Memory-assisted prompt editing to improve GPT-3 after deployment

Anonymous ACL submission

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 homonym, 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 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 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 is to pair 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., 2021c) 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 was “The homonym 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 that they actually wanted a synonym.

Our memory enhanced GPT-3 implementation.

User: What word is similar to good?

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

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

GPT-3: Noted [writes to memory]

User: What word is similar to surprised?

GPT-3: [Retrieves and adds to prompt ““Similar to" means "with a similar meaning””].

The synonym of surprised is: amazed.

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.
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 $M$ 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 sets out the general architecture and a simple implementation of its components. 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) ethics (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 (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. Our contributions are thus a general architecture and an implementation showing how user feedback might continuously improve model performance, without retraining, in a few-shot prompt setting.

2 Related work

Our method builds upon the recent advances in prompt-tuning and few-shot prompting.

Our use of recalled memories is a form of “prompt engineering”, where GPT-3’s behavior is modified by adding to the query (prompt) (Le Scao and Rush, 2021). 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 (Liu et al., 2021a), or even 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 feedbacks.

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 for 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.

Given an erroneous answer, our approach aims to modify the model’s behavior through prompting. An alternative, recently explored approach is “model editing” - updating the model itself by modifying its parameters to fix erroneous answers.
(Mitchell et al., 2021; De Cao et al., 2021; Hase et al., 2021). However, model editing approaches have to date only been demonstrated in a limited context (e.g., correcting a single error), and even then can lead to uncontrollable out of scope changes (Mitchell et al., 2021). In contrast, our goal is not just to correct a specific prediction, but to generalize that correction for new problems by collecting feedback to clarify the misunderstanding and without risking damage to the model’s basic problem-solving acumen.

Finally, our work is a simple example of debugging and learning via dialog. While system debugging through dialog has been explored in many contexts, e.g., (Hixon et al., 2015; Wang et al., 2016; Davis, 1977), our novel contribution is dialog 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 B). The user can then critique $u$ by providing natural language 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, MEM-PROMPT uses $fb$ from similar, prior queries to enrich the (few-shot) prompt $p$. We use the principle that if $x_i$ and $x_j$ have similar errors (i.e., $x_i \sim x_j$), then their feedbacks $fb_i$ and $fb_j$ should be exchangeable ($x_i \sim x_j \implies fb_i \sim fb_j$). Fig. 2 gives an overview of MEM-PROMPT, with the following components:

- **Memory $M$**: $M$ 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 $\Omega(x, M)$**: $\Omega$ is a learned retriever that matches the query=$x$ against all the keys of $M$.

- **Combiner $C(x, \Omega(x, M))$**: A gating function allowing irrelevant, retrieved feedback to be ignored.

- **Prompter $P(p, C)$**: $P$ passes the output of $C$ to GPT-3 prompt.

**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. 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.

$P$ supplements this few-shot prompting workflow, with a memory of user feedbacks from $C()$. To enable the model to react to such feedback, we include $k$ samples of the form $(x, fb \rightarrow u, y)$ in the prompt, so the question contains $fb$.

3.2 Feedback on model’s understanding

In the setup $(x \rightarrow u, y)$, there are three modes of failure for a model:

- **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 (error on $u$)**: when the model understands the task type, but misunderstands the subtle intent in a question.

- **Task modeling**: if the task is clearly understood, but the answer is not correct, then it requires updating the model parameters. Existing approaches do not scale to very large LMs such as GPT-3, see Section §2 for related work on model editing.

The first two failure modes are due to the inability of the model to understand the input, and are our focus for this work. This paper provides an architecture for a user to critique on model failures.

Both these types of understanding (task understanding and task intent understanding) can be critiqued via feedback. In the case of NL feedback, the user may provide a counter argument refuting $u$ as the possible intent. Similarly, in the case of categorical feedback, the user can critique the category provided by the model as its understanding of the situation. While feedback on the model output is our primary goal, we also experiment with settings where an Oracle is available to provide feedback on the labels (Section §4.3).

The model reacts to the feedback because some in-context samples are of the form: $(x, fb \rightarrow u, y)$ and $(x \rightarrow u, y)$. We consider a diverse set of tasks $(x \rightarrow y)$, $fb$ and $u$, summarized in Table I.
We use the original dataset by Brown et al. (2020). The lexical relation task is to predict a word with synonyms, Figure 1, and anagrams by changing all but the first and last characters. There are four transformation operations: reversal of words (rev), random insertions (rand), and anagrams by changing all but the first and last characters. 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), yupp (→ puppy), cycle letters in word (cyc, atc → cat), random insertions (rand, clr ic/ke!t → cricket), and anagrams by changing all but the first and last (anag1, elephnat → elephant) or all but the first and last 2 characters (anag2, elapehnt → elephant).

We use the original dataset by Brown et al. (2020). 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 can use 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.

3.3 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, a user can still point out this problem and provide feedback.

We use a subset 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.

We use a subset 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 freeform 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 $\mathbf{x}$, with a short feedback $\mathbf{fb}$. Specifically, we use the top 1% of the inputs by length to create a challenging set of input situations ($\mathbf{x}$). User feedback $\mathbf{fb}$ is a natural language feedback on the understanding of user $\mathbf{u}$. ERT-NL serves as the most challenging case in our setting. This is in part because our setup relies on the hard problem of retrieving questions that would assume similar feedback.

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.

3.4 MEM-PROMPT Implementation

Implementation of memory $\mathcal{M}$ We implement $\mathcal{M}$ using $\mathbf{x}$ as the key and the corresponding feedback $\mathbf{fb}$ as value. Given a question $\mathbf{x}_j$, if the user detects that the model has misunderstood the question, they may provide a $\mathbf{fb}_i$ with probability $Pr(f_i)$. The feedback 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 $\Omega$ (described below) 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, the model might misunderstand a question asking for synonym, e.g., what is akin to fast? as one that requires 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 $\mathbf{u} = \text{The opposite of fast is:}$ 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 3 illustrates a sample memory for this task.

Implementation of retriever $\Omega$ An incorrect feedback might cause the model to make a mistake, thus necessitating a good retrieval function. In our setting, we use two different retrieval functions:

1) Semantic similarity: the query is encoded using Sentence transformers (Reimers and Gurevych, 2019), and we use cosine distance with a threshold of 0.9 to find a matching key $\mathbf{x}_m$.

2) Lexical similarity: We also experiment with low-resource settings for which trained retrieval is not an option. In such cases, we rely on heuristics for similarity matching (details in Appendix §D).

Implementation of combiner $C$ $C$ concatenates $\mathbf{x}$ and $\mathbf{fb}$ retrieved by $\Omega$. 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.

Implementation of prompter $P$ $P$ concatenates $\mathbf{C}$ at the end of $\mathbf{p}$. If available, MEM-PROMPT can employ recent strategies on prompt-fine tuning (Zhao et al., 2021) to best combine $\mathbf{fb}$ with $\mathbf{p}$ e.g., deciding the position of $p$ or format of $C$’s output for best gains.

Although the model has not changed, adding $\mathbf{fb}$ corrects its erroneous behavior because we provide a few positive “training” examples containing feedback ($\mathbf{x}, \mathbf{fb} \rightarrow \mathbf{u}, \mathbf{y}$) in the prompt (Appendix B).
4 Experiments

Baselines We compare our system, MEM-PROMPT (memory-assisted prompt editing) with two different baselines:
- **NO-MEM** This is the standard GPT-3\(^4\) in few-shot prompting mode, with the suggested parameters (Appendix §A). Input is \( p \neq x_i \) (i.e., question \( x_i \) appended to prompt \( p \)). It generates answer \( y_i \) and its understanding of the user’s intent \( u_i \).
- **GROW-PROMPT**: Similar to NO-MEM, but the \( p \) is continuously grown with a subset of memory \( M \) that can fit within the prompt (max. 2048 tokens). The most recent subset of \( M \) 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:
- \( \text{Acc}(y) \): % of cases where answer matched the ground truth.
- \( \text{Acc}(u) \): % of cases where the model’s understanding of user’s intent is correct. As discussed in Section §3.2, depending on the task, the model generates its understanding on either the instruction or semantics of the question.

4.1 Main result: MEM-PROMPT improves GPT-3 accuracy

Does pairing GPT-3 with MEM-PROMPT improves performance? Section §4.1.1 empirically validates this question on ethical reasoning tasks and Section §4.1.2 on word reasoning tasks.

<table>
<thead>
<tr>
<th>model</th>
<th>ERT-CAT</th>
<th>ERT-NL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO-MEM</td>
<td>48.3</td>
<td>34.4</td>
</tr>
<tr>
<td>GROW-PROMPT</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MEM-PROMPT</td>
<td>60.0</td>
<td>38.5</td>
</tr>
</tbody>
</table>

Table 2: MEM-PROMPT outperforms NO-MEM (on 1000 test points) for both the categorical and the more challenging ERT-NL setup having longer, ambiguous inputs.

4.1.1 Ethical reasoning tasks

Table 2 presents results from running MEM-PROMPT on the DELPHI dataset (1,000 points in the test set). Recall from §3.3 that there are two kinds of feedback on DELPHI questions: CAT and

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

Figure 4: ERT-CAT: Label accuracy increases with time

Figure 5: ERT-CAT: Instruction accuracy sharply increases with a larger clarification probability. This shows that MEM-PROMPT responds to feedback.

NL feedback. MEM-PROMPT gets over 10% relative improvement in ERT-NL and over 25% relative improvement in ERT-CAT.

MEM-PROMPT effectively incorporates feedback, improving accuracy over time Figure 5 demonstrates that the instruction accuracy increases over time. We also see that if the user frequently provides clarifications then the model gets better more quickly. Figure 4 shows the label accuracy also improves over time. The baseline (NO-MEM) quickly saturates after 200 time steps, but MEM-PROMPT continues to improve. This is one of the advantages of engaging humans in the loop - the model keeps improving with time. It is also evident from these charts that as instruction accuracy improves, the label accuracy also improves (corr. coeff = 0.36).

Error analysis: Ethical-NL In both the ERT NL and CAT tasks, one of the primary source of label errors is confusion between labels such as OKAY and GOOD because of the nuanced differences e.g., input = teaching your child a musical instrument.
MEM-PROMPT predicted GOOD, while the expected answer was OKAY. Similar trends in this dataset were also observed by Jiang et al. (2021).

We randomly sampled from the ERT-NL test set where the model generates an incorrect understanding (i.e., $\text{Acc}(u) = 0$ 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 the performance of our model (as the task involves generation). \(\sim 30\% \ u\) is a lexical variation of the reference gold understanding. E.g., *telling a spouse your true feeling* vs. *loving your partner*. Notably, the generated label in these cases is still correct. (Example in 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 helpful, although not as effective as failure-driven retrieval (our model). However, GROW-PROMPT is \(\sim 3x\) more expensive (larger prompts) and cannot scale beyond the 2048 tokens limit. Our model MEM-PROMPT substantially outperforms both the baselines, showing the effectiveness of failure-driven reminding. 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.

- **Retrieval failures (18%)**: These errors are caused by an irrelevant retrieved understanding from the memory. A better retrieval function (e.g., one that models analogies between input situations) can potentially help alleviate these issues in the future (See Table 3, row 3)

Canonical examples of these error categories are shown in Table 3. We also find that over time, more relevant past examples are fetched (see Table 7).

<table>
<thead>
<tr>
<th>Error category</th>
<th>Input</th>
<th>Output (MEM-PROMPT)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct, lexically variant understanding</td>
<td>when I said &quot;I do&quot; to my wife, I am now realizing I only meant &quot;I'll try&quot;, does she deserve to know and a chance to back out?</td>
<td>This question is about: telling your true feelings.</td>
<td>This question is about: loving your partner.</td>
</tr>
<tr>
<td>Distracted understanding</td>
<td>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.</td>
<td>This question is about: wanting to kill yourself. The answer is BAD.</td>
<td>This question is about: seeking medical attention if you feel there is something wrong with you. The answer is GOOD.</td>
</tr>
<tr>
<td>Retrieval failures</td>
<td>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.</td>
<td>[retrieved feedback] This question is about: looking at someone’s social media. The answer is GOOD.</td>
<td>This question is about: judging people based on physical attributes. The answer is BAD.</td>
</tr>
</tbody>
</table>

Table 3: ERT NL task- error categories

4.1.2 Word Reasoning Tasks

For these tasks, we compare gold $u^*$ and generated $u$ based on some hard-coded linguistic variations (e.g., *the antonym is matches the opposite is*). Failure to generate $u$ is also considered incorrect. While we do not explicitly evaluate the accuracy of the task, we found a near-perfect correlation between the accuracy of $y$ and $u$ (i.e., if the GPT-3 understands the task correctly, the output was almost always correct).

Figure 6 reports the overall performance on the five lexical tasks overall. The accuracy improves substantially within 300 examples when using memory (in yellow) vs. no memory (in blue). Table 4 breaks down the performance by tasks. We note again that we are operating in a few-shot prompting regime (i.e., there is no training data over which we train). The fact that the model saturates within 300 examples shows that our method can continue to improve. 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 \(\sim 3x\) more expensive (larger prompts) and cannot scale beyond the 2048 tokens limit. Our model MEM-PROMPT substantially outperforms both the baselines, showing the effectiveness of failure-driven reminding. 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.

We also note that the performance gains achieved by MEM-PROMPT are less dramatic for word-level tasks. This is explained by the fact that task descriptions for the word scrambling tasks are less ambiguous (Section §3.3), preventing the model from getting confused by users’ instructions.
When the memory is used for every example (green line in Fig 6, top), the performance improves quickly as compared to the yellow line, where fb from memory is drawn with $P_r(f_b) = 0.5$.

### 4.3 MEM-PROMPT 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 MEM-PROMPT 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 find clear evidence that MEM-PROMPT is effective even with label feedback (see Appendix §C.3 for details).

### 4.4 Using MEM-PROMPT for language and dialects based personalization

We demonstrate an application of MEM-PROMPT 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 §D).

### 5 Conclusion

We have presented a simple, 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. Our implementation of system components are illustrative, not definitive; rather, the goal of this paper is to suggest a general architecture for future researchers, where more sophisticated component implementations can be designed. This architecture is significant as it suggests how deployed systems with fixed models can still be dynamically taught by interacting with end-users, potentially improving their performance and broadening their utility.
References


A Querying GPT-3-175B using OpenAI API

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

```python
import os
import openai
openai.api_key = os.getenv("OPEN_API_KEY")
response = openai.Completion.create(
    engine="davinci",
prompt="PROMPT",
temperature=0.7,
top_p=1,
frequency_penalty=0,
presence_penalty=0,
)
```

B Prompt

GPT3 is queried using a prompt p of example i/o behaviors, followed by the actual question x and (optionally) retrieved feedback fb. It then generates the understood intent u and answer y as a continuation. u and y are expressed a single sentence, e.g., "[The synonym for <word> is] [<word>]" Figure 7 shows this prompt p, containing a mixture of (x → u, y) and (x, fb → u, y) "training" tuples.

C 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).  

2. The homonyms (hom) were obtained using homz at https://github.com/camronehrlich/homz. We use the closest homonym returned by homz for each word in the English dictionary.

3. The definitions (defn) were sourced from The Online Plain Text English Dictionary at https://github.com/eddydn/DictionaryDatabase

4. Examples for usage in a sentence (sent) are from Commongen (Lin et al., 2020).

C.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 anonymized code repository https://anonymous.4open.science/r/memprompt-D548/templates.py.

C.2 Sample questions

Tables 8, 9, and 9 list some sample x-y for settings where the question was asked as a linguistic variation, in Hindi, and in Punjabi, respectively.

C.3 MEM-PROMPT 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 MEM-PROMPT 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
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 >
#
the word after removing symbols from s is straightforward 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 8: The prompt used for our word scrambling tasks derived from Brown et al. (2020)
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.

<table>
<thead>
<tr>
<th>Timestep</th>
<th>Question</th>
<th>Neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>is not wanting a dirty person harassing you morally ok?</td>
<td>is not wanting to hear foul language morally good?</td>
</tr>
<tr>
<td>70</td>
<td>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?</td>
<td>are you expected to pretend you’re straight when you’re gay?</td>
</tr>
<tr>
<td>85</td>
<td>Should I help someone when they are having trouble?</td>
<td>must you intervene if you see someone taking advantage of another disabled person?</td>
</tr>
<tr>
<td>230</td>
<td>needing a hug to feel good.</td>
<td>is wanting to cuddle with loved ones okay?</td>
</tr>
</tbody>
</table>

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.

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 9). 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 12 shows the top three of the $8 (k = 16/2)$ examples retrieved from $M$ for the alma mater cluster. 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 9: Instruction accuracy vs. time for WEBQA.

D 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 10 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., 2017) for trained memory, and suffix-trees for fast retrieval using surface form similarity.

![Finding 2 Large gains on queries asked in English and Punjabi by MEM-PROMPT.](image)

### E Sample results

Table 11 shows randomly sampled x-y pairs, and the corresponding y generated by GPT-3-175B and MEM-PROMPT. The complete set of outputs is located in the anonymized repository [https://anonymous.4open.science/r/memprompt-D548/results/results.csv](https://anonymous.4open.science/r/memprompt-D548/results/results.csv)
Factual question answering

Tables 12 and 13 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.
templates = [
    {
        "type": "syn",
        "template_id": "syn1",
        "question": lambda word1: f"What is similar to < {word1} > ?",
        "question_clarification": lambda word1: f"What is similar to < {word1} > ? | clarification: when I ask for similar to , I want a synonym.",
        "clarification": "clarification: when I ask for similar to , I want a synonym.",
        "answer": lambda word1, word2: f"the synonym for {word1} is {word2}"
    },
    {
        "type": "ant",
        "template_id": "ant0",
        "question": lambda word1: f"What is unlike < {word1} > ?",
        "question_clarification": lambda word1: f"What is unlike < {word1} > ? | clarification: when I ask for unlike , I want an antonym."
    },
    {
        "type": "defn",
        "template_id": "defn0",
        "question": lambda word: f"< {word} > means what ?",
        "question_clarification": lambda word: f"< {word} > means what ? | clarification: when I ask for means what , I want a definition."
    },
    {
        "type": "sent",
        "template_id": "sent1",
        "question": lambda word: f"< {word} > can be used how ?",
        "question_clarification": lambda word: f"< {word} > can be used how ? | clarification: when I ask for can be used how , I want a sentence."
    }
]

Listing 1: "Sample templates for the five tasks."

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

Table 8: Sample x-y pairs in English. The same type of question can be asked in multiple ways.
<table>
<thead>
<tr>
<th>Question (x)</th>
<th>Answer (y)</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; tabulate &gt; ka ulta kya hai ?</td>
<td>the antonym for tabulate is randomize</td>
<td>ant</td>
</tr>
<tr>
<td>&lt; foot &gt; ka vilom kya hai ?</td>
<td>the antonym for foot is head</td>
<td>ant</td>
</tr>
<tr>
<td>&lt; lettish &gt; ka matlab kya hota hai ?</td>
<td>the definition of lettish is The language spoken by the Letts. See Lettic.</td>
<td>defn</td>
</tr>
<tr>
<td>&lt; housing &gt; ka arth kya hai ?</td>
<td>the definition of housing is An appendage to the hames or collar of a harness.</td>
<td>defn</td>
</tr>
<tr>
<td>sunne mai &lt;perl&gt; jaisa kya hai ?</td>
<td>the homonym for perl is pearl</td>
<td>hom</td>
</tr>
<tr>
<td>&lt; council &gt; jaisa kya sunai deta hai?</td>
<td>the homonym for council is conceal</td>
<td>hom</td>
</tr>
<tr>
<td>&lt; city &gt; ko ek vakya mai kaise likhen ?</td>
<td>a sentence with city is: the city takes on an even more interesting hue during event</td>
<td>sent</td>
</tr>
<tr>
<td>&lt; fly &gt; ko ek vakya mai kaise likhen ?</td>
<td>a sentence with fly is: airplane fly into a storm cloud</td>
<td>sent</td>
</tr>
<tr>
<td>&lt; critique &gt; kai samaan kya hota hai ?</td>
<td>the synonym for critique is evaluate</td>
<td>syn</td>
</tr>
<tr>
<td>&lt; psychiatric &gt; kai samaan kya hota hai ?</td>
<td>the synonym for psychiatric is mental</td>
<td>syn</td>
</tr>
</tbody>
</table>

Table 9: Sample x-y pairs in Hindi. The same type of question can be asked in multiple ways.

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

Table 10: Sample x-y pairs in Punjabi. The same type of question can be asked in multiple ways.
<table>
<thead>
<tr>
<th>Question (x)</th>
<th>Expected answer (y *)</th>
<th>GPT-3 generated answer</th>
<th>MEM-PROMPT generated answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>give me something i would hear as &lt; whether &gt; ?</td>
<td>the homonym for whether is whither</td>
<td>the example of something you would hear as whether is a question word used to ask about possibilities.</td>
<td>the homonym for whether is weather</td>
</tr>
<tr>
<td>what has a &lt; mane &gt; like ring to it ?</td>
<td>the homonym for mane is made</td>
<td>the word that has a mane like ring to it is mane</td>
<td>the homonym for mane is main</td>
</tr>
<tr>
<td>What is like &lt; confine &gt; ?</td>
<td>the synonym for confine is hold</td>
<td>the homonym for confine is confined</td>
<td>the synonym for confine is restrain</td>
</tr>
<tr>
<td>&lt; leper &gt; means what ?</td>
<td>the definition of leper is a person affected with leprosy.</td>
<td>a sentence with leper is: a leper is a person with a disease that makes his skin scaly and white.</td>
<td>the definition of leper is a person who has leprosy.</td>
</tr>
<tr>
<td>expand on &lt; chaperon &gt; ?</td>
<td>the definition of chaperon is a matron who accompanies a young lady in public</td>
<td>the expansion of chaperon is one who chaperons.</td>
<td>the definition of chaperon is a woman who goes with a young girl to social functions.</td>
</tr>
<tr>
<td>what is a sentence that can be used to define &lt; wassail &gt; ?</td>
<td>the definition of wassail is an ancient expression of good wishes on a festive occasion</td>
<td>a sentence with wassail is: &lt; wassail &gt; wassails the christmas season</td>
<td>the definition of wassail is to drink to the health of.</td>
</tr>
<tr>
<td>expand on &lt; paralbumin &gt; ?</td>
<td>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</td>
<td>the expansion of paralbumin is an albuminoid globulin occurring in the blood of the horse and other animals, and in the urine of humans.</td>
<td>the definition of paralbumin is a particular kind of albumin.</td>
</tr>
<tr>
<td>what can one confuse with &lt; kew &gt; ?</td>
<td>the homonym for kew is key</td>
<td>a word that can be confused with kew is due</td>
<td>the homonym for kew is cue</td>
</tr>
<tr>
<td>What is like &lt; pneumatic &gt; ?</td>
<td>the synonym for pneumatic is windy</td>
<td>the homonym for pneumatic is hydraulic</td>
<td>the synonym for pneumatic is airy</td>
</tr>
<tr>
<td>give me something i would hear as &lt; seeder &gt; ?</td>
<td>the homonym for seeder is seeer</td>
<td>something you would hear as seeder is: someone who plants things in the ground or causes seeds to be planted.</td>
<td>the homonym for seeder is said</td>
</tr>
</tbody>
</table>

Table 11: A subset of random samples where GPT-3-175B without memory was incorrect. The complete set of outputs is located in the anonymized repository https://anonymous.4open.science/r/memprompt-D548/results/results.csv
<table>
<thead>
<tr>
<th>Domain</th>
<th>% Finished</th>
<th>Question</th>
<th>Neighbor 1</th>
<th>Neighbor 2</th>
<th>Neighbor 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alma mater</td>
<td>1</td>
<td>what high-school did harper lee go to?</td>
<td>what did st augustine do?</td>
<td>who is keyshia cole dad?</td>
<td>when did charles goodyear invented rubber?</td>
</tr>
<tr>
<td>Alma mater</td>
<td>5</td>
<td>what college did albert einstein go to?</td>
<td>what high-school did harper lee go to?</td>
<td>who did tim tebow play college football for?</td>
<td>what timezone is utah in?</td>
</tr>
<tr>
<td>Alma mater</td>
<td>10</td>
<td>what university did gordon brown attend?</td>
<td>what all does google now do?</td>
<td>what team did david beckham play for in 2011?</td>
<td>who did tim tebow play college football for?</td>
</tr>
<tr>
<td>Alma mater</td>
<td>40</td>
<td>where did john mayer go to college?</td>
<td>what school did michael jackson go to high school?</td>
<td>where did derek fisher go to college?</td>
<td>what style of music does john mayer play?</td>
</tr>
<tr>
<td>Alma mater</td>
<td>75</td>
<td>where did john steinbeck go to college?</td>
<td>where did john mayer go to college?</td>
<td>what college did john stockton go to?</td>
<td>where did otto frank go to college?</td>
</tr>
<tr>
<td>Alma mater</td>
<td>95</td>
<td>where did scott fitzgerald go to college?</td>
<td>where did scott fitzgerald go to college?</td>
<td>where did otto frank go to college?</td>
<td>where did derek fisher go to college?</td>
</tr>
<tr>
<td>Soccer</td>
<td>1</td>
<td>what team did david beckham play for before 2011?</td>
<td>who did tim tebow play college football for?</td>
<td>what super bowl did peyton manning win?</td>
<td>what type of music did john lennon sing?</td>
</tr>
<tr>
<td>Soccer</td>
<td>25</td>
<td>what team did ronaldo play for in 2003?</td>
<td>what part did winona ryder play in star trek?</td>
<td>what to do in richardson dalas?</td>
<td>who did the voice of darth vader in episode 3?</td>
</tr>
<tr>
<td>Soccer</td>
<td>33</td>
<td>who did nasri play for before arsenal?</td>
<td>what year did ray allen join the nba?</td>
<td>who does donnie wahlberg play in the sixth sense?</td>
<td>what does david beckham play?</td>
</tr>
<tr>
<td>Soccer</td>
<td>65</td>
<td>who has pudge rodriguez played for?</td>
<td>who does nolan ryan play for?</td>
<td>who did carlos boozer play for?</td>
<td>who does ronaldinho play for now 2011?</td>
</tr>
<tr>
<td>Soccer</td>
<td>99</td>
<td>what team did david beckham play for before la galaxy?</td>
<td>who does david beckham play for?</td>
<td>what does david beckham play?</td>
<td>what team does david beckham play for in 2012?</td>
</tr>
</tbody>
</table>

Table 12: Relevant examples
<table>
<thead>
<tr>
<th>Domain</th>
<th>% Finished</th>
<th>Question</th>
<th>Neighbor 1</th>
<th>Neighbor 2</th>
<th>Neighbor 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>1</td>
<td>what does jamaican people speak?</td>
<td>when was ancient egypt created?</td>
<td>where is the denver broncos stadium located?</td>
<td>what is the name of the capital of spain?</td>
</tr>
<tr>
<td>Language</td>
<td>20</td>
<td>what are the two official languages of paraguay?</td>
<td>what do portuguese people speak?</td>
<td>what language does cuba speak?</td>
<td>where is mission san buenaventura located?</td>
</tr>
<tr>
<td>Language</td>
<td>37</td>
<td>what language does colombia?</td>
<td>what language does cuba speak?</td>
<td>what was the first language spoken in spain?</td>
<td>what is serbian language called?</td>
</tr>
<tr>
<td>Language</td>
<td>85</td>
<td>what language does peru speak?</td>
<td>what are the official languages of the eu?</td>
<td>where is the latin language from?</td>
<td>what do portuguese people speak?</td>
</tr>
<tr>
<td>Language</td>
<td>90</td>
<td>what language do they speak in colombia south america?</td>
<td>how many languages do they speak in spain?</td>
<td>where is the latin language from?</td>
<td>what language does cuba speak?</td>
</tr>
</tbody>
</table>

Table 13: Relevant examples