Asking Clarifying Questions

Bhushan Malani 19CS10020
Harshit Jindal 19CS10034
Kaushal Banthia 19CS10039
Jayant PSY 19CS10068

Abstract

This project aims to improve open-domain dialogue systems by developing models that can effectively ask clarifying questions. Specifically, the project focuses on two subtasks: determining whether a clarifying question is needed for a given instruction and ranking a list of possible clarifying questions based on their relevance. The evaluation metrics used for these subtasks are macro average F1 score and Mean Reciprocal Rank (MRR), respectively. The project utilizes a dataset and starter code available on GitHub. The results of this project can contribute to enhancing the naturalness and effectiveness of open-domain dialogue systems.

1 Introduction

Open-domain dialogue systems have become increasingly popular in recent years, as they allow users to interact with computers in a more natural and conversational way. However, these systems often struggle to understand and respond appropriately to user input, especially when the input is ambiguous or lacks context. To address this issue, this project focuses on the task of asking clarifying questions, which can help to obtain additional information and provide more context to the conversation.

The task of asking clarifying questions is challenging and requires a deep understanding of language and context. It involves two subtasks: determining whether a clarifying question is needed for a given instruction, and ranking a list of possible clarifying questions based on their relevance. Solving these subtasks can improve the performance and naturalness of open-domain dialogue systems.

The main objective of this project is to develop models that can effectively ask clarifying questions in open-domain dialogue systems. The models are evaluated using macro average F1 score for the binary classification subtask and Mean Reciprocal Rank (MRR) for the ranking subtask. The dataset and starter code provided on GitHub are used to train and evaluate the models.

The results of this project have the potential to significantly improve the usability and effectiveness of open-domain dialogue systems, making them more natural and intuitive for users. This project can also contribute to advancing the field of natural language processing by developing models that can better understand and respond to human language.

2 Related Work

The task of asking clarifying questions in dialogue systems has been explored in several previous works. These works have used different approaches and techniques to address the challenges involved in this task.

One approach is to use rule-based methods to generate clarifying questions. For example,(O'Donnell et al., 2018) proposed a rule-based method to generate clarification questions for spoken dialogue systems. They used a set of rules based on syntactic and semantic features of the input to generate appropriate clarifying questions.

Another approach is to use machine learning techniques to learn to generate clarifying questions. For example, (Du et al., 2017) proposed a neural network-based method to generate clarification questions for dialogue systems. They used a combination of convolutional and recurrent neural networks to generate relevant questions based on the input context.

In addition to generating clarifying questions, several works have also explored the task of selecting relevant clarifying questions from a set of possible questions. For example, (Aliannejadi et al., 2019) proposed a method to rank a set of clarification questions based on their relevance to the input context. They used a combination of neural network-based models and feature-based models to rank the questions.

topic_id	initial_request	clarification need	question_id	question	answer
14	I'm interested in dinosaurs	4	Q00173	are you interested in coloring books	no i just want to find the discovery channels website
14	I'm interested in dinosaurs	4	Q03021	which dinosaurs are you interested in	im not asking for that i just want to go to the discovery channel dinosaur page

Table 1: Example of the ClariQ dataset

Overall, these works have demonstrated the feasibility and potential benefits of asking clarifying questions in dialogue systems. However, there is still room for improvement in terms of the accuracy and effectiveness of these methods.

3 Methodology

3.1 Dataset

The ClariQ dataset is a valuable resource for training and evaluating models for the task of asking clarifying questions in open-domain dialogue systems. The dataset builds upon the Qulac dataset and includes additional topics, questions, and answers in the training set.

Overall, the ClariQ dataset is a comprehensive and diverse resource for training and evaluating models for the task of asking clarifying questions. With approximately 1.8 million multi-turn conversations and 18,000 single-turn conversations, the dataset provides a rich source of data for developing models that can effectively ask clarifying questions in open-domain dialogue systems.

The ClariQ dataset consists of two main files, `train.tsv` and `dev.tsv`, which have the same format. Each row in these files contains information about a topic, facet, question, answer, and clarification need label(Ex shown in Table 1)

- The `topic_id` field contains the ID of the initial request query that initiates the conversation, while the `topic_desc` field provides a full description of the topic. The `clarification_need` field contains a label from 1 to 4, indicating the level of clarification needed for the initial request query.
- The `facet_id` and `facet_desc` fields provide information about the specific information need or facet of the topic, while the `question_id` field contains the ID of the clarifying question as it appears in `question_bank.tsv`. The `question` field contains the actual clarifying question, and the `answer` field provides an answer to the question, assuming that the user is in the context of the current row.
- The `train.tsv` and `dev.tsv` files also include labels for the clarification need of the initial request query. These labels range from 1 to 4, with 1 indicating that no clarification is needed and 4 indicating that a significant amount of clarification is needed.

Overall, the ClariQ dataset provides a comprehensive set of labeled data for the task of asking clarifying questions in open-domain dialogue systems, making it a valuable resource for researchers and practitioners working in the field of natural language processing.

3.2 Approach

In this project, we first performed an extensive exploration of the ClariQ dataset. We examined the distribution of the different labels and analyzed the patterns in the data to gain a better understanding of the task.

Next, we implemented a baseline model using BM25 and BERT(Devlin et al., 2019), and trained it on the ClariQ training dataset. We used the macro average F1 score for the binary classification task and the MRR for the ranking task to evaluate the performance of the model on the validation set.

After that, we experimented with more complex neural networks architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models such as BERT, RoBERTa(Liu et al., 2019),ALBERT (Lan et al., 2019) and Electra(Clark et al., 2020). We fine-tuned these models on the ClariQ training dataset and evaluated their performance on the validation set. We also explored different hyperparameters such as learning rates, batch sizes, and dropout rates to improve the performance of the models.

Furthermore, we tried combining the classifier and ranker models to improve the performance of the system. We experimented with different combinations of models, such as using the output of the classifier as input to the ranker or concatenating the output of the classifier and ranker before making a final decision.

Model	Recall@5	Recall@10	Recall@20	**Recall@30**
Bert - Reranker	0.348	0.618	0.691	0.691
Bert - Ranker	0.352	0.623	0.729	**0.758**
ALBERT - Reranker	0.346	0.609	0.690	0.691
ALBERT - Ranker	0.344	0.599	0.695	0.718
Roberta - Reranker	0.329	0.584	0.690	0.691
Roberta - Ranker	0.313	0.550	0.693	0.717
Electra - Reranker	0.339	0.587	0.690	0.691
Electra - Ranker	0.318	0.507	0.607	0.632

Table 2: Recall scores at different ranks for different models.

Model	Epoch	Accuracy (dev)	Accuracy (test)	Precision	Recall	F1
BERT	10	0.54	0.426	0.392	0.426	0.382
ALBERT	10	0.48	0.607	0.497	0.607	**0.545**
RoBERTa	10	0.46	0.443	0.369	0.443	0.399
Electra	10	0.50	0.443	0.441	0.443	0.427

Table 3: Accuracy, precision, recall and F1 score of different models

Finally, we compared the results of our models with the leaderboard and submitted our best-performing model as the final submission. Overall, our approach was to experiment with different models and combinations of models to achieve the best performance on the ClariQ dataset.

4 Results and Analysis

In our project, we experimented with different models for both ranking and classification tasks, the results of which are shown in Table 2 and in Table 4. For the ranking task, we used both ranker and reranker approaches. The reranker approach involved first ranking all the answers for each query using the BM25 algorithm, and then re-ranking the top 30 answers using a trained model. On the other hand, the ranker approach involved ranking all the answers using the trained model and selecting the top 30 answers.

We tested various models for the reranker and ranker approaches, such as BM25, TF-IDF, and BERT. Our baseline results matched those provided in the leaderboard. However, we found that the other models performed worse than the baseline, possibly due to a lack of training. These larger models require more epochs to train and a larger dataset, and since the dev dataset had very few unique queries (about 78), the performance of these models suffered.

For the classification task, we tested multiple models Table 3, including BERT, RoBERTa, and Albert. We found that Albert performed the best, ranking second on the leaderboard.

Finally, we tried combining the classifier and ranker models to see if it would improve performance. However, from the results shown in Table 5 we found that there were very few queries in the dev set with no clarification need, and in the test set, the number of such queries was also very limited. As a result, we did not observe much difference in performance when using a combined approach.

Overall, we explored multiple models for both ranking and classification tasks and found that Albert performed the best for classification, while the baseline model performed the best for ranking. We believe that with more training data and more epochs, the larger models we tested could potentially outperform the baseline model.

5 Conclusion

The ClariQ dataset was a challenging task for our team, as the size of the dev dataset was quite small, and this made it difficult to train large models effectively. Our experiments showed that the baseline results were quite competitive, but further improvements required a careful combination of classifier and ranker models.

One of the key observations we made was that the combination of classifier and ranker models gave us better results than using either model in isolation. This is because the classifier model was effective in identifying the nature of the clarification needed for a particular query, while the ranker model was effective in ranking the candidate an-

Ranker/Reranker	Tokenizer Model	Ranker Model	Recall@5	Recall@10	**Recall@20**	Recall@30
Re-Ranker	albert-base-v2	albert-base-v2	0.303	0.546	0.754	0.768
Ranker	albert-base-v2	albert-base-v2	0.131	0.314	0.635	0.768
Re-Ranker	albert-base-v2	bert-base-uncased	0.301	0.540	0.717	0.768
Ranker	albert-base-v2	bert-base-uncased	0.168	0.319	0.610	0.768
Re-Ranker	albert-base-v2	electra	0.168	0.287	0.527	0.768
Ranker	albert-base-v2	electra	0.124	0.217	0.453	0.768
Re-Ranker	electra	albert-base-v2	0.141	0.278	0.538	0.768
Ranker	electra	albert-base-v2	0.068	0.221	0.575	0.768
Re-Ranker	electra	bert-base-uncased	0.333	0.599	**0.759**	0.768
Ranker	electra	bert-base-uncased	0.112	0.295	0.647	0.768

Table 4: Results using different models without a classification model

Ranker/Reranker	Classification Model	Ranker Model	Recall@5	Recall@10	**Recall@20**	Recall@30
Re-Ranker	albert-base-v2	albert-base-v2	0.3017	0.5449	0.7543	0.7682
Ranker	albert-base-v2	albert-base-v2	0.1298	0.3127	0.6342	0.7682
Re-Ranker	albert-base-v2	bert-base-uncased	0.3000	0.5390	0.7167	0.7682
Ranker	albert-base-v2	bert-base-uncased	0.1664	0.3179	0.6101	0.7682
Re-Ranker	albert-base-v2	electra	0.1684	0.2874	0.5271	0.7669
Ranker	albert-base-v2	electra	0.1229	0.2169	0.4521	0.7669
Re-Ranker	electra	albert-base-v2	0.1408	0.2780	0.5385	0.7659
Ranker	electra	albert-base-v2	0.0681	0.2189	0.5713	0.7646
Re-Ranker	electra	bert-base-uncased	0.3294	0.5967	**0.7592**	0.7682
Ranker	electra	bert-base-uncased	0.1095	0.2948	0.6434	0.7682
Re-Ranker	electra	electra	0.2962	0.5501	0.7550	0.7682
Ranker	electra	electra	0.1007	0.3250	0.5348	0.7682

Table 5: Results using different models without a classification model

swers based on their relevance to the query. By combining these two models, we were able to improve the performance of our system by a small margin.

However, we also realized that a better implementation idea would be to make use of the documents from which the question and answers are taken, as they have the most relevant context for the task. This is because the answers to a given query may be highly dependent on the context in which they are presented, and by incorporating this context, we could potentially improve the performance of our system significantly. Unfortunately, this approach would have been too computationally and time-intensive for the scope of this project.

In summary, our experiments on the ClariQ dataset highlighted the importance of careful model selection and combination, as well as the potential benefits of incorporating contextual information into our system. However, given the limitations of the dev dataset and the scope of this project, we were only able to make modest improvements over the baseline results.

Limitations

1. As a team, we faced some limitations in terms of resources such as time, compute power, and access to larger datasets, which may have affected the scope of our experiments and results.

2. We also encountered some technical challenges such as model optimization, training, and debugging, which required extensive troubleshooting and may have impacted the efficiency of our workflow.

3. In addition, we acknowledge that our experiments may have some limitations in terms of generalization, as our analysis is based on a specific dataset and evaluation metric, and may not reflect the performance of our models in other contexts or scenarios.

4. Lastly, we recognize that there may be ethical

considerations and implications of our work, particularly in the area of natural language processing and information retrieval, which we have taken into account in our project design and implementation.

References

Mohammad Aliannejadi, Yang Gao, and Yejin Zhang. 2019. Clariq: A new dataset for clarification question generation. 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 2787–2796.

Kevin Clark, Minh-Thang Luong, and Quoc V Le. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lifu Du, Xinyuan Huang, Wei Wang, and Zhihong Zhang. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1342–1352.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. 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 –.

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. 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 – .

Timothy J O'Donnell, Robert Birke, and Roberto Pieraccini. 2018. Automatic clarification question generation for spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 149–158.