examples fetched from the memory become increasingly relevant to the input question, leading to increasingly accurate predictions.

Timestep	Question	Neighbor
12	is not wanting a dirty person harassing you morally ok?	is not wanting to hear foul language morally good?
70	when adam decided to stop living a lie and announces he's gay, are you expected to pretend you're straight when you're gay?	are you expected to pretend you're straight when you're gay?
85	Should I help someone when they are having trouble?	must you intervene if you see someone taking advantage of another disabled person?
230	needing a hug to feel good.	is wanting to cuddle with loved ones okay?

Table 7: Relevant examples retrieved at increasing timesteps: as time proceeds, the examples fetched from the memory become relevant to the input question, leading to accurate predictions.

from M for the *alma mater* cluster.⁸ All of these questions relate to the alma mater of famous personalities. As the inference begins (with an empty M), the examples are not relevant to q . However, towards the end, almost all the samples are relevant to the given question.

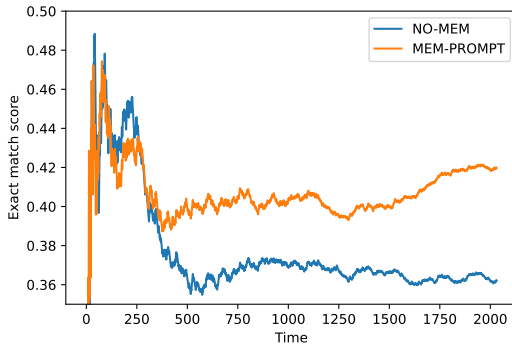


Figure 17: Instruction accuracy vs. time for WEBQA.

F.1 Factual question answering Examples

Tables 8 and 9 show additional examples for questions from WEBQA which get additionally relevant examples as time proceeds. The examples include questions that belong to the domains of Alma mater, Soccer, and Language.

⁸Additional examples are included in Appendix §F.1.

G Finding similar questions in low-resource settings

We also experimented using queries in Hindi and Punjabi, with (English) feedback clarifying the queries' intent when GPT3 predictably misunderstands the task. Figure 18 confirms significant gains using memory in this OOV setting. This setup highlights the case when the user does not speak fluent English and uses mixed language code, e.g., transcription in English and mixing words from another language to ask questions.

In low-resource settings (e.g., queries in transcribed Punjabi or Hindi), we perform similarity matching between a given question and a question in the memory by using surface-form similarity. Specifically, we use Levenshtein distance to determine the closest query in the memory. We note that as the memory grows large, we can use mechanisms such as FAISS (Johnson et al., 2019) for trained memory, and suffix-trees for fast retrieval using surface form similarity.

H Sample results

Table 13 shows randomly sampled x-y pairs, and the corresponding y generated by GPT-3-175B and MemPrompt. The complete set of outputs is located in the project repository <https://github.com/madaan/memprompt/tree/main/results>.

Domain	% Finished	Question	Neighbor 1	Neighbor 2	Neighbor 3
Alma mater	1	what high-school did harper lee go to?	what did st augustine do?	who is keyshia cole dad?	when did charles goodyear invented rubber?
Alma mater	5	what college did albert einstein go to?	what high-school did harper lee go to?	who did tim tebow play college football for?	what timezone is utah in?
Alma mater	10	what university did gordon brown attend?	what all does google now do?	what team did david beckham play for in 2011?	who did tim tebow play college football for?
Alma mater	40	where did john mayer go to college?	what school did michael jackson go to high school?	where did derek fisher go to college?	what style of music does john mayer play?
Alma mater	75	where did john steinbeck go to college?	where did john mayer go to college?	what college did john stockton go to?	where did otto frank go to college?
Alma mater	95	where did scott fitzgerald go to college?	who was f. scott fitzgerald?	where did otto frank go to college?	where did derek fisher go to college?
Soccer	1	what team did david beckham play for in 2011?	who did tim tebow play college football for?	what super bowl did peyton manning win?	what type of music did john lennon sing?
Soccer	25	what team did ronaldo play for in 2003?	what part did winona ryder play in star trek?	what to do in richardson dal-las?	who did the voice of darth vader in episode 3?
Soccer	33	who did nasri play for before arsenal?	what year did ray allen join the nba?	who does donnie wahlberg play in the sixth sense?	what does david beckham play?
Soccer	65	who has pudge rodriguez played for?	who does nolan ryan play for?	who did carlos boozer play for?	who does ronaldinho play for now 2011?
Soccer	99	what team did david beckham play for before la galaxy?	who does david beckham play for?	what does david beckham play?	what team does david beckham play for in 2012?

Table 8: Relevant examples retrieved for WEBQA QA task (Section §4.3). The retrieved examples get increasingly relevant as time proceeds.

Domain	% Finished	Question	Neighbor 1	Neighbor 2	Neighbor 3
Language	1	what does ja- maican people speak?	when was an- cient egypt cre- ated?	where is the denver bron- cos stadium located?	what is the name of the capital of spain?
Language	20	what are the two official languages of paraguay?	what do por- tuguese people speak?	what language does cuba speak?	where is mission san buenaventura located?
Language	37	what language does colombia?	what language does cuba speak?	what was the first language spoken in spain?	what is ser- bian language called?
Language	85	what language does peru speak?	what are the of- ficial languages of the eu?	where is the latin language from?	what do por- tuguese people speak?
Language	90	what language do they speak in colombia south america?	how many lan- guages do they speak in spain?	where is the latin language from?	what language does cuba speak?

Table 9: Relevant examples retrieved for WEBQA QA task (Section §4.3). The retrieved examples get increasingly relevant as time proceeds.

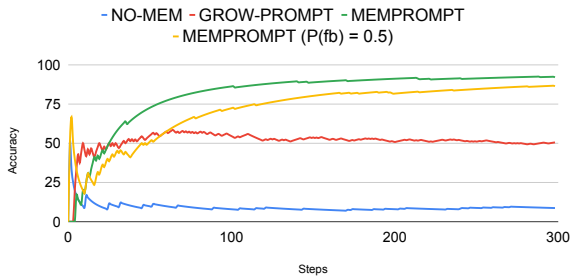


Figure 18: **Finding 2** Large gains on queries asked in English and Punjabi by MemPrompt.

```

1 templates = [
2     {
3         "type": "syn",
4         "template_id": "syn1",
5         "question": lambda word1: f"What is similar to < {word1} > ?",
6         "question_clarification": lambda word1: f"What is similar to < {word1} > ? | clarification: when I ask for similar to , I want a synonym.",
7         "clarification": "clarification: when I ask for similar to , I want a synonym.",
8         "answer": lambda word1, word2: f"the synonym for {word1} is {word2}",
9     },
10    {
11        "type": "ant",
12        "template_id": "ant0",
13        "question": lambda word1: f"What is unlike < {word1} > ?",
14        "question_clarification": lambda word1: f"What is unlike < {word1} > ? | clarification: when I ask for unlike , I want an antonym.",
15        "clarification": "clarification: when I ask for unlike , I want an antonym.",
16        "answer": lambda word1, word2: f"the antonym for {word1} is {word2}",
17    },
18    {
19        "type": "defn",
20        "template_id": "defn0",
21        "question": lambda word: f"< {word} > means what ?",
22        "question_clarification": lambda word: f"< {word} > means what ? | clarification: when I ask for means what , I want a definition.",
23        "clarification": "clarification: when I ask for means what , I want a definition.",
24        "answer": lambda word, definition: f"the definition of {word} is {definition}",
25    },
26    {
27        "type": "sent",
28        "template_id": "sent1",
29        "question": lambda word: f"< {word} > can be used how ?",
30        "question_clarification": lambda word: f"< {word} > can be used how ? | clarification: when I ask for can be used how , I want a sentence.",
31        "clarification": "clarification: when I ask for can be used how , I want a sentence.",
32        "answer": lambda word, sentence: f"a sentence with {word} is: {sentence}",
33    }
34 ]

```

Listing 1: Sample templates for the five tasks.

Question (x)	Answer (y)	type
What is the opposite of < misconstrue > ?	the antonym for misconstrue is verify	ant
What is the opposite of < gross > ?	the antonym for gross is polite	ant
expand on < chelicera > ?	the definition of chelicera is One of the anterior pair of mouth organs	defn
what is a sentence that can be used to define < mawseed > ?	the definition of mawseed is The seed of the opium poppy.	defn
what has a < bitt > like ring to it ?	the homophone for bitt is bit	hom
what can one confuse with < holed > ?	the homophone for holed is hold	hom
< spread > can be used how ?	a sentence with spread is: a couple of sheep are spread out in a field	sent
make something with < pot > ?	a sentence with pot is: bonsai tree in pot at zen garden .	sent
What is akin to < musician > ?	the synonym for musician is instrumental-ist	syn
What is akin to < zigzag > ?	the synonym for zigzag is move	syn

Table 10: Sample x-y pairs in English. The same type of question can be asked in multiple ways. Our method makes no assumptions as to how a question might be asked.

Question (x)	Answer (y)	type
< tabulate > ka ulta kya hai ?	the antonym for tabulate is randomize	ant
< foot > ka vilom kya hai ?	the antonym for foot is head	ant
< lettish > ka matlab kya hota hai ?	the definition of lettish is The language spoken by the Letts. See Lettic.	defn
< housing > ka arth kya hai ?	the definition of housing is An appendage to the hames or collar of a harness.	defn
sunne mai < perl > jaisa kya hai ?	the homophone for perl is pearl	hom
< council > jaisa kya sunai deta hai ?	the homophone for council is conceal	hom
< city > ko ek vakya mai kaise likhen ?	a sentence with city is: the city takes on an even more interesting hue during event	sent
< fly > ko ek vakya mai kaise likhen ?	a sentence with fly is: airplane fly into a storm cloud	sent
< critique > kai samaan kya hota hai ?	the synonym for critique is evaluate	syn
< psychiatric > kai samaan kya hota hai ?	the synonym for psychiatric is mental	syn

Table 11: Sample x-y pairs in Hindi.

Question (x)	Answer (y)	type
< edit > de ult ki hunda ae ?	the antonym for edit is expand	ant
< spring > ton bhin ki ae ?	the antonym for spring is compress	ant
< patricide > di paribhasha dasso ?	the definition of patricide is The crime of one who murders his father. Same as Parricide.	defn
< sankhya > da matlab ki hunda ae ?	the definition of sankhya is A Hindoo system of philosophy which refers all things to soul and a rootless germ called prakriti	defn
sunnan vich < site > varga ki ae ?	the homophone for site is sites	hom
< chance > da samnam ki ae ?	the homophone for chance is chants	hom
< enter > nu ek vak vich kidan vartiye ?	a sentence with enter is: as you enter you will be greeted by these rows of trees .	sent
< tree > da prayog ki ae ?	a sentence with tree is: wide shot of an eagle and its nest on top of a tree	sent
< unlawful > jidan ki hunda ae ?	the synonym for unlawful is illegal	syn
< constrict > jidan ki hunda ae ?	the synonym for constrict is close	syn

Table 12: Sample x-y pairs in Punjabi.

Question (x)	Expected answer (y *)	GPT-3 generated answer	MemPrompt generated answer
give me something i would hear as < whether > ?	the homophone for whether is whither	the example of something you would hear as whether is a question word used to ask about possibilities.	the homophone for whether is weather
what has a < mane > like ring to it ?	the homophone for mane is made	the word that has a mane like ring to it is mane	the homophone for mane is main
What is like < confine > ?	the synonym for confine is hold	the homophone for confine is confined	the synonym for confine is restrain
< leper > means what ?	the definition of leper is a person affected with leprosy.	a sentence with leper is: a leper is a person with a disease that makes his skin scaly and white.	the definition of leper is a person who has leprosy.
expand on < chaperon > ?	the definition of chaperon is a matron who accompanies a young lady in public	the expansion of chaperon is one who chaperons.	the definition of chaperon is a woman who goes with a young girl to social functions.
what is a sentence that can be used to define < wassail > ?	the definition of wassail is an ancient expression of good wishes on a festive occasion	a sentence with wassail is: < wassail > wassails the christmas season	the definition of wassail is to drink to the health of.
expand on < paralbumin > ?	the definition of paralbumin is a proteidlike body found in the fluid from ovarian cysts and elsewhere. it is generally associated with a substance related to	the expansion of paralbumin is an albuminoid globulin occurring in the blood of the horse and other animals, and in the urine of humans.	the definition of paralbumin is a particular kind of albumin.
what can one confuse with < kew > ?	the homophone for kew is key	a word that can be confused with kew is due	the homophone for kew is cue
What is like < pneumatic > ?	the synonym for pneumatic is windy	the homophone for pneumatic is hydraulic	the synonym for pneumatic is airy
give me something i would hear as < seeder > ?	the homophone for seeder is seer	something you would hear as seeder is: someone who plants things in the ground or causes seeds to be planted.	the homophone for seeder is said

Table 13: A subset of random samples where GPT-3-175B without memory was incorrect.