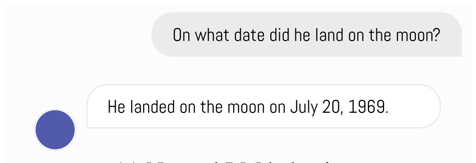
CLAM: Selective Clarification for Ambiguous Questions with Generative Language Models

Lorenz Kuhn [*] [1] **Yarin Gal** [1] **Sebastian Farquhar** [1]

Abstract

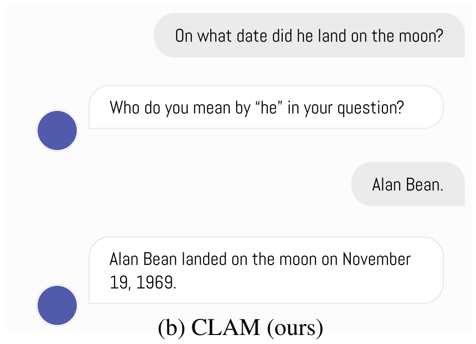
Users often ask dialogue systems ambiguous questions that require clarification. We show that current language models rarely ask users to clarify ambiguous questions and instead provide incorrect answers. To address this, we introduce CLAM: a framework for getting language models to selectively ask for clarification about ambiguous user questions. In particular, we show that we can prompt language models to detect whether a given question is ambiguous, generate an appropriate clarifying question to ask the user, and give a final answer after receiving clarification. We also show that we can simulate users by providing language models with privileged information. This lets us automatically evaluate multi-turn clarification dialogues. Finally, CLAM significantly improves language models' accuracy on mixed ambiguous and unambiguous questions relative to SotA.



On what date did he land on the moon?

He landed on the moon on July 20, 1969.

(a) Normal LM behaviour.



On what date did he land on the moon?

Who do you mean by "he" in your question?

Alan Bean.

Alan Bean landed on the moon on November 19, 1969.

(b) CLAM (ours)

Figure 1. (a) Normally, LMs answer one of many interpretations given an ambiguous question. (b) Our method uses few-shot classification to detect ambiguous questions and selectively asks for clarifying information needed to answer the question.

1. Introduction

Recent Transformer-based large language models (LLMs) are often accurate on open- and closed-book question-answering tasks (Chung et al., 2022; Hoffmann et al., 2022). These data sets typically consist of well-defined questions with enough information to have a unique answer. As these language models are deployed, however, they will often face *vague* user questions. A user will have some well-defined question in mind but accidentally pass an under-specified question to the question-answering model. For example, a user might want to ask "On what date did Alan Bean land on the moon?" but accidentally pass the question "When did he land on the moon?" to the language model, as illustrated in Figure 1. The fact that user requests are often ambiguous is well-established in the information retrieval literature (see Keyvan & Huang (2022) for an overview) but has so far received little attention in the LLM question-answering community. This is despite widespread reports of models "hallucinating" responses when faced with unanswerable questions (see Ji et al. (2022) for an overview). In this paper, we show that state-of-the-art language models rarely ask for clarification about ambiguous user inputs and thus perform poorly when answering ambiguous questions.

To address this issue, we introduce CLAM, a framework that can significantly improve language models' question-answering performance in a setting we describe as **selective clarification question answering**. The framework involves: identifying ambiguous questions, prompting the model to resolve ambiguity, and answering the disambiguated question. In this paper, we demonstrate some of the tools that can be used to implement this procedure, showing that even a relatively simple implementation can greatly improve performance. We also show that this method reliably only asks for clarification when the user input is actually ambiguous

[*]Equal contribution [1]OATML, Department of Computer Science, University of Oxford. Correspondence to: Lorenz Kuhn <lorenz.kuhn@cs.ox.ac.uk>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

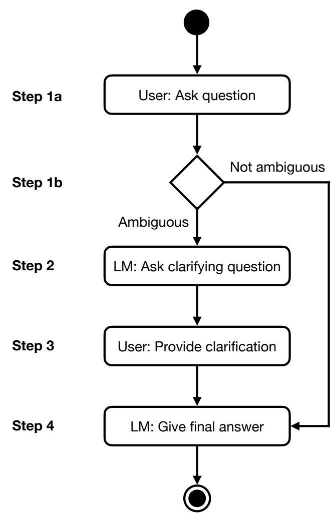


Figure 2. **Overview of Selective Clarification**

and thus avoids asking unnecessary clarifying questions.

Conceptually, CLAM can be seen as a form of **meta-cognition**—often described as thinking-about-thinking (Lai, 2011). That is, CLAM is a way of improving model performance by prompting the model to explicitly "reason" about a property of the given problem before trying to solve it. Our method provides a proof of concept for the use of meta-cognition for language models as a strategy for more reliable and safer deployment. We introduce "meta-cognition" as a term of art for machine learning systems design because we believe it will be an important component of future foundation model-based research.

Selective clarification QA involves interactive multi-turn dialogue. Evaluating multi-turn dialogue with human participants is expensive, hard to reproduce, and can require ethics-board approval. At the same time, automatic evaluations for multi-turn dialogue are often unreliable (Liu et al., 2016; Wahde & Virgolin, 2022). In order to allow scalable model evaluation we describe an automatic prompt-driven evaluation protocol that allows a language model that is given privileged information to stand in for a person during evaluation. In this way, we propose that language model research should shift towards **evaluation data-generating processes** rather than evaluation data sets. See Appendix A for a discussion of related work.

In summary, our contributions are:

- We introduce the **CLAM framework** for detecting ambiguous questions and clarifying them through multi-turn dialogue in LLMs (Section 2).
- We introduce the concept of language model **meta-cognition** as a general category of which CLAM is a proof-of-concept (Section C).
- We desribe an automatic evaluation protocol for **selective clarification QA** (Section 3).
- We show that our implementation of CLAM delivers large accuracy improvements on data sets that contain ambiguous questions (Section 4).

2. CLAM: Selective clarification Framework

In this section, we introduce **CLarify-if-AMbiguous (CLAM)**—a framework for language models to ask for selective clarification about possibly vague user questions.

The framework involves four stages, illustrated in Figure 2. In the first stage, the user asks a language model a question. The question is then classified as `ambiguous` or `not ambiguous`. For questions that are not ambiguous, we return the answer immediately. However, when questions are ambiguous, we generate a disambiguating follow-up question, as illustrated in Figure 3. The model then uses the entire dialogue, including clarifying information, to answer the original question as it was intended. Although we complete only a single iteration, it is also possible to recur the clarification process until the entire dialogue is considered to unambiguously ask a precise question which is an interesting direction for future research. We describe this formally in Algorithm 1.

Implementing the CLAM framework requires choosing a specific technique for: 1) classifying questions as ambiguous or not ambiguous and 2) producing a clarifying question to follow up with.

In this paper, we implement both of these steps using prompting. To classify a question's ambiguity, we prompt the model using few-shot prompts consisting of examples of ambiguous and unambiguous questions from the given data set. See Appendix G for the prompts used in our experiments.

We then take the model's log probability of the next token being `True` as a continuous predictor of whether the given question is ambiguous or not.

Interestingly, the fact that this works means that SotA models are able to detect ambiguous questions but do normally not ask the user for clarification. We conjecture that this is the case because there are few dialogues including clarifying questions in the pre-training or finetuning data sets of these models. We leave a thorough investigation of why current models do not ask for clarification for future work.

We then produce a clarifying question using another prompt. We simply append the string: "In order to answer this question, I have to ask the following clarifying question:" to the original ambiguous question.

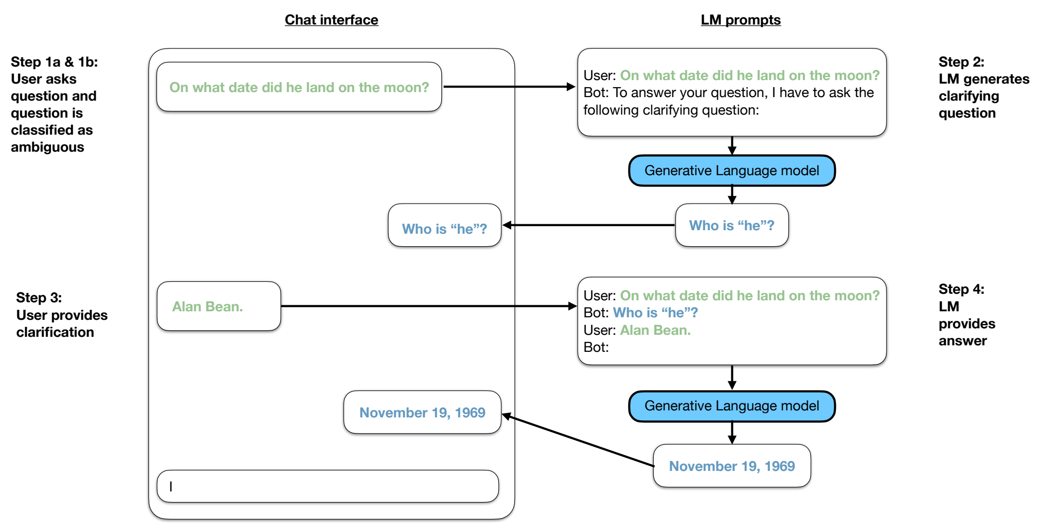

Chat interface
LM prompts
Step 1a & 1b: User asks question and question is classified as ambiguous
On what date did he land on the moon?
User: On what date did he land on the moon?
Bot: To answer your question, I have to ask the following clarifying question:
Step 2: LM generates clarifying question
Generative Language model
Who is "he"?
Who is "he"?
Step 3: User provides clarification
Alan Bean.
User: On what date did he land on the moon?
Bot: Who is "he"?
User: Alan Bean.
Bot:
Step 4: LM provides answer
Generative Language model
November 19, 1969
November 19, 1969

Figure 3. **Overview of the prompts used to clarify ambiguous user inputs**. In step 1a, the user asks an ambiguous question. In step 1b (omitted for clarity) the question is classified as ambiguous using a few-shot prompt. In step 2, the model is few-shot prompted to generate a clarifying question about the ambiguous user input. In step 3, the user (or an oracle model, see Section 3) provides clarifying information given the clarifying question. In step 4, the model is prompted to answer the initial question given the clarification from the user.

We use a zero-shot prompt on the Ambiguous TriviaQA data set, and a few-shot prompt on the more challenging CLAQUA data sets. We describe the data set specific prompts in Appendix G.

We then present the user with the clarifying question that is generated by the model based on this newly constructed prompt. Lastly, we present the combination of ambiguous question, clarifying question and the user's clarification to the language model to generate a final answer to the original question.

3. Automatic Evaluation Protocol

In this section, we introduce an automatic evaluation protocol that allows us to evaluate multi-turn dialogues without requiring human input. In the selective clarification QA setting (Figure 2), the user provides clarification when the model asks a clarifying question about an ambiguous user input. We show that we can automate this step of providing clarification using a language model that is given privileged information about the ambiguous question. On a high-level, we thus suggest that model evaluation should move towards *evaluation data-generating processes* rather than evaluation data sets.

The automated evaluation of dialogue systems is attractive given that human evaluations have numerous problems. They are too expensive for most researchers to access. They cannot be easily reproduced and are idiosyncratic in ways that are hard for external researchers to observe and critique (unlike automatic evaluations, whose flaws are relatively easy to examine). Lastly, in many cases experiments involve humans can create additional ethics risks that in even the best cases incur additional administrative and ethical approvals costs, which can reduce research iteration speed, and in the worst cases create hazards for research participants.

Instead, we therefore use a language model to provide clarifying information when asked. This then allows us to automatically evaluate performance on the selective clarification task. Using machine learning models to simulate users to evaluate dialogue systems has been suggested before (see e.g. Su et al. (2016)). To the best of our knowledge, our work is the first to suggest that a parallel corpus of unambiguous and ambiguous questions can be used to prompt large language models to provide clarifying information about ambiguous questions.

For selective clarification QA, the language model may ask the user for clarifying information about the user's initial question (see Figure 3). Instead of a human, in our protocol, an 'oracle' language model which has access to privileged information about the unambiguous question provides the clarifying information when asked (see Figure 4). Since our data set contains both ambiguous and corresponding unambiguous questions, we provide the oracle model with privileged information by including the unambiguous question in its prompt. When appropriately asked for clarification, the oracle can then reliably provide clarifying information based on the unambiguous question in its prompt.

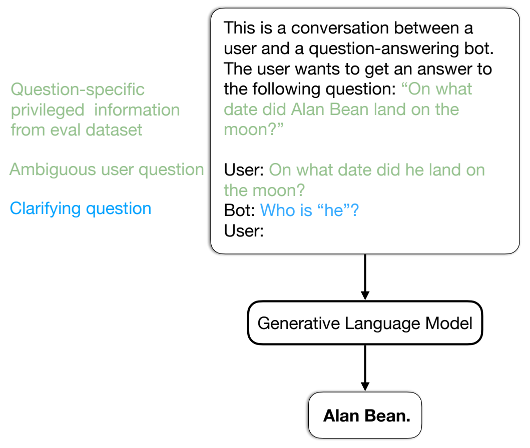

Question-specific privileged information from eval dataset
Ambiguous user question
Clarifying question
This is a conversation between a user and a question-answering bot. The user wants to get an answer to the following question: "On what date did Alan Bean land on the moon?"
User: On what date did he land on the moon?
Bot: Who is "he"?
User:
Generative Language Model
Alan Bean.

Figure 4. **Using a language model to provide clarification.** We prompt a language model to provide clarifying information given a clarifying question about an ambiguous user input. Our parallel corpus of ambiguous and corresponding unambiguous questions allows us to provide the unambiguous question to the oracle LM, based on which it can then provide appropriate clarifying information about the ambiguous question. See Figure 3 for a description of the other conversational turns.

4. Experiments

In this section, we validate CLAM's ability to detect and seek clarification for ambiguous questions through experiments. We first demonstrate its improved accuracy on question answering using a dataset containing both ambiguous and unambiguous questions. Next, we evaluate CLAM's performance on each step of the pipeline, including distinguishing ambiguous questions, generating clarifying questions, and providing correct final answers given clarifying information. Lastly, we assess the reliability of our automatic evaluation setup by testing whether the oracle language model provides the correct clarification when presented with a clarifying question.

Our framework applies to any large language model architecture, and we use the `text-davinci-002` and `davinci` models via the OpenAI API in our experiments. We evaluate CLAM on a range of datasets covering different types of ambiguity, including our own Ambiguous TriviaQA (see Appendix B), ClariQ (Aliannejadi et al., 2020), and CLAQUA (Xu et al., 2019). However, only our dataset can be used to evaluate all steps of our pipeline end-to-end, while the other datasets can only be used for a subset of the steps. We measure the accuracy of model answers by evaluating whether they contain the reference answer and introduce an adjusted accuracy metric that penalizes the language model system for asking unnecessary clarifying questions about unambiguous questions. See Appendix D for further experimental details.

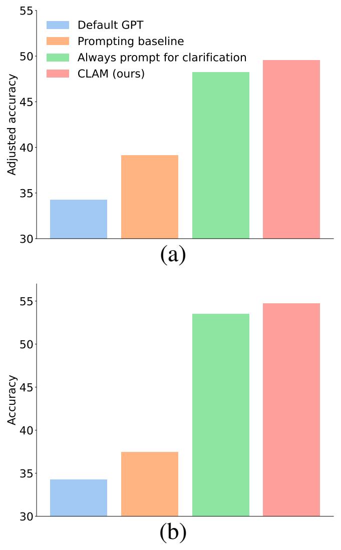


Figure 5. **CLAM improves question-answering accuracy on a set of ambiguous and unambiguous Trivia questions**. (a) CLAM clarifies ambiguous questions without asking for unnecessary clarification on unambiguous questions (which is reflected in the adjusted accuracy metric). Always prompting the language model to ask the user for clarification increases the accuracy on ambiguous questions but incurs a penalty on unambiguous questions. The prompting baseline rarely asks the user for clarification and thus only improves the accuracy slightly. (b) **Accuracy on full data set:** Without penalizing unnecessary clarifying questions, always prompting for clarification and CLAM perform comparably well, and much better than default GPT and the prompting baseline. CLAQUA and ClariQ only cover parts of the selective clarification pipeline which is why only TriviaQA results are reported here.

We compare CLAM to three baselines, each with different approaches to handling ambiguous questions: **Default GPT** uses the standard GPT model without addressing ambiguity or seeking clarification. **Prompting baseline** modifies the prompt to encourage the model to selectively ask for clarification when faced with ambiguous questions. **Force clarification** always prompts the model to ask for clarification, regardless of whether the question is ambiguous or not.

We introduce an automatic evaluation protocol for multi-turn dialogues using a language model to provide clarifying information in the selective clarification QA setting, reducing reliance on human evaluations and their associated challenges; see Section 3 for further details.

4.1. Results

We first establish the overall accuracy improvement of our implementation of the CLAM framework on selective clarification QA (steps 1 through 4 in Figure 2). Then, we study the performance on each of the necessary steps in Figure 2:

detecting ambiguous questions (step 1b), generating useful clarifying questions (step 2), and giving final answers after receiving clarification (step 4). Lastly, we show that our automatic evaluation scheme for selective clarification QA works reliably, by showing that the oracle language model correctly provides clarification when asked (step 3).

We report the results of the `text-davinci-002` model in the main part of the model and provide the results for `davinci` in the Appendix. `davinci` performs less well across all tasks but the differences between the different methods are qualitatively similar to those of the `text-davinci-002` model.

ClariQ and CLAQUA each only contain components to evaluate some of the parts of the pipeline (see Table 4), and some figures thus report results on a subset of the data sets.

Overall performance

Overall, we find that CLAM boosts the language model's adjusted accuracy on ambiguous TriviaQA (containing both ambiguous and unambiguous questions) by roughly 20 percentage points (Figure 5a). Recall that the adjusted accuracy multiplies the accuracy on a given question with a penalty term $\lambda = 0.8$ each time the model unnecessarily asks for clarification on unambiguous questions as described in Section 4. *Force clarification* improves the accuracy on ambiguous questions but incurs a large penalty on unambiguous questions for unnecessarily asking for clarification. Using the *Prompting baseline* only leads to a moderate accuracy improvement. Without the penalty term for asking unnecessary clarifying questions, the performance of always prompting for clarification and *CLAM* on the data set of both ambiguous and unambiguous questions is comparable, and they both clearly outperform default GPT and the prompting baseline (Figure 5b).

On the ambiguous questions only, we find that, as expected, both *Force clarification* and *CLAM* greatly improve the question-answering accuracy as compared to the default GPT performance, see appendix Figure 10a. Importantly, always clarifying and CLAM almost entirely close the gap on performance on ambiguous questions as compared to the performance on unambiguous questions, see appendix Figure 10b. On the unambiguous questions, the different methods generally do not affect the model performance as compared to the default GPT behaviour. However, always prompting for clarification always leads to unnecessary turns of conversation on unambiguous questions which is bad for the user experience, and sometimes actually hurts the accuracy by leading the conversation off-topic.

We additionally evaluate the performance on each of the individual pipeline steps in Appendix F, and show that the models can reliably identify ambiguous questions, ask appropriate clarifying questions and provide final answers. We also show that our automated evaluation protocol reliably simulates human users.

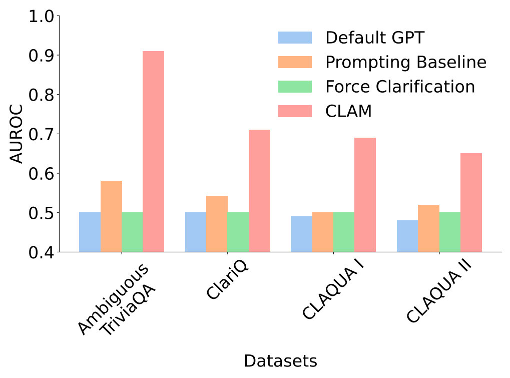


Figure 6. **CLAM reliably distinguishes ambiguous from unambiguous questions**. The AUROC measures how accurately the different methods predict whether a given question is ambiguous or not. *Default GPT* almost never asks for clarification whereas *Force clarification* always asks for clarification. Both methods thus do not distinguish ambiguous from unambiguous questions.

5. Conclusion

In this paper, we introduce CLAM, a framework for selective clarification QA with large language models which detects and resolves ambiguity by asking clarifying questions. We provide this as an example of meta-cognition in foundation models. We implement the framework using few-shot prompting. This additional clarifying conversational turn significantly increases language models' question-answering accuracy on ambiguous questions without affecting unambiguous questions. Moreover, we show that few-shot prompting is a highly reliable way of detecting whether a given question is ambiguous which allows us to answer clarifying questions about ambiguous questions only and avoid asking unnecessary questions about precise user inputs.

In order to support scalable research, we motivate a shift towards evaluation data-generating processes and introduce a method to automatically evaluate multi-turn dialogues involving ambiguous questions using an oracle language model with access to extra information.

Acknowledgements

Sebastian Farquhar carried out this work in his capacity as an associate member of the OATML lab, but he is also employed by DeepMind.

We are grateful to Geoffrey Irving and Laura Rimell for their advice and feedback on earlier drafts of this paper. We are also grateful to the members of the OATML group for helpful discussions about this project.

References

Aliannejadi, M., Kiseleva, J., Chuklin, A., Dalton, J., and Burtsev, M. Convai3: Generating clarifying questions for open-domain dialogue systems (clariq). *arXiv preprint arXiv:2009.11352*, 2020.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Clarke, C. L., Craswell, N., and Voorhees, E. M. Overview of the trec 2012 web track. Technical report, NATIONAL INST OF STANDARDS AND TECHNOLOGY GAITHERSBURG MD, 2012.

Dhole, K. D. Resolving intent ambiguities by retrieving discriminative clarifying questions. *arXiv preprint arXiv:2008.07559*, 2020.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2022.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Keyvan, K. and Huang, J. X. How to approach ambiguous queries in conversational search? a survey of techniques, approaches, tools and challenges. *ACM Computing Surveys (CSUR)*, 2022.

Krasheninnikov, D., Krasheninnikov, E., and Krueger, D. Assistance with large language models. In *NeurIPS ML Safety Workshop*, 2022. URL `https://openreview.net/forum?id=OE9V81spp6B`.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *NeurIPS ML Safety Workshop*, 2022.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K. N., Jones, L., Chang, M.-W., Dai, A., Uszkoreit, J., Le, Q., and Petrov, S. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

Lai, E. R. Metacognition: A literature review research report. *Research Reports*, 41, 2011.

Liu, C.-W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., and Pineau, J. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

Min, S., Michael, J., Hajishirzi, H., and Zettlemoyer, L. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*, 2020.

Rao, S. and Daumé III, H. Answer-based adversarial training for generating clarification questions. *arXiv preprint arXiv:1904.02281*, 2019.

Su, P.-H., Gasic, M., Mrksic, N., Rojas-Barahona, L., Ultes, S., Vandyke, D., Wen, T.-H., and Young, S. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689*, 2016.

Trienes, J. and Balog, K. Identifying unclear questions in community question answering websites. In *European conference on information retrieval*, pp. 276–289. Springer, 2019.

Wahde, M. and Virgolin, M. Conversational agents: Theory and applications. *arXiv preprint arXiv:2202.03164*, 2022.

Wang, J. and Li, W. Template-guided clarifying question generation for web search clarification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3468–3472, 2021.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Xu, J., Wang, Y., Tang, D., Duan, N., Yang, P., Zeng, Q., Zhou, M., and Sun, X. Asking clarification questions in knowledge-based question answering. 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)*, pp. 1618–1629, 2019.

A. Related work

A range of approaches for i) detecting ambiguous questions and then ii) asking clarifying questions have been proposed in information retrieval and in the conversational search literature (see Keyvan & Huang (2022) for an overview). In terms of *detecting* ambiguous queries, Trienes & Balog (2019) use a logistic regression to identify ambiguous queries based on the characteristics of similar queries. Dhole (2020) use a BiLSTM model to distinguish ambiguous from unambiguous queries. In terms of *generating clarifying questions*, both rule-based and neural network-based approaches have been proposed. Wang & Li (2021), for instance, use a clarifying question template that is completed with words from a vocabulary. Dhole (2020) frame disambiguation as distinguishing between different plausible user intents for a given question. They use a set of syntactic transformations of a given ambiguous question, and then select a clarifying question that will best disambiguate between different user intents. Rao & Daumé III (2019) use a GAN to generate clarifying questions.

AmbigQA (Min et al., 2020) is an alternative data set intended to explore ambiguous question answering derived from Natural Questions (Kwiatkowski et al., 2019). However, Min et al. (2020) note that the ambiguity in their data set is "sometimes subtle" and that "many [ambiguities] are only apparent after examining one or more Wikipedia pages". Furthermore, Krasheninnikov et al. (2022) then find that the performance of AmbigNQ, a baseline they introduce to find various precise questions for a given ambiguous question, is low, attesting to the difficulty of this task. Given how difficult AmbigNQ seems to be, we suggest that our Ambiguous TriviaQA data set, which requires less factual knowledge, is a more effective evaluation tool for selective clarification QA.

In transformer-based dialogue systems, however, dealing with ambiguous queries has received little attention so far. To the best of our knowledge, Krasheninnikov et al. (2022) (concurrent with our work), is the only paper that addresses ambiguous question resolution in GPT-like language models. The authors fine-tune a 175B parameter GPT-3 model on a data set of conversations consisting of ambiguous user requests, clarifying questions, and final answers. They show that fine-tuning the model on this data set leads to a slight accuracy improvement in answering ambiguous questions derived from AmbigQA (Min et al., 2020). The authors note that under this approach the model often does not recognize ambiguous inputs (false omission rate of 44.5%). We further note that in contrast to this method our approach does not require any fine-tuning, neither to improve the performance of the question-answering nor for the oracle model.

B. Selective Clarification QA Data Set

In this section, we introduce a data set that allows us to evaluate the performance on *selective clarification QA* (Figure 2). Selective clarification QA models the real-world observation that some user questions will be well-defined and thus will not need clarification, while other user questions will be ambiguous, and require clarification. The desired behaviour of a language model is to directly provide answers to unambiguous questions without asking for unnecessary clarification and on the other hand, ask the user for clarification if the user's question is ambiguous.

Existing data sets such as ClariQ (Aliannejadi et al., 2020) and CLAQUA (Xu et al., 2019) do not allow us to evaluate models on all the steps of the selective clarification QA setting, see Table 4. We describe existing data sets and their limitations in Section 4.

To study this setting, we therefore introduce a data set of pairs of questions that we call **Ambiguous TriviaQA**. For each pair, there is one ambiguous question and one precisely disambiguated question. We construct the data set so that just one piece of clarifying information is needed to make an ambiguous question precise. In Section 3, we explain how these sets of ambiguous and unambiguous questions let us automatically provide clarify ambiguous questions.

Our data set consists of 200 pairs of ambiguous and unambiguous questions that we derive from TriviaQA (Joshi et al., 2017). Given a randomly sampled TriviaQA question, we derive an ambiguous question by either:

- Replacing a name or noun with a generic pronoun, e.g. "Where in England was Dame Judi Dench born?" becomes "Where in England was she born?".
- Replacing a noun phrase with a class the noun belongs to, e.g. "Which country is Europe's largest silk producer?" becomes "Which country is Europe's largest producer?"

We use closed-book TriviaQA questions, that is, questions that stand alone and for which no accompanying context is provided. An additional advantage of deriving a data set from TriviaQA is that the reference answers are typically short, and only contain few non-essential words both of which increase the reliability of automatic accuracy metrics to evaluate models.

C. Meta-cognition

In humans, meta-cognition is the process of thinking about your own thought process. Working with foundation models, meta-cognition can involve using information that we have about the default sequence completion task to choose a new sequence completion task to perform instead.

In this paper, we use the example of prompting the model to detect whether a given question is ambiguous to then ask the user a clarifying question, rather than directly trying to answer the ambiguous question. But in broader contexts, one can imagine many useful pipelines in which a secondary classifier is used to 'redirect' the 'thought process' of a foundation model. For example, a classifier detecting some form of toxicity might be able to redirect the question to a more constructive frame by selecting an alternative prompt, allowing it to answer the question in a less toxic way. Using this sort of pipeline, predictable failure modes can be gracefully and easily recovered from without resulting in any errors that are visible to the user. Chain-of-thought prompting (Wei et al., 2022) can be thought of as an implicit form of meta-cognition, while a more sophisticated pipeline might use additional information about the problem to shape the prompt construction.

D. Experimental details

In this section, we describe the experimental setup that we use to validate the ability of CLAM to successfully detect and seek clarification to ambiguous questions. See Section 4 for the main results of our experiments.

We first demonstrate CLAM's overall improved accuracy on the end-to-end task of question answering on a data set that contains both ambiguous and unambiguous questions.

We then evaluate the performance on CLAM on each step of the pipeline (Figure 2) that involves the language model individually: step 1b, distinguishing ambiguous from unambiguous questions; step 2, generating appropriate clarifying questions; and step 4, giving correct final answers given clarifying information.

Lastly, we evaluate whether our automatic evaluation setup works reliably. That is, we test whether the oracle language model reliably provides the correct clarification when presented with a clarifying question.

Models: In principle, our framework applies to any large language model architecture. In our experiments, we use the `text-davinci-002` and `davinci` models via the OpenAI API. `davinci` is a pre-trained language model similar to the original GPT-3 model (Brown et al., 2020) and `text-davinci-002` [1] is a GPT-3 model with an additional supervised fine-tuning stage on human-written demonstrations. These models are publicly available to researchers from any institution, which improves the reproducibility of our results and evaluation protocol. It can, however, be difficult to confirm whether the currently actively served version of the model is the same as the one used in these experiments, which is a challenge for reproducibility.

Data sets: There are many different types of ambiguity in text (see (Keyvan & Huang, 2022) for a taxonomy). To account for this, we evaluate CLAM on a range of data sets that cover different types of ambiguity. Our own data set *Ambiguous TriviaQA* (described in Appendix B) primarily covers under-specified user questions. *ClariQ* (Aliannejadi et al., 2020) is an information retrieval data set consisting of real search engine queries from the TREC Web Track (Clarke et al., 2012). For each query human labellers also indicate on a scale of 1 to 4 to what extent clarification is needed. We convert ClariQ into a binary classification problem by labelling queries with *clarification needs* of 1 and 2 as `unambiguous` and queries with *clarification needs* of 3 and 4 as `ambiguous`. Furthermore, we evaluate CLAM on *CLAQUA* (Xu et al., 2019), a data set consisting of two sub-datasets that cover different types of ambiguity: a task that we call *CLAQUA I*, in which a proper noun in the question can refer to different entities, and a task that we call *CLAQUA II* which consists of multi-turn conversation where the last turn is a question that could refer to multiple different previous conversational turns. To reduce the costs of our experiments from calling the OpenAI API, we use random sub-samples of 400 of each data set when evaluating ambiguity detection, and 100 random samples from each data set to evaluate the generation of clarifying questions and question answering accuracy.

However, only our data set can be used to evaluate all of the steps of our pipeline end-to-end. Because of limitations in the design of the other data sets, which were designed for different purposes, they can only be used for a subset of the steps in

[1] https://beta.openai.com/docs/model-index-for-researchers

Table 1. The `davinci` model reliably generates the correct clarifying question and clarification

	Ambiguous TriviaQA	CLAQUA I	CLAQUA II
Correct clarifying question	97%	99%	92%
Correct oracle answer	98%	76%	67%

our pipeline. Table 4 provides an overview of which data set can be used to test which of the steps.

Metrics: We measure the accuracy of a model answer by evaluating whether the model answer contains the reference answer. This accounts for the fact that the language model often answers in full sentences while the reference answer consists only of the target terms themselves.

One challenge in the automatic evaluation of free-form generations is that one answer can be expressed in many different ways, see e.g. Kuhn et al. (2022). To account for this, we manually evaluate the accuracy metric on all of our data sets. Similarly, we manually evaluate whether the language models ask appropriate clarifying questions.

In addition to the raw accuracy, we introduce an *adjusted accuracy* which is suitable for selective clarification QA specifically. This measure penalizes the language model system for asking unnecessary clarifying questions about unambiguous questions. To adjust the accuracies we begin with a score of 1 for a correct answer and 0 for an incorrect answer (as normal) but then multiply it with 0.8 if the question is unambiguous and the model nonetheless asks for clarification. The specific value of 0.8 is arbitrary and our results hold for a range of penalty terms, see Table 2.

To evaluate how well the different methods can distinguish ambiguous from unambiguous questions, we measure the commonly used Area Under the Receiver Operator Characteristic Curve (AUROC). The AUROC metric is equivalent to the probability that a randomly chosen correct answer has a predictor score than a randomly chosen incorrect answer. Higher scores are better, with perfect uncertainty scoring 1 while a random uncertainty measure would score 0.5.

Baselines: We compare CLAM to three baselines. *Default GPT* prepends the question with a question-answering prompt: `This is a conversation between a user and a question-answering bot.`

Prompting baseline uses a prompt that explicitly instructs the model to selectively ask the user for clarifying questions: `This is a conversation between a user and a question-answering bot. The bot asks the user for clarification if the user's question is ambiguous or imprecise.`

Force clarification does not distinguish between ambiguous and unambiguous questions, and always prompts the language model to ask for clarification about the user's input.

Automatic Evaluation Protocol

We introduce an automatic evaluation protocol for multi-turn dialogues using a language model to provide clarifying information in the selective clarification QA setting, reducing reliance on human evaluations and their associated challenges; see Appendix for further details.

E. Additional experimental results

E.1. Results on `davinci` model

In this section, we report the results of our experiments using the `davinci` model whereas we used the `text-davinci-002` model in the main part of the paper. Overall, we observe qualitatively similar results using the `davinci` model as with the `text-davinci-002`, although at a lower level. In summary, we find that

- We find that `davinci` is also able to distinguish ambiguous from unambiguous questions, see Figure 7, although less reliably than `text-davinci-002`.
- `davinci` is able to reliably generate clarifying questions across all data sets, and relatively reliably provides clarification, see Table 1.
- Lastly, we show that clarifying improves QA accuracy on ambiguous questions across all data sets Figure 8.

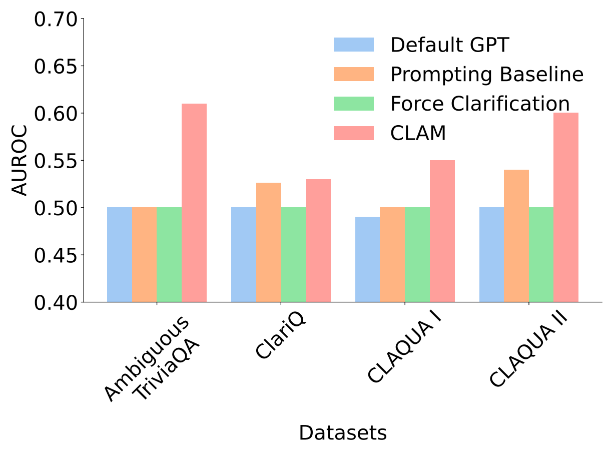


Figure 7. Using the `davinci` model for CLAM, we are able to detect ambiguous questions more reliably than *Default GPT*, *Force clarification* and *Prompting baseline*. `davinci` detects ambiguous questions less well than the`text-davinci-002` model used in the main part of the paper.

Table 2. **Adjusted accuracy results for different penalty terms on the full Ambiguous TriviaQA data set.** In our adjusted accuracy metric, the accuracy of the subset of questions that are *unambiguous* and on which the language model nonetheless asks for clarification are multiplied with a penalty term $0 < \lambda < 1$. We show that CLAM outperforms the default GPT model and the baselines regardless of the particular choice of λ. Note that default GPT and the prompting baseline never ask for clarification on unambiguous questions and thus do not incur a penalty.

Method / λ	0.5	0.6	0.7	0.8	0.9	1.0
Default GPT	34.25	34.25	34.25	34.25	34.25	34.25
Prompting baseline	37.50	37.50	37.50	37.50	37.50	37.50
Always prompt for clarification	40.37	43.0	45.62	48.25	50.88	53.5
CLAM (ours)	53.88	54.05	54.22	54.40	54.58	54.75

E.2. Ablation of penalty term in adjusted accuracy

In Table 2, we show that our adjusted accuracy results hold up across different penalty terms λ. The results in the main part of the paper use $\lambda = 0.8$

F. Evaluation of individual pipeline components

In addition to the overall results presented in Section 4, we investigate how well CLAM performs on each of the steps of the pipleine.

Step 1b: Detect Ambiguity

CLAM reliably distinguishes ambiguous from unambiguous questions, see Figure 6. The few-shot prompting-based log probability of a given question being ambiguous or unambiguous achieves high AUROCs on all data sets. *Default GPT* almost never asks for clarification and thus achieves an AUROC of roughly 0.5 on all data sets. *Force clarification* on the other hand treats all questions as ambiguous questions, and thus always has an AUROC of 0.5. The *Prompting baseline* does sometimes asks for clarification on ambiguous questions and almost never asks for clarification on unambiguous questions, and is thus slightly better than random at detecting ambiguity.

Step 2: Ask clarifying questions

We manually label 100 randomly selected pairs of ambiguous questions and model-generated corresponding clarifying questions for each data set to test how reliably the language model generates the correct clarifying question (Table 3). According to our judgment, the model generates the correct clarifying question for 84%, 99% and 95% of the questions

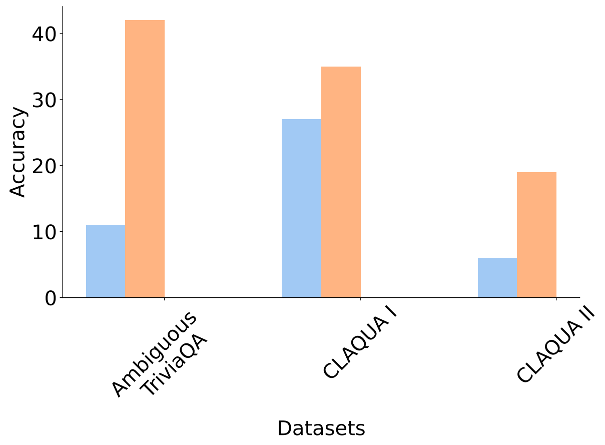


Figure 8. **`davinci`: Prompting the model to ask the user for clarification improves QA accuracy on ambiguous questions.** ClariQ does not contain answers for the ambiguous questions and is thus not displayed in this figure.

Table 3. **LMs can accurately generate clarifying questions and provide clarification during evaluation**. Human evaluation of each of the conversational turns. *ClariQ* does not contain answers for the ambiguous questions and is thus not displayed in this table.

METHOD	AMBIGUOUS TRIVIAQA	CLAQUA I	CLAQUA II
STEP 2:CORRECT CLARIFYING QUESTION	84.0%	99.0%	95.0%
STEP 3: CORRECT ORACLE ANSWER	98.8%	67.7%	98.7%

in Ambiguous TriviaQA, CLAQUA I and CLAQUA II respectively. We find that on TriviaQA the model sometimes asks incorrect clarifying questions either by just repeating the given ambiguous question or by asking an unrelated additional question about the subject of the ambiguous question.

Steps 2 through 4: Final answer after clarification

Ignoring the task of detecting ambiguity and focusing only on questions which are known to be ambiguous, asking the user for clarification lets the model answer ambiguous questions much more accurately for all data sets (Figure 9).

Step 3: Oracle automatic evaluation

Our oracle language model (Section 3) provides accurate clarifying information given a clarifying question by the language model under evaluation. We manually label 100 randomly sampled conversations based on ambiguous questions from Ambiguous TriviaQA, CLAQUA I and CLAQUA II. We report for what fraction of correct clarifications the oracle language model provides.

The oracle model reliablly provides the correct clarifying information on TriviaQA and CLAQUA II, see Table 3. On CLAQUA I the clarifying information has to disambiguate two possible meanings of a proper noun. We find that, while the model is generally able to do so, it will sometimes provide information that applies to both possible entities, and thus does not correctly disambiguate the two options.

F.1. Additional analysis of CLAM performance

In Figure 5a in the main part of the paper, we show that CLAM improves the question-answering accuracy on the full Ambiguous TriviaQA data set. In this section, we additionally compare the different baselines on the ambiguous and unambiguous questions separately, see Figure 10. We find that, as expected, both *Force clarification* and *CLAM* lead to a large accuracy improvmenet on ambiguous questions, and largely leave the performance on unambiguous questions

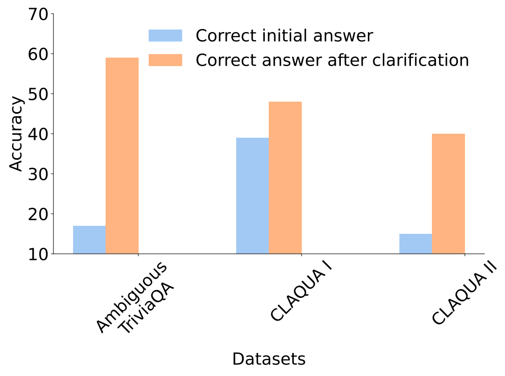


Figure 9. **Prompting the model to ask the user for clarification improves QA accuracy on ambiguous questions.** ClariQ does not contain answers for the ambiguous questions and is thus not displayed in this figure.

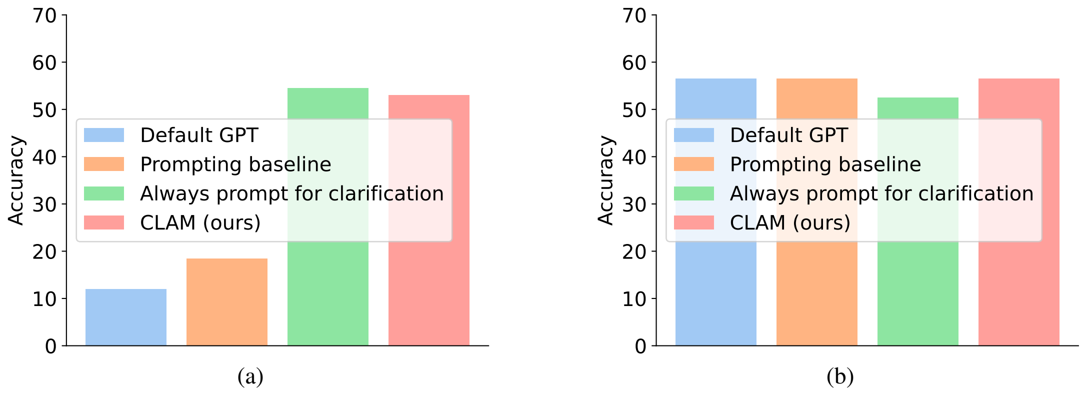


Figure 10. (a) **Accuracy on ambiguous questions from Ambiguous TriviaQA only:** CLAM and always prompting the language model to ask the user for clarification yields large improvements in accuracy over the default GPT behaviour and the prompting baseline. (b) **Accuracy on unambiguous questions from Ambiguous TriviaQA only:** The model performance on unnecessary questions remains largely unaffected. Note that always prompting for clarification leads to unnecessary turns of conversation on unambiguous questions which is in itself undesirable and sometimes decreases the accuracy by leading the conversation off-topic.

unaffected.

Table 4. Overview of which data sets can be used to evaluate which of the steps of selective clarification QA.

	CLARIQ	CLAQUA I	CLAQUA II	AMBIGUOUS TRIVIAQA (OURS)
1B: DETECT AMBIGUITY	√	√	√	√
2: ASK CLARIFYING QUESTIONS		√	√	√
2-4: FINAL ACCURACY (AMBIGUOUS ONLY)		√	√	√
2-4: FINAL ACCURACY (UNAMBIGUOUS ONLY)				√

G. Additional information on data sets and prompts

In this section, we provide additional details on the data sets and prompts used in our experiments.

G.1. Coverage of different pipeline steps

See Table 4 for an overview which of the data sets can be used to evalaute which of the pipeline steps. Our data set Ambiguous TriviaQA is the only data set that can be used to evaluate all steps of the pipeline.

G.2. Ambiguous TriviaQA

As described in Appendix B, we derive the ambiguous TriviaQA from the original TriviaQA Joshi et al. (2017) data set. Given an unambiguous question from the TriviaQA data set, we derive an ambiguous question as follows:

- Replacing a name or noun with a generic pronoun, e.g. “Where in England was Dame Judi Dench born?” becomes “Where in England was she born?”.
- Replacing a noun phrase with a class the noun belongs to, e.g. “Which country is Europe’s largest silk producer?” becomes “Which country is Europe’s largest producer?”

Ambiguity detection

```
Q: Who was the first woman to make a solo flight across this ocean?
This question is ambiguous: True.

Q: Who was the first woman to make a solo flight across the Atlantic?
This question is ambiguous: False.

Q: In which city were Rotary Clubs set up in 1905?
This question is ambiguous: False.

Q: Who along with Philips developed the CD in the late 70s?
This question is ambiguous: False.

Q: Where is the multinational corporation based?
This question is ambiguous: True.

Q: [question to be classified]
This question is ambiguous:
```

As explained in the main part of the paper, we then take the log probability of the next token being `True` as a continuous predictor of whether the given question is ambiguous or not.

Clarifying question generation

```
This is a conversation between a user and a question-answering bot.
User: On what date did he land on the moon?
Bot: To answer this question, I need to ask the following clarifying question:
Who is he?

###

User: Which country on this continent has the largest population?
Bot: To answer this question, I need to ask the following clarifying question:
Which continent?

###

User: {initial_question}
Bot: To answer this question, I need to ask the following clarifying question:
```

G.3. ClariQ

ClariQ (Aliannejadi et al., 2020) is based on search engine queries from the TREC Web Track 2009-2012 data set (Clarke et al., 2012). Every query in the data set has a human label *clarification need* (1-4) that indicates how unclear the human labeller perceives this question to be. We convert this data into a binary classification data set by labelling queries with clarification needs of 1 and 2 as *unambiguous*, and queries with clarification needs of 3 and 4 as *ambiguous*.

This data set does not contain reference answers for the given questions. We can thus only use this data set to measure ambiguity detection

Ambiguity detection

```
This bot determines whether a given question is ambiguous or not.
Question: Tell me about Obama family tree.
This question is ambiguous: False.

Question: TV on computer.
This question is ambiguous: True.

Question: What is Fickle Creek Farm
This question is ambiguous: False.

Question: Find condos in Florida.
This question is ambiguous: True.

[Three additional examples]

Question: [question to be classified]
This question is ambiguous:
```

As explained in the main part of the paper, we then take the log probability of the next token being `True` as a continuous predictor of whether the given question is ambiguous or not.

G.4. CLAQUA I

This data set corresponds to the single-turn part of the CLAQUA (Xu et al., 2019) data set. A given sample in this data set consists of the following components: two fields *entity1* and *entity2* that provide descriptions on two entities which both have the same name but are two distinct entities. Each sample also contains a question that is either ambiguous or unambiguous, that is a question that could possibly refer to either of the two entities or not. For the ambiguous question, the data set additionally contains a label that indicates which of the two entities the question is supposed to refer to, as well as an answer to the ambiguous initial question. This data set does not contain answers for the unambiguous questions in the data set.

As done in Xu et al. (2019), the description of the two entities is provided to the question answering model together with the given question.

Ambiguity detection

We use the following few-shot prompt to detect whether a given question is ambiguous or not

```
This bot determines whether a given question is ambiguous or not.

entity1: Aggrenox  Aggrenox (Aspirin, Dipyridamole) (...)
entity2: Aggrenox Aggrenox contains a combination of (...)
"What is Aggrenox’s ingredient?" could refer to both entities "Aggrenox": True.

entity1: Jack Shelton Jack Shelton (born November 30, 1931) is an American bebop and(...)
entity2: Jack Shelton John Jack Shelton was an English footballer who played as a right-half
"What are the written works of Jack Shelton?" could refer to both \\
entities "Jack Shelton": False.

[Three additional examples]

[description of entity1]
[description of entity2]
[question] could refer to both entities [entity name]:
```

As explained in the main part of the paper, we then take the log probability of the next token being `True` as a continuous predictor of whether the given question is ambiguous or not.

Clarifying question generation

```
Question: Casting director for Fakers
When you say Fakers, are you referring to the TV movie or the movie?

###

Question: What is name of place where Ernest Pollard was born?
When you say Ernest Pollard, are you referring to the Nebraska Republican politician or the p

###

Question: Cunningham Elementary\’s rank
Which Cunningham Elementary are you referring to?

###

Question: Who is the host of Room 101?
Which Room 101?


###
Question: {initial_question}
```

G.5. CLAQUA II

This data set corresponds to the multi-turn part of the CLAQUA (Xu et al., 2019) data set. Every sample of the data set contains the following components: A three-turn dialogue where the last turn is a question. For some samples, this question can refer to

Ambiguity detection

```
This bot determines whether a given question is ambiguous or not.

Context: Bazil Broketail. Bazil Broketail (1992) is a fantasy novel written by (...)
Context: A Sword  for a Dragon. A Sword for a Dragon is a fantasy novel written (...)
User: Is there a sequel to bazil broketail
Bot: A Sword for a Dragon
User: How about the style of this creative work?
"How about the style of this creative work?" could refer both \\
to Bazil Broketail and A Sword for a Dragon: True

###

Context: 807 Ceraskia.  807 Ceraskia is a minor planet orbiting the Sun.
Context: Eos family.  The Eos family (adj. Eoan; FIN: 606) is a very large asteroid (...)
User: What is family for 807 ceraskia
Bot: Eos family
User: What is the star system of the object
"What is the star system of the object" could refer both \\
to 807 Ceraskia and Eos family: False.

[Three additional examples]

###

[context]
[dialogue]
{question} could refer both to [entity1] and [entity2]:
```

As explained in the main part of the paper, we then take the log probability of the next token being `True` as a continuous predictor of whether the given question is ambiguous or not.

Clarifying question generation

```
This is a conversation between a user and a question-answering bot.
Is there a sequel to bazil broketail <EOS>\\
A Sword for a Dragon <EOS>\\
How about the style of this creative work?
Bot: To answer this question, I need to ask the following clarfiying question:
Are you referring to bazil broketail or a sword for a dragon?

###

What is higher classification for schadonia <EOS>\\
Lecanorineae <EOS>\\
Which cataloge it should be classified into?
To answer this question, I need to ask the following clarfiying question:
Are you referring to schadonia or Lecanorineae?
```

```
[Two additional examples]

###

{question}
Bot: To answer this question, I need to ask the following clarfiying question:
```

H. CLAM Algorithm

We tune the decision threshold τ used to detect ambiguity in CLAM (see Algorithm 1) to maximize the QA accuracy on a holdout data set of ambiguous and unambiguous questions and use $\tau = -0.3$ in the remaining experiments.

Algorithm 1 Selective clarification of imprecise questions

```
Require: A language model $\mathcal{M}$, a question $\mathcal{Q}$, a user $\mathcal{U}$, ambiguity classifier $f$.
  $\mathcal{A} \leftarrow \mathcal{M}(\mathcal{Q})$ {Ask language model to answer question $\mathcal{Q}$}
  ambiguous $\leftarrow f(\mathcal{Q}, \mathcal{M})$ {Classify ambiguity, e.g., with few-shot prompted LLM}
  if ambiguous then
    $\mathcal{Q}' \leftarrow concat(\mathcal{Q}$, “To answer this question, I have to ask the following clarifying question”)
    $\hat{\mathcal{Q}} \leftarrow \mathcal{M}(Q')$ {Ask a clarifying question $\hat{\mathcal{Q}}$}
    $\hat{\mathcal{A}} \leftarrow \mathcal{U}(concat(Q, \hat{\mathcal{Q}}))$ {Get clarification from user or oracle}
    $\mathcal{A} \leftarrow \mathcal{M}(concat(Q, \hat{\mathcal{Q}}, \hat{\mathcal{A}}))$ {Return answer given entire dialogue}
  end if
  return $\mathcal{A}$
```