### O.2 Experiments

The experiments involve several LLM calls with specific prompts to simulate the conversation, which we list below. We refer readers to the GitHub repository for how they are incorporated.

User simulator

```
You are simulating a user of an interactive LLM system (like ChatGPT).
The user is inherently lazy, and answers in short form, providing only minimal information to the
system. You should not be proactive.

Here's the conversation so far:
[[CONVERSATION_SO_FAR]]

Here are the shards that have already been revealed:
[[SHARDS_REVEALED]]

Here are all the shards that have not been revealed yet:
[[SHARDS_NOT_REVEALED]]

You must generate a response to the conversation so far. Here are the rules:
- [Providing a shard] You can reveal the content of a shard to the system in your response if it will
help the system move closer to answering the problem. You should select the shard to reveal that is
most "basic" and currently the most relevant.
- [One Shard at a Time] You should only reveal at most one shard at a time.
- [Reveal Entire Shard] If you reveal a shard, you must make sure to include *all the information in
the shard*. For example, if the shard is "your symptoms are that you have a headache in the
mornings", your response can't just be ``yeah I have headaches'', you must say ``yup mostly headaches
in the mornings``.
- [Irrelevant Clarifications] If the system asks you a question irrelevant to the shards, asks you a
generic question (``Can you give me a hint?``), you should respond with an answer that does not
provide a shard. (``I don't know``, ``Is that really important?``, etc.) You should not reveal any
information beyond what is available in the shards.
- [No Repeated Shards] You should not reveal the same shard more than once. Carefully review the
already revealed shards, and only reveal a shard if its `shard_id` is not on the list.
- [Rephrase Shards] If you reveal a shard, you should rephrase it in a conversational way. Do not
copy the shard verbatim.
- [Do Not Ask Questions] Your response should always be declarative sentences, and not questions.
- [Brevity of Response] You should favor being succint. Your answer can also have typos, improper
grammar, capitalization, etc. You are simulating a real person talking to an AI, who is in a hurry.
- [Format] Your response should be formatted as a JSON object with the following keys:
    - `response`: The response to the conversation so far.
    - `shard_id`: The shard you are revealing to the system. The shard_id can be an integer, or -1 if
    you did not reveal any shards.
For example:
{"response": "I don't know", "shard_id": -1}
or:
{"response": "yeah I want it to [...]", "shard_id": 1}
```

Response strategy categorization

```
You are reviewing a multi-turn conversation between a user and an assistant, and are given the last
turn of the conversation.

Here is the full specification of the problem the system is attempting to solve:
[[INITIAL_SHARD]]

Specification:
[[SHARDS]]

You must classify the response of the assistant according to the response type:
- `answer_attempt`: The response contains a complete answer attempt to the user's question (not
templated or hypothetical), that can be extracted verbatim. See the task-specific answer description
for more details.
- `clarification`: The response  is short (less than 100 words) and contains a single question
addressed to the user that directly inquires about an aspect of the user's query. A clarification
turn cannot be long (see `discussion`), cannot contain a vague question (see `discussion`) and cannot
contain multiple questions (see `interrogation`).
- `interrogation`: The response contains multiple questions addressed to the user, sometimes
organized in a list or bullet-points.
- `discussion`: The response discusses the question in detail, without providing a final answer,
asking a specific clarification question, or a refusal to answer. The response may or may not contain
a vague question (e.g., “What else can I help you with?”).
- `hedge`: The response contains multiple answer candidates based on hypotheticals (ifs) or branching
(case 1, case 2) with corresponding descriptions.
- `refuse`: The response contains an explicit or implicit refusal to answer the user's question
without a follow-up question or a request.
- `missing`: The response is empty/blank.

You must output your answer in the following JSON format:
{"response_type": "refuse|missing|answer_attempt|hedge|clarification|interrogation|discussion"}

Rules:
- The assistant giving a hint at how an answer could look like is not a final answer. You should only
select `answer_attempt` if the conversation could end at this stage with the user having an entirely
final answer to the problem they've formulated.
- [Task Specific Answer] [[ANSWER_DESCRIPTION]]

Conversation's last turn:
[[CONVERSATION_SO_FAR]]
```

Answer Extraction

```
You are reviewing a multi-turn conversation between a user and an assistant, and are given the last
turn of the conversation.
In the final response from the assistant, a final answer has been provided. Your goal is to extract
verbatim what the answer is:
- If the answer is short (less than 10 words), then you should copy verbatim what the answer is in
the `answer` field.
- If the answer is long, then you should produce the answer with an ellipses, to indicate the exact
start and end of the answer (e.g, ```def funny_function(n): [...]  return funny_output```). You
should include *at least* 4 words or one full line for the start (before the ellipses) and *at least*
4 words or one full line for the end (after the ellipses), such that the answer can be identified
exactly.

Rules:
- [Exact Answer Only] only extract the exact answer, and nothing else (including ``` for code blocks,
or intro/outro text).
- [Verbatim Only] Only extract verbatim text, do not modify the text in any way. If there's a typo,
an error, you must absoltutely include it, and not correct it in any way.
- [Task Specific Answer] [[ANSWER_DESCRIPTION]]
- [String output] the <answer_str> must be a string, not a number and not a dictionary.

You must output your answer in the following JSON format:
{"answer": "<answer_str>"}

Conversation's last turn:
[[CONVERSATION_SO_FAR]]
```