

CLAM: SELECTIVE CLARIFICATION FOR AMBIGUOUS QUESTIONS WITH LARGE LANGUAGE MODELS

Lorenz Kuhn* Yarin Gal Sebastian Farquhar

OATML, Department of Computer Science, University of Oxford

ABSTRACT

State-of-the-art language models are often accurate on many question-answering benchmarks with well-defined questions. Yet, in real settings questions are often unanswerable without asking the user for clarifying information. We show that current SotA models often do not ask the user for clarification when presented with imprecise questions and instead provide incorrect answers or ‘hallucinate’. To address this, we introduce CLAM, a framework that first uses the model to detect ambiguous questions, and if an ambiguous question is detected, prompts the model to ask the user for clarification. Furthermore, we show how to construct a scalable and cost-effective automatic evaluation protocol using an oracle language model with privileged information to provide clarifying information. We show that our method achieves a 20.15 percentage point accuracy improvement over SotA on a novel ambiguous question-answering data set derived from TriviaQA.

1 INTRODUCTION

Recent Transformer-based large language models 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 that contain 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

*Correspondence to lorenz.kuhn@cs.ox.ac.uk

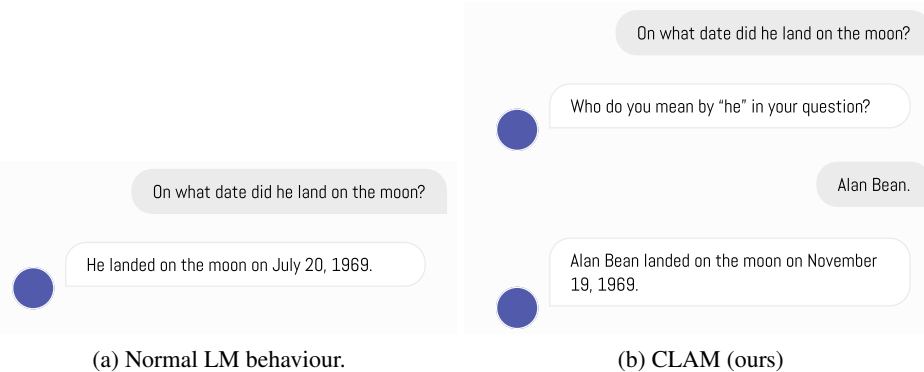


Figure 1: (a) Normally, LMs answer one of many interpretations given an ambiguous question. (b) Our method uses few-shot classification to detect ambiguous questions and selectively asks for clarifying information needed to answer the question.

far received little attention in the large language modelling 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.

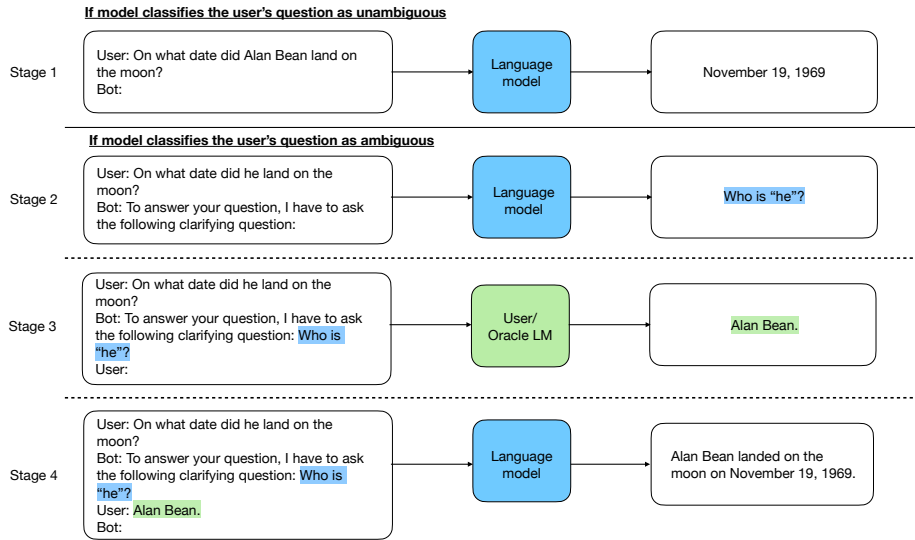


Figure 2: **Overview of the prompts used to clarify ambiguous user inputs.** If the language model classifies a given question as unambiguous (using few-shot prompting), the language model’s answer to the question is directly returned to the user (Stage 1). If the model classifies the given question as ambiguous, stage 2 is entered. In stage 2, the model is prompted to generate a clarifying question about the ambiguous user input. In stage 3, the user or an oracle model (see Section 4) provide clarifying information given the clarifying question. In stage 4, the model is prompted to answer the initial question given the additional information from the user.

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 existing performance. We also show that this method reliably only asks for clarification when the user input is actually ambiguous 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 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 datasets.

In summary, our contributions are:

- We introduce the **CLAM framework** for detecting ambiguous questions and clarifying them through multi-turn dialogue in LLMs (Section 3).
- We show that our implementation of CLAM delivers a 20.15 pt. adjusted accuracy improvement over current techniques on a QA data set that includes ambiguous questions (Section 5).
- We introduce the concept of language model **meta-cognition** as a general category of which CLAM is a proof-of-concept (Section 3.1).
- We codify an automatic evaluation protocol for **selective clarification QA** (Section 4).

In Section 2, we introduce our selective clarification QA task and the data set we use to evaluate it. In Section 3, we describe our framework for getting large language models to ask for clarification about ambiguous user requests. In Section 4, we describe how we can automatically evaluate how good a language model is at asking for clarification about ambiguous user questions. In Section 5, we present our experiments and show that our method leads to large performance improvements in answering ambiguous questions while not affecting the performance on answering unambiguous questions. We review related work in Section 6. In Section 7, we draw conclusions and point to future research directions.

2 SELECTIVE CLARIFICATION QA DATA SET

In this section, we introduce a data set that allows us to evaluate the performance on **selective clarification QA**. 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 but on the other hand, ask the user for clarification if the user’s question is ambiguous.

To study this setting, we introduce a data set of pairs of questions. 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 4, we explain how this pairing of ambiguous and unambiguous questions lets us automatically provide clarifying information about ambiguous questions to the language model we want to evaluate.

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.

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, the authors 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.

Based on these results, our selective clarification QA dataset is a more effective evaluation tool requiring less factual knowledge. This lets us study clarification with current generation language models.

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.

Algorithm 1 Selective clarification of imprecise questions

Require: A language model \mathcal{M} , a question Q , a user \mathcal{U} , ambiguity classifier f .

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

3 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. We then classify the question 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. 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. We describe this formally in Algorithm 1.

Implementing the CLAM framework requires choosing a specific technique for:

- classifying questions as ambiguous or not ambiguous;
- 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 the following 5-shot prompt consisting of examples of ambiguous and unambiguous questions:

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:

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.

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 then produce a clarifying question using a zero-shot prompt. We simply append the string

“In order to answer this question, I have to ask the following clarifying question:”

to the original question. We then present the user with the model’s clarifying question that is generated based on this newly constructed prompt.

3.1 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.

4 AUTOMATIC EVALUATION PROTOCOL

In this section, we provide an automatic evaluation protocol that researchers can use as an evaluation data-generating process. We believe that the future of language model evaluation requires a shift from thinking about evaluation datasets as static objects.

At the same time, 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.

As an alternative, we 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 2). 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 3). Since our data set is a parallel corpus of 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. We show in §5 that our oracle has roughly 99% accuracy when answering clarifying questions.

5 EXPERIMENTS

In this section, we experimentally validate the ability of CLAM to successfully detect and seek clarification to ambiguous questions. We first demonstrate its overall improved performance relative to baselines at the end-to-end task relative to the baselines.

We then examine each step of the pipeline individually. We show that our method is able to successfully identify 91% ambiguous questions. Having identified ambiguous questions, we verify that it usually asks the correct clarifying question and that it can then answer ambiguous questions with almost the same accuracy as unambiguous questions. We further show that although adding our clarification framework slightly reduces performance on a dataset of entirely unambiguous questions because of the model can get confused by its own clarifying question, this effect is small.

Stage 3: Prompt to obtain clarifying information from oracle LM

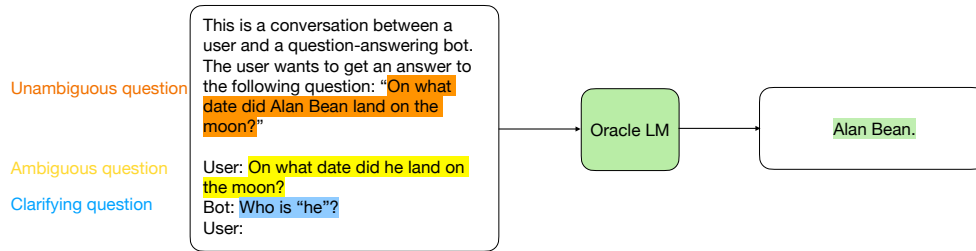


Figure 3: **Illustration of our automatic evaluation protocol.** We prompt a language model to provide clarifying information (green) given a clarifying question (blue) about an ambiguous user input (yellow). Our parallel corpus of ambiguous and corresponding unambiguous questions allows us to provide the unambiguous question to the oracle model (orange), based on which it can then provide appropriate clarifying information about the ambiguous question (yellow). See Figure 2 for a description of the other conversational turns.

5.1 EXPERIMENTAL SETUP

In principle, our framework should apply to any large language model architecture. In our experiments, however, we specifically use the `text-davinci-002` model via the OpenAI API. `text-davinci-002` (Ouyang et al., 2022), a GPT-3 model additionally fine-tuned on a range of tasks. This model is 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.

We measure the accuracy of a given model answer by evaluating whether the reference answer is contained in the model 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.

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.

In addition to providing 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 3.

We compare our methods against two prompting-based baselines. First, the approach we call **default GPT** uses a vanilla question-answering prompt instructing the model to answer the given question: This is a conversation between a user and a question-answering bot. Second, we test a prompt that explicitly asks the model to ask the user for clarifying questions if the user’s question is ambiguous: 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. We refer to this second method as **prompting baseline**.

5.2 RESULTS

We establish the overall performance of our implementation of the CLAM framework before investigating each step in the pipeline.

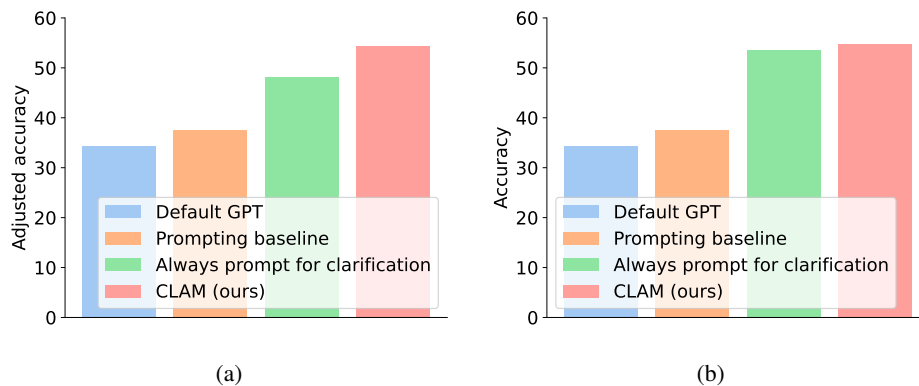


Figure 4: (a) **Adjusted accuracy on full data set:** CLAM improves accuracy on ambiguous questions without asking for unnecessary clarification on unambiguous questions (both of these desiderata are reflected in the adjusted accuracy metric). Always prompting the language model to ask the user for clarification 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.

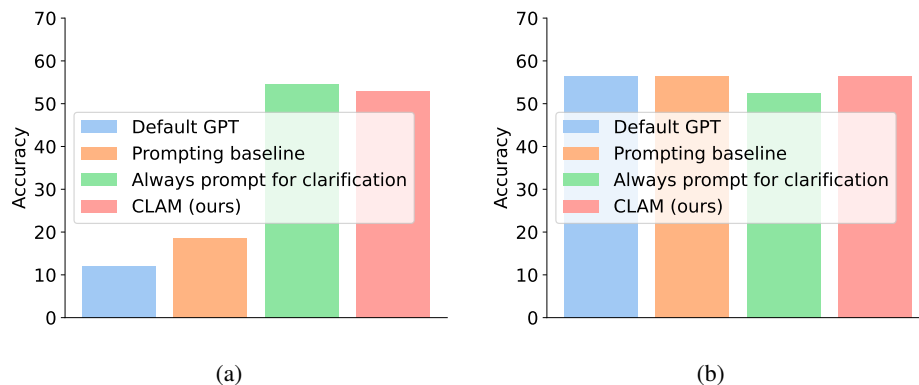


Figure 5: (a) **Accuracy on ambiguous questions only:** CLAM and always prompting the language model to ask the user for clarification yields large improvements in accuracy over the default GPT behaviour and the prompting baseline. (b) **Accuracy on unambiguous questions only:** The model performance 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.

5.2.1 OVERALL PERFORMANCE

Overall, we find that CLAM boosts the language model’s adjusted accuracy on a data set of both ambiguous and unambiguous questions from 34.25 to 54.4 (see Figure 4a). Recall that the adjusted accuracy multiplies the accuracy on a given question with a penalty term $\lambda = 0.8$ if the model unnecessarily asks for clarification on unambiguous questions as described in Section 5. Prompting the model to always ask for clarification improves the accuracy on ambiguous questions but incurs a large penalty on unambiguous questions for necessarily asking for clarification. Using the *prompting baseline*, that is instructing the model at the beginning of the question-answering conversation to ask the user for clarification if the user’s question is ambiguous, 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 (see Figure 4b).

On the ambiguous questions only, we find that, as expected, both always prompting the model to ask a clarifying question and CLAM greatly improve the question-answering accuracy as compared to the default GPT performance, see Figure 5a. Importantly, always clarifying and CLAM almost entirely close the gap on performance on ambiguous questions as compared to the performance on unambiguous questions, see Figure 5b. On the unambiguous questions, the different methods generally do not affect the model performance as compared to the default GPT behaviour. Note, however, that always prompting for clarification always leads to unnecessary turns of conversation on unambiguous questions which is generally undesirable from a user experience perspective, and sometimes actually hurts the accuracy by leading the conversation off-topic.

5.2.2 INDIVIDUAL PIPELINE COMPONENTS

Detecting Ambiguity

We find that the CLAM reliably distinguishes ambiguous from unambiguous questions. The few-shot prompting-based log probability of a given question being ambiguous or unambiguous achieves an AUROC of 0.93. For the decision threshold used in our experiments, CLAM achieves a true positive rate is 91% and a true negative rate of 90%, see Table 1. Per default, GPT does not ask for any clarifications neither on ambiguous or unambiguous questions. The prompting baseline only rarely asks for clarification on ambiguous questions and never asks for clarification on unambiguous questions.

Table 1: True positive and true negative rates of different approaches of detecting ambiguous questions. Default GPT never asks for clarification and thus has a TPR of 0.0 and a TNR of 1.0; always prompting for clarification yields a TPR of 1.0 and a TNR of 0.0. The prompting baseline rarely asks for clarification on ambiguous questions (TPR of 0.1) and never on unambiguous questions (TNR of 1.0). CLAM correctly detects most ambiguous questions (TPR of 0.91) and rarely asks for clarification of

Ambiguity Detection method	TPR	TNR
Default GPT	0.00	1.00
Always prompt for clarification	1.00	0.00
Prompting baseline	0.10	1.00
CLAM (ours)	0.91	0.90

Generating clarifying questions

We manually label 100 randomly selected pairs of ambiguous questions and model-generated corresponding clarifying questions to test how reliably the language model is able to generate the correct clarifying question, see Table 2. We find that for 84% of the ambiguous questions, the model is able to generate the correct clarifying question. Further improving the model performance in generating relevant clarifying questions is an interesting avenue for further research.

Table 2: Human evaluation of each of the conversational turns. We find that CLAM closes the performance gap between answering ambiguous and unambiguous questions almost entirely. Furthermore, the oracle model almost always provides the correct clarifying information for a given clarifying question. While the language model is good at finding the correct clarifying question given some ambiguous input, there is room for improvement on this task.

Turn of dialogue	Ambiguous questions	Unambiguous questions
Correct direct answer to initial question	17.0%	64.0%
Correct clarifying question	84.0%	—
Correct clarification by oracle given correct clarifying question	98.8%	—
Correct final answer after clarification	59.0%	62.0%

Automatic evaluation using the oracle model

Lastly, we verify how well our oracle language model (described in Section 4 and illustrated in stage 3 of Figure 2) provides clarifying information given a clarifying question by the language model under evaluation. To this end, we manually label 100 randomly sampled conversations based on ambiguous questions from our data set. We both label how often the language model under evaluation generates the correct clarifying question given the user request (84%) and then measure for how many of those clarifying questions, the oracle provides the correct clarifying information. We find that for 83/84 (98.8%) the oracle model provides the correct clarifying information, showing that our automatic evaluation protocol for the selective clarification QA task is highly reliable, see Table 2.

6 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.

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.

7 CONCLUSION

In this paper, we provide 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 a few-shot prompting approach. This additional clarifying conversational turn increases the model’s adjusted accuracy from 34.25 to 54.4. 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

Lorenz Kuhn gratefully acknowledges FHI at the University of Oxford for supporting this work through a DPhil Scholarship.

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.

REFERENCES

- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.
- Kaustubh D Dhole. Resolving intent ambiguities by retrieving discriminative clarifying questions. *arXiv preprint arXiv:2008.07559*, 2020.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2022.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.
- Kimiya Keyvan and Jimmy Xiangji Huang. How to approach ambiguous queries in conversational search? a survey of techniques, approaches, tools and challenges. *ACM Computing Surveys (CSUR)*, 2022.
- Dmitrii Krasheninnikov, Egor Krasheninnikov, and David Krueger. Assistance with large language models. In *NeurIPS ML Safety Workshop*, 2022. URL <https://openreview.net/forum?id=0E9V81spp6B>.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.
- Emily R Lai. Metacognition: A literature review research report. *Research Reports*, 41, 2011.
- Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*, 2020.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
- Sudha Rao and Hal Daumé III. Answer-based adversarial training for generating clarification questions. *arXiv preprint arXiv:1904.02281*, 2019.
- Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. Continuously learning neural dialogue management. *arXiv preprint arXiv:1606.02689*, 2016.
- Jan Trienes and Krisztian Balog. Identifying unclear questions in community question answering websites. In *European conference on information retrieval*, pp. 276–289. Springer, 2019.
- Mattias Wahde and Marco Virgolin. Conversational agents: Theory and applications. *arXiv preprint arXiv:2202.03164*, 2022.
- Jian Wang and Wenjie Li. 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.

A ADDITIONAL EXPERIMENTAL RESULTS

Table 3: **Adjusted accuracy results for different penalty terms on full 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 (see Table 1) 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