To Clarify or not to Clarify: A Comparative Analysis of Clarification Classification with Fine-Tuning, Prompt Tuning, and Prompt Engineering

Alina Leippert
alina.leippert@dfki.de
Tatiana Anikina
tatiana.anikina@dfki.de
Bernd Kiefer
bernd.kiefer@dfki.de
Josef van Genabith
josef.van_genabith@dfki.de
DFKI / Saarland Informatics Campus, Saarbrücken, Germany

Abstract

Misunderstandings occur all the time in human conversation but deciding on when to ask for clarification is a challenging task for conversational systems that requires a balance between asking too many unnecessary questions and running the risk of providing incorrect information. This work investigates clarification identification based on the task and data from (Xu et al., 2019), reproducing their Transformer baseline and extending it by comparing pre-trained language model fine-tuning, prompt tuning and manual prompt engineering on the task of clarification identification. Our experiments show strong performance with a joint LM and prompt tuning approach with BERT and RoBERTa, outperforming LM fine-tuning, while manual prompt engineering with GPT-3.5 proved to be less effective, although informative prompt instructions have the potential of steering the model towards generating more accurate explanations for why clarification is needed.

1 Introduction

Humans often communicate when they do not understand something and are able to collaboratively avoid and resolve misunderstandings by clarifying them. Clarification questions can be used to establish common ground between interlocutors (Clark and Brennan, 1991). Effectively repairing misunderstandings would be a desirable feature for conversational systems, thereby keeping the conversation between user and system as natural and efficient as possible. As noted by Rahmani et al. (2023), understanding users' underlying needs is critical for conversational systems, where the input is often limited to short questions. When system confidence in user intent is low, a clarification request (CR) should be generated to resolve ambiguity. However, handling uncertainty in conversational systems moves along a thin line between over- and under-generation of clarification (Skantze, 2007). Asking too many or unnecessary clarification questions can lead to user frustration (Xu et al., 2019), while asking too few runs the risk of providing the user with incorrect information. Hence, clarification request identification is an important task for conversational systems and it may also rely on additional information coming from a knowledge base (KB), as in the CLAQUA (Xu et al., 2019) dataset used in this work. We focus on clarification in a knowledge-based question answering (KBQA) setting and compare three approaches for modelling clarification identification with CLAQUA: pre-trained language model fine-tuning, prompt tuning and manual prompt engineering.

2 Related Work

While research on clarification in conversational systems was for a long time held back by a lack of datasets (Xu et al., 2019; Kumar and Black, 2020), Rahmani et al. (2023) now observe a growing number of research approaches and datasets on the topic. Datasets for clarification in question answering (QA) systems include RaoCQ (Rao and III, 2018) and ClarQ (Kumar and Black, 2020), built from StackExchange posts. Qulac is a dataset for conversational search introduced by Aliannejadi et al. (2019) and CLAQUA (Xu et al., 2019) supports clarification identification with a knowledge base.

Several approaches focus on identifying ambiguity in user queries to improve performance of KBQA systems. Wu et al. (2020) predict whether system confidence is high enough to answer the query before the user is asked to choose from a list of possible relevant entities. Guo et al. (2021) study to which extent neural models can generate CRs in conversational QA and introduce the Abg-CoQA corpus for clarifying ambiguities in reading comprehension questions. The NeurIPS NLP Shared

Task (Kiseleva et al., 2022) also addresses the problem of when the agent should ask for clarification using a simulated Minecraft environment to benchmark different models.

3 Experiments

Our work compares three different approaches of leveraging pre-trained language models (PLMs) for clarification identification in a KBQA system: language model (LM) fine-tuning, manual prompt engineering with large language models (LLMs) and prompt tuning. The task is to determine whether a CR is needed or the context already provides enough information to decide for which entity to answer the question.

3.1 Data and Methodology

The public release of CLAQUA[1] Xu et al. (2019) was used in all our experimental settings. The statistics of the publicly released data differ from the published version and are shown in Table 5 in Appendix A.1. CLAQUA consists of dialogues between a user and a KBQA system. The need for clarification arises through user questions which seem ambiguous at first glance. The corpus is split between single- and multi-turn dialogues. In the single-turn case, the ambiguity stems from an ambiguous entity label, that could refer to two entities in the KB which share the same surface string. In the multi-turn data, it comes from an unresolved referent, which could refer to either one of two previously mentioned entities. A multi-turn example from Xu et al. is:

> A_1: What is the name of the game played on Windows?
> C_2: "Insane."
> A_3: Who is its developer?

where "its" could refer to either the game or the operating system because both have a developer for which the question could be answered. Clarification identification is modelled as a binary classification task that relies on the context information that includes current (and previous, in the multi-turn case) conversation turn(s) and entity text descriptions.

Xu et al. (2019) provide a task baseline for clarification identification with several models including a Transformer (Vaswani et al., 2017) trained from scratch, but not yet a pre-trained Transformer-based LM. Nowadays, the use of PLMs leads to substantial progress on many NLP tasks (Brown et al., 2020). We explore PLM fine-tuning, manual prompt engineering and prompt tuning, comparing them to a Transformer baseline from Xu et al. (2019) reproduced for this work.

3.1.1 Fine-tuning

PLMs implicitly store a certain amount of knowledge acquired in pre-training (Roberts et al., 2020). This can be leveraged in a fine-tuning process on a downstream task, posing a convenient alternative to training a model from scratch. Using the HuggingFace (Wolf et al., 2020) library, we fine-tuned four models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), AlBERT (Lan et al., 2020) and DistilBERT (Sanh et al., 2019). Clarification identification is modelled with only the context and entity text descriptions, without any KB entity attributes. Input items are concatenated with separator tokens:

> [CLS] CONTEXT [CON_SEP] ENTITY1 TEXT [ENT_SEP] ENTITY2 TEXT [SEP]

A maximum input length of 300 tokens was chosen, truncating from both entity text descriptions where necessary, and the PLMs were fine-tuned with task-specific heads in the form of linear classification layers. Details on model architecture and hyperparameters can be found in A.2.

3.1.2 Manual Prompt Engineering

Open AI's GPT models, such as GPT-3 (Brown et al., 2020) and GPT-4 (OpenAI, 2023b), have recently gained remarkable success through their publicly available tool ChatGPT (OpenAI, 2023a). The LLMs are capable of inference processes: it is possible to let the model solve a task in a zero-shot setting, without fine-tuning or training a model from scratch. The textual inputs to the models, used for eliciting output in response to data and task, are called prompts.

Our manual prompt engineering experiments were conducted with the `gpt-3.5-turbo` model[2]. Its training data is up to September 2021, meaning that the CLAQUA corpus published in 2019 might have been included in the training data, which could give the model an advantage regarding clarification identification performance.

[1] https://github.com/msra-nlc/MSParS_V2.0

[2] https://platform.openai.com/docs/models

		P0	P1	P2	P3
Out	Classification	51	51	51	51
	Explanation	-	51	51	51
In	Context + entity descriptions	51	51	51	51
	Split previous and current turn	51	-	-	51
	Detailed task instruction	-	-	51	51
	Task instruction incl. previous turn	-	-	-	51

Table 1: Manual prompt engineering: Prompt0-3 (abbreviated as P0-3 in the Table), characterised based on output they ask for and input they provide.

The GPT-3.5 model was prompted with several task formulations, described in Table 1. The four prompts differ with regard to the output they are asked to provide as well as the input they receive with it. The full prompts can be found in Appendix A.3. All prompts ask whether clarification is needed for a given data item. Three prompts (Prompt1-3) additionally request an explanation. Two prompts (Prompt2&3) provide a detailed task instruction: in the prompts, it is explained how ambiguity arises in this specific task scenario and which steps are needed to reach a decision, followed by the question whether a clarification request is needed given the current data item. The steps include considering the entity text descriptions and deciding - based on the context provided through the conversation turn(s) - whether the user question is ambiguous in that it could be answered for both entities *(need for clarification)* or unambiguous in that the context implicitly specifies to which entity the question applies *(no need for clarification)*. Prompt3 is especially tailored to and only used for the multi-turn data, as its task instruction includes reference to the previous turn. The GPT-3.5 responses were evaluated for the correctness of their classification decisions, explanations as well as the following phenomena. For examples of the phenomena, see A.3.2.

- **Hallucination**: Based on the general definition of hallucination (for example, Ji et al. (2023)) in model-generated responses, defined as statements which are not supported by the external knowledge source, here: context and entity descriptions.

- **Omission**: The prompt response holds a statement which is not fully explained, e.g. the model jumps to a conclusion and leaves out important steps in the argumentation.

- **Incoherence**: The prompt response is not coherent, e.g. states something which is then negated or not in line with other statements in the response.

- **Focus-deviation**: The prompt response fails to produce an explanation which fits the task of clarification identification with the CLAQUA corpus.

The generative model was prompted with the `role` of "user" and the `temperature` (degree of randomness) was set to 0, as recommended for tasks that require reliability and predictability[3], which applies to a conversational system. The input information is the same as in the fine-tuning experiments, except for the length of the entity descriptions since GPT-3.5 can process longer inputs than e.g. BERT and the entity text descriptions were not truncated. An example of a prompt response for a CLAQUA item is presented in Figure 1. That the example given does not require clarification is well illustrated by the prompt response elicited by Prompt2, while the Prompt1 response fails to target the task. The two prompts differ in the level of detail of the task instruction.

Data:

`context`: "What is the setting of Oz?"

`entity1`: "Oz Daniel Oz Osbourne is a supporting character featured in the WB Network/UPN television series Buffy the Vampire Slayer. [...] "

`entity2`: "Oz The Land of Oz is a fantasy region containing four lands under the rule of one monarch. It was first introduced in The Wonderful Wizard of Oz by L. Frank Baum, one of many fantasy countries that he created for his books. [...] "

`label`: 0

Prompt: Prompt1

Model response: "Yes, the user question needs a clarification request. The user's question is not clear as it contains a lot of information about different characters and settings related to the name 'Oz'. It is not clear what specific information the user is seeking."

Prompt: Prompt2

Model response: "The user question does not need a clarification request. The question can only apply to Entity2, which is the Land of Oz described in the books by L. Frank Baum. The context of the user question suggests that they are asking about the setting of a story or a fictional world, which aligns with the description of Entity2. Entity1, on the other hand, is a character from a television series and does not have a setting of its own. Therefore, the user question can be answered for only Entity2 and does not need a clarification request."

Figure 1: Example for a single-turn GPT-3.5 prompt response. Prompted with Prompt2, it shows an explanation annotated as correct, while with Prompt1, we find a deviation of focus in the response.

We conducted fine-grained manual evaluation for 100 randomly selected samples from the single- and multi-turn test sets each and annotated them according to whether the model generated a plausible

[3] https://learn.deeplearning.ai/chatgpt-prompt-eng/

explanation (i.e., whether the reasoning provided by the model corresponds to the label) and whether the output includes any hallucinations, omissions, incoherence or focus-deviation. Note that the labels are not mutually exclusive and it is possible to have some overlap between them, e.g., omissions may lead to increased incoherence and hallucinations can result in focus-deviation. We computed the inter-annotator agreement on the single-turn test set and observed high agreement for the explanation-based evaluation: 0.82 Cohen's κ for Prompt1 and 0.75 for Prompt2. However, the inter-annotator agreement on the fine-grained errors was considerably lower, ranging from 0.31 to 0.52 Cohen's κ that shows the intrinsic difficulty of the task.

3.1.3 Prompt Tuning

Another approach to make use of the capabilities of PLMs is prompt tuning, where the downstream task is cast as a language modelling task (Vu et al., 2022). Each task example in this setting typically has a context and a desired completion (Brown et al., 2020), here the conversation turns and the entity descriptions with a binary prediction for clarification need. In this work, we explored two strategies, as identified by Liu et al. (2023): **Frozen-LM Prompt Tuning**, where the prompt parameters are updated while the LM parameters stay frozen and **Prompt+LM Tuning**, where the parameters of the prompt are updated together with the LM parameters. We used OpenPrompt framework (Ding et al., 2022) and experimented with T5 (Raffel et al., 2020), GPT-2[4] (Radford et al., 2019), BERT and RoBERTa . Input and truncation strategy follow the fine-tuning experiments. We optimized the hyperparameter settings on the development set and varied the number of additional soft tokens from 0 to 100. The results reported in Section 4.4 were achieved with the best performing configuration for each prompt. All models were tuned for three epochs.

4 Results and Discussion

4.1 Baseline

The clarification identification baseline with the models from Xu et al. (2019) was reproduced, showing the Transformer scores in Table 2. The difference between our baseline scores and (Xu et al., 2019) can be attributed to the smaller size of the published single-turn data and the additional pre-processing we implemented to make sure that none of the truncated entity spans is missing from the input.

[4]The latest GPT-model available on Huggingface.

4.2 Fine-tuning

Table 2 shows a comparison of the clarification identification results from fine-tuning different models from the BERT-family. We use the term "F1" to refer to the macro-averaged F1 score.

Model	Data	Acc	F1
BERT	Single	0.884	0.881 ± 0.006
RoBERTa	Single	0.896	**0.893** ± 0.009
AlBERT	Single	0.775	0.758 ± 0.021
DistilBERT	Single	0.873	0.869 ± 0.004
Xu et al. Baseline	Single	-	0.811 ± 0.002
BERT	Multi	0.928	0.928 ± 0.016
RoBERTa	Multi	0.952	**0.952** ± 0.023
AlBERT	Multi	0.737	0.737 ± 0.044
DistilBERT	Multi	0.916	0.916 ± 0.032
Xu et al. Baseline	Multi	-	0.727 ± 0.027

Table 2: Fine-tuning performance for clarification identification. Results are averaged over three model runs and shown with standard deviation.

The results show that the classifier built with RoBERTa performs best for both single- and multi-turn data: The best results for clarification identification are an F1-score of 89.3% on the single-turn and 95.2% on the multi-turn data. The scores comprehensively beat the reproduced baseline with a Transformer trained from scratch, showing that the use of the pre-trained models from the BERT-family is of benefit for this task and data. However, the results also show that the choice of the specific model from the BERT-family makes a remarkable difference in task performance.

A possible explanation for the differences in results can be made based on the model sizes. RoBERTa, the best-scoring model on the task, has the highest number of parameters (125M) and the task performance gradually decreases with the model size which is consistent with the previous findings (Devlin et al., 2019; Hernandez et al., 2021). In contrast to the baseline, the PLMs show better results for the multi-turn data than for single-turn, with all models except for AlBERT. A possible reason could be the difference in train dataset size (since the multi-turn one is twice as large as single-turn, see Table 5). Another explanation was proposed by Xu et al. (2019) and it is based on the idea that multi-turn conversations include more context, which can help the models to better capture the entity information.

4.3 Manual Prompt Engineering

The performance of a manual prompt engineering approach for clarification identification with GPT-3.5 is shown in Table 3. For each item, we evaluated whether GPT-3.5 predicted the correct class label (e.g. the response "Yes, clarification is needed" corresponds to the positive class), resulting in F1-scores. Further, for each item where the class was correctly predicted by the model, we annotated whether the model-generated response for *why* this item needs clarification (or not) can be categorised as correct from a human perspective.

Prompt	Data	Acc	F1	Explanation
Prompt0	Single	0.48	0.40	-
Prompt1	Single	0.41	**0.57**	22.5%
Prompt2	Single	0.39	0.40	56.5%
Random Baseline	Single	0.46	-	
Prompt0	Multi	0.56	0.27	-
Prompt1	Multi	0.52	**0.68**	9.6%
Prompt2	Multi	0.48	0.56	30.6%
Prompt3	Multi	0.47	0.40	41.3%
Random Baseline	Multi	0.53	-	

Table 3: GPT-3.5 clarification identification results for single- and multi-turn data, showing accuracy, F1-score and the amount of model-generated explanations annotated as correct.

With the best single-turn F1-score at 57% and multi-turn at 68%, GPT-3.5 is not able to beat the Transformer baseline introduced in Section 4.2. Furthermore, we find that the manual prompting results scored by GPT-3.5 barely beat a random uniform baseline accuracy on the task (see Table 3). However, as judged by human evaluation, the model can generate increasingly correct explanations for why an item needs clarification when provided with an informative prompt (such as Prompt2 or Prompt3, see A.3).

While the number of correct explanations grows with more elaborate prompts, the results show a lot of room for improvement. For the single-turn sample, the highest number of correct explanations is 56.5%, for multi-turn 41.3%, indicating that the single-turn data can be better processed by GPT-3.5. Figure 2 shows, for each prompt, the percentage of responses with: incoherence, omission, hallucination and focus-deviation. The categories are not mutually exclusive, a response may include several phenomena at once. The evaluation shows that the high amount of focus-deviations can be reduced considerably by providing more informative task instructions in the prompt. However, the number of hallucinations, omissions and incoherence grows with more informative prompts (except for incoherences in multi-turn responses, which can be reduced with Prompt3).

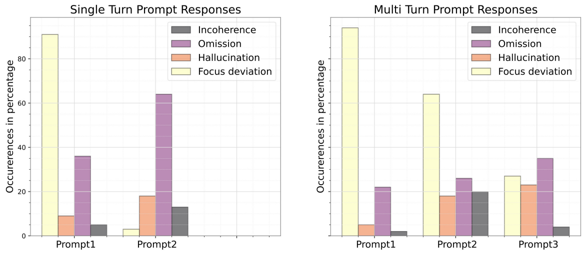

Figure 2: Percentage of GPT-3.5 responses showing focus-deviation, hallucination, omission and incoherence for single- and multi-turn data with Prompt1-3.

We also found that prompting with GPT-3.5 can point out cases where the entity descriptions are uninformative, e.g. just consisting of links. Cases like this can occur especially when the underlying KB is partly constructed automatically.

4.4 Prompt tuning

The clarification identification results with a prompt tuning approach are shown in Table 4. The results were scored with the *Prompt+LM tuning* strategy, since it became apparent that this leads to much better results for clarification identification on CLAQUA than *Frozen-LM Prompt Tuning*. Preliminary results with T5 showed that freezing the LM and tuning only the prompt results in a huge performance drop (of around 30% in accuracy and 50% in macro F1-score, even when provided with a longer training time of 10 epochs).

Model	Data	Acc	F1
T5	Single	0.888	0.885
GPT-2	Single	0.868	0.864
BERT	Single	0.877	0.875
RoBERTa	Single	0.896	**0.894**
T5	Multi	0.964	0.964
GPT-2	Multi	0.949	0.949
BERT	Multi	0.981	**0.981**
RoBERTa	Multi	0.978	0.978

Table 4: Prompt tuning performance for clarification identification, comparing different PLMs. The results were obtained with the best-performing prompt in each case (for details on the prompts, see A.4).

For the single-turn data, the best result was achieved when tuning RoBERTa, showing an F1-score of 89.4%. For the multi-turn data, BERT scores the best results with 98.1% F1-score. All models perform better on the multi- than on the single-turn data, with a difference of almost ten percent between the best results. Adding 50 tunable soft prompts was beneficial for task performance. Regarding the prompt formulation and using hard vs. soft prompts, no clear pattern emerged which confirms the findings of inconsistent model performance when using manual prompts as reported by Zhao et al. (2021).

5 Conclusion

Our comparative analysis of different approaches shows that LM fine-tuning and Prompt+LM tuning lead to good task performance. The best clarification identification results on CLAQUA are achieved with a joint LM and prompt tuning approach. The results indicate that the linguistic knowledge gained from pre-training can be leveraged with Transformer-based LMs, modelling the clarification identification task with only the conversation context and entity text descriptions.

For future work, we consider the use of various other models, for example DeBERTa (He et al., 2020) or ELECTRA (Clark et al., 2020). Other promising research directions include: (1) generating clarification questions and joint modeling of clarification identification and generation, (2) conducting a user study to investigate how users react to under- and over-represented clarification questions in dialogue and (3) analysing to what extent state-of-the-art dialogue systems can benefit from explicit clarification question identification.

Manual prompt engineering with GPT-3.5 was not competitive in terms of clarification identification scores. However, with informative prompt instructions, manual prompt engineering can be used for deeper analysis of the interaction between context and entity information and the reasoning process for why user questions need clarification or not. Even though prompt responses leave room for improvement, they show a direction worth exploring further.

6 Acknowledgements

We would like to thank the anonymous reviewers for their valuable feedback on our work and Katharina Jana Christian for the help with annotations. The authors were supported by the German Ministry of Education and Research (BMBF): A. Leippert and T. Anikina in project CORA4NLP (grant Nr. 01IW20010).

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 475–484, Paris France. ACM.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Herbert H. Clark and Susan E. Brennan. 1991. Grounding in communication. In Lauren B. Resnick, John M. Levine, and Stephanie D. Teasley, editors, *Perspectives on socially shared cognition*, pages 127–149. American Psychological Association.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. Openprompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022 - System Demonstrations, Dublin, Ireland, May 22-27, 2022*, pages 105–113. Association for Computational Linguistics.

Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. 2021. Abg-CoQA: Clarifying ambiguity in conversational question answering. In *3rd Conference on Automated Knowledge Base Construction*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. 2021. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Julia Kiseleva, Alexey Skrynnik, Artem Zholus, Shrestha Mohanty, Negar Arabzadeh, Marc-Alexandre Côté, Mohammad Aliannejadi, Milagro Teruel, Ziming Li, Mikhail Burtsev, Maartje ter Hoeve, Zoya Volovikova, Aleksandr I. Panov, Yuxuan Sun, Kavya Srinet, Arthur Szlam, and Ahmed Awadallah. 2022. IGLU 2022: Interactive grounded language understanding in a collaborative environment at neurips 2022. *CoRR*, abs/2205.13771.

Vaibhav Kumar and Alan W. Black. 2020. Clarq: A large-scale and diverse dataset for clarification question generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7296–7301. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

OpenAI. 2023a. ChatGPT [Large language model].

OpenAI. 2023b. GPT-4 technical report. *Preprint*, arxiv:2303.08774.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Hossein A. Rahmani, Xi Wang, Yue Feng, Qiang Zhang, Emine Yilmaz, and Aldo Lipani. 2023. A survey on asking clarification questions datasets in conversational systems. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2698–2716. Association for Computational Linguistics.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2737–2746. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5418–5426. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Gabriel Skantze. 2007. *Error Handling in Spoken Dialogue Systems : Managing Uncertainty, Grounding and Miscommunication*. Ph.D. thesis, Royal Institute of Technology, Stockholm, Sweden.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022. Spot: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5039–5059. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, and Morgan Funtowicz. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Zhiyong Wu, Ben Kao, Tien-Hsuan Wu, Pengcheng Yin, and Qun Liu. 2020. PERQ: Predicting, explaining, and rectifying failed questions in KB-QA systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 663–671.

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. 2019. 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)*, pages 1618–1629, Hong Kong, China. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

A Appendix

A.1 CLAQUA Corpus

CLAQUA Single-Turn

	Positive	Negative	*Total*
Train	3,507	6,592	10,099
Dev	431	422	853
Test	503	672	1,175
Total	4,441	7,686	12,127

CLAQUA Multi-Turn

	Positive	Negative	*Total*
Train	12,173	8,289	20,462
Dev	372	601	973
Test	384	444	828
Total	12,929	9,334	22,263

Table 5: Statistics of the CLAQUA corpus as found in the released corpus version on Github.

A.2 Fine-tuning Experiments

Several classifier architectures and hyperparameter configurations were tested. Experiments include feed-forward neural network architectures consisting of one, two and three hidden layers on top of the Transformer output and test ReLU and Tanh activations. The whole model, including the Transformer layers, was trained, comparing three different learning rates (2e-5, 3e-5 and 5e-5, as recommended for fine-tuning by Devlin et al. (2019)) and two batch sizes (16 and 32). The models were each trained for 10 epochs, picking the best model on the validation data for test data evaluation based on macro-averaged F1 score.

A.3 Manual Prompt Engineering Experiments

A.3.1 Prompts

For the manual prompt engineering approach, the following prompts were used:
Prompt0 is a simple prompt asking for a binary answer, either "yes" or "no", without explanation.
Instruction: "Does the following user question to a knowledge-based question answering system need a clarification request or not? Answer with 'yes' or 'no'."
Data: The input corpus items are given in form of each sub-item (context and entity descriptions), in the prompt indicating the structure. For the multi-turn data, the context is split between previous and current turn, providing them separately.
Prompt1 asks for classification as well as explanation. The prompt is the same for single- and multi-turn.
Instruction: "Does the following user question to a knowledge-based question answering system need a clarification request or not and why?". *Data:* The corpus items are given as a concatenation of the context and the two entity descriptions, without indicating the structure in the prompt. The multi-turn context is provided as concatenation of previous turns and current turn.
Prompt2 provides a detailed task instruction. When providing the corpus item, it splits context and entities explicitly. The prompt formulation is shown without formatting:
Instruction: "Your task is to determine whether the following user question to a knowledge-based question answering system needs a clarification request or not. To fulfill the task, do the following: First, consider the context given in the user question. The knowledge base holds two entities, entity1 and entity2, to which this user question could refer to. Read the text descriptions of both entities. There are two options: If the user question can be answered for only entity1 or for only entity2, the user question is not ambiguous and does therefore not need a clarification request. If the user question can be answered for both entities, it is an ambiguous question and needs a clarification request. By reading the descriptions of both entities, decide whether the user question needs a clarification request or not. State whether the user question needs a clarification request or not. If the user question needs a clarification request, state why the question can apply to both entities. If it does not need a clarification request, state why the question can only apply to one of the entities and which."
Data: The input corpus items are again given in form of each sub-item, in the prompt indicating the structure: `context: [...] + entity1 description: [...] + entity2 description: [...]`. The multi-turn context is given as concatenation of previous turns and current turn.
Prompt3 is a modification of Prompt2 tailored to the multi-turn data. It splits previous and current turns in the input and also mentions that the ambiguity arises from the entities mentioned in the previous turn.

A.3.2 Prompt Response Phenomena

The prompt responses were manually annotated regarding for the presence of the following phenomena (which are not mutually exclusive). An example of each phenomenon is given below:

Hallucination

> `Entity2 description`: "The Magic Finger is a 2013 theater production of the play by David Wood."
>
> `Prompt response`: "Entity2 is a theater production of the play and does not have a cover artist."

This is an example of hallucination. That the play does not have a cover artist is not supported by the entity description.

Omission

> `Context`: "On the software license of VTune, what is stated?"
>
> `Entity1 description`: "VTune puts the wealth of Intel's hardware knowledge to work for analyzing and optimizing your software's performance. This developer's utility features multiple perspectives–you can zoom in or take a bird's-eye view using static or dynamic code-analysis functions. Once VTune spots the inefficiencies, it sends in the VTune Coach to suggest solutions. This latest version contains full support for Java. This demo is a full-featured, 30-day trial."
>
> `Entity2 description`: "VTune Amplifier performance profiler is a commercial application for software performance analysis of 32 and 64-bit x86 based machines. It has both a GUI (graphical user interface) and command line and comes in versions for Linux or Microsoft Windows operating systems. An optional download lets you analyze the Windows or Linux data with a GUI on OS X. Many features work on both Intel and AMD hardware, but advanced hardware-based sampling requires an Intel-manufactured CPU."
>
> `Prompt response`: "The user question does not need a clarification request. The question can only apply to Entity2 because the description of Entity1 does not mention anything about a software license."

This prompt response shows an omission of facts. The argument for no clarification is based on the prompt response stating that the description of Entity1 would not mention a software license. It suggests that the description of Entity2 does mention a software license, which is not the case.

Incoherence

> `Prompt response`: "Entity2, Chris Robinson, is described as an American actor, screenwriter, and film director, but there is no mention of him directing any movies. Therefore, the user question can only refer to Entity1."

This response shows incoherence in reasoning. In the response it is mentioned that the second entity is described as "film director" and at the same time it is argued that there would be no mention of them directing any movies.

Focus-deviation:

> `Prompt response`: "Yes, the user question needs a clarification request. The question is not clear about what specifically they are asking for in [user question]. It mentions two different entities named [entity name]. The system needs to ask for clarification on which [entity name] the user is referring to."

Generic responses as the one shown here are an example of focus-deviation. They fail to target the task being addressed, namely to explain whether a user question in a specific context needs a clarification request or not with respect to the given entities in the KG.

A.4 Prompt Tuning Experiments

For the prompt tuning approach, the following prompts were used:

- **Prompt1** consists of the corpus item and soft tokens. The question "Does this context need clarification?" is initialized to the specified tokens which are then tuned. It is followed by a mask token which the model must fill with a clarification identification decision.

- **Prompt2** consists of the corpus item, soft tokens and a textual template in the form of hard tokens for the question of whether the provided context needs clarification. It has the same structure as Prompt1, the only difference being whether soft or hard tokens are used.

- **Prompt3-7** consist of slight modifications of Prompt2 regarding the hard tokens. They modify the prompt formulation by mentioning the knowledge base, the two entities or rephrasing the task into an ambiguity problem.