Can Large Language Models Be a Good Evaluator for Review-based Product Question Answering?

Tony Danhui Huang
School of Computing Technologies
RMIT University
Melbourne, Victoria, Australia
tony.huang2@student.rmit.edu.au

Yongli Ren
School of Computing Technologies
RMIT University
Melbourne, Victoria, Australia
yongli.ren@rmit.edu.au

Xiuzhen Zhang
School of Computing Technologies
RMIT University
Melbourne, Victoria, Australia
xiuzhen.zhang@rmit.edu.au

Abstract

Large language models (LLMs) have demonstrated impressive performance on evaluating natural language generation tasks. Review-based Product Question Answering (RPQA) evaluation, which is a domain knowledge-intensive Question Answering (QA) evaluation, still largely relies on lexicon-based metrics and frozen embedding-based metrics. Those metrics fail if reference answers are absent. Despite some model-based metrics being learned from in-domain data for the RPQA task evaluation, little research has been done on using LLMs. Chain-of-Thought (CoT) is a state-of-the-art prompting method that has been proposed to induce LLMs to solve complex problems efficiently. The CoT reasoning steps can be resolved using the internal knowledge of LLMs, but this internal knowledge may be insufficient in domain knowledge-intensive RPQA evaluation settings. In this work, we explore the feasibility of leveraging LLMs as evaluators for RPQA evaluation. Specifically, we design a structured prompt template with one-shot or few-shot in-context learning, incorporating the CoT mechanism and the System 2 thinking process for RPQA evaluation. We call it as **E**xamp**L**e-**E**nhanced **C**o**T** prompting (**ELECT**). We conduct a comprehensive study on various LLMs, including evaluation-oriented and general-purpose LLMs. Experimental results show that LLM-based evaluators can effectively evaluate RPQA. Upon integrating ELECT prompting into the demonstration examples in the prompt, we observe an improvement in the performance of the LLM-based evaluator. Code is publicly available at https://github.com/tonydhuang/elect-prompting.

CCS Concepts

• **Computing methodologies** → **Natural language generation**.

Keywords

Product question answering evaluation, LLM-based evaluation, Evaluation, System 2 thinking process, Chain-of-Thought

ACM Reference Format:

Tony Danhui Huang, Yongli Ren, and Xiuzhen Zhang. 2025. Can Large Language Models Be a Good Evaluator for Review-based Product Question Answering?. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3715586

WWW Companion '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/25/04
https://doi.org/10.1145/3701716.3715586

1 Introduction

LLMs have demonstrated incredible performance on various NLG task evaluation, such as open-domain QA [8, 17] and text summarisation [13, 18]. Specifically, in-context learning has emerged as a new paradigm for many NLG tasks [2, 10]. It provides an interpretive interface for communicating with LLMs and, more importantly, offers built-in knowledge through a few demonstration examples, allowing LLMs to learn without updating their parameters [10]. Different prompting strategies are employed in various evaluation tasks. Chain-of-Thought (CoT) is one of the popular prompting strategies, playing an important role in the evaluation of specific tasks [13]. CoT prompting is a sequence of intermediate reasoning steps that break down a complex problem into a set of sub-problems, resolving them one by one before generating the final output. For instance, G-Eval [13], a reference free LLM-based metric, performs well for open-domain QA evaluation using CoT reasoning steps. One of the reasons behind this is that LLMs can make correct judgments for open-domain QA using their internal knowledge learned during the pre-training or fine-tuning stage. However, the RPQA system requires access to domain-specific data, such as rich product information and product reviews that address users' opinions on specific products, when generating answers to product-related questions. RPQA is thus a domain knowledge-intensive task.

RPQA [7, 14] evaluation is still largely relying on text similarity [12] or contextual embedding-based similarity [22]. Traditional metrics cannot perform NLG task evaluation when reference text is not available. Some recent works propose reference-free evaluation metrics by training end-to-end neural models on in-domain labeled data. However, these learned metrics can only evaluate in-domain specific tasks, such as text summarization evaluation - LS_Score [20]. The work closest to ours is the learned metric with small model size - DGML [6]. Though the DGML metric shows promising performance, its correlation with human judgments is still very low. This motivates us to investigate the ability of LLMs to serve as evaluators for domain knowledge-intensive RPQA evaluation, an area that has seen limited research in the literature.

In this paper, we propose Example-Enhanced CoT prompting (ELECT) to improve CoT prompting by providing a few demonstration examples of question-answer pairs and an explanation of the reasoning process used to reach their rating scores. Our approach is inspired by the System 2 thinking process in dual-process theories [4]. The difference between the two types of thinking processes in dual-process theories is that the response of System 1 thinking is intuitive, quick, and based on implicit knowledge, whereas System 2 thinking is reflective and slow, involving hypothetical thinking and reasoning, and requiring explicit knowledge to solve complex

problems and emotions. System 2 thinking process has been used to induce LLMs for logical thinking on solving complex problem, such as Chain-of-Thought (CoT) [19]. We adopt the System 2 thinking process in our approach. One reason is that the answer generated by the RPQA system is opinionated, which means that it inherits sentiments from product reviews written by humans. Another reason is that the RPQA system is domain knowledge-intensive, and requires explicit knowledge. Both of these features are supported by the System 2 thinking process [4]. We design a structured prompt template using one-shot or few-shot in-context learning, where example-enhanced CoT with explainable rating scores is applied to demonstrations using the System 2 thinking process. We then use this prompt to ask LLMs to rate how well the model-generated answer addresses the question, using a 4-point Likert scale.

Specifically, we aim to address the following research questions:

- RQ1: Can LLM-based metrics outperform traditional metrics in RPQA setting?
- RQ2: How does ELECT prompting help with LLM-based RPQA evaluation?

We comprehensively examine the use of various LLMs for RPQA evaluation, including general-purpose LLMs [1, 3] and evaluation-oriented LLMs [11]. We experimentally compare the performance of LLM-based evaluators with the traditional metrics, such as BERTScore [22] and DGML[6]. Experiment results show that LLMs can be good evaluators for RPQA. We further investigate the effectiveness of our proposed ELECT prompting method compared to various other prompting methods. We found that ELECT prompting is more efficient than general demonstration prompting and general CoT reasoning. To the best of our knowledge, we are the first in investigating the feasibility of using LLMs for RPQA evaluation.

2 Methodology

In this work, we investigate the capability of reference free LLM-based evaluators for RPQA evaluation. Specifically, we design a structured prompt template (Section 2.1) that integrates the innovative adaptation of CoT reasoning in demonstration examples using System 2 thinking process for RPQA evaluation (Section 2.2). Furthermore, we use various LLMs as the backbone models for RPQA evaluation (Section 2.3).

2.1 Structured prompt template

We take inspiration from G-Eval [13] to create a general form-filling prompting template including a standard **task instruction** together with the desired output format, **evaluation criteria** and **inputs**. Here we briefly describe those components that are used in our prompt template.

Task instruction. We define the role of the evaluation task followed by a task description together with the desired output format:

> ***Role***: *You are a product question answering task evaluator.*
> ***Task description***: *You will be given a product question, and a model generated answer. Your task is to rate how well the model generated answer can answer the question based on the evaluation criteria.*
> ***Desired output format***: *Please do not generate any opening, closing, and explanations. The response should be one word with a number between 0 and 3 only.*

Evaluation criteria. We design an evaluation criteria specifically for RPQA setting as to assess the quality of the model-generated answer given a product question:

> **Evaluation Criteria: Rating Score (0-3)**
> ***0***: *the answer was irrelevant to the question;*
> ***1***: *the answer was related but didn't answer the question;*
> ***2***: *the answer somehow answered or partially answered the question;*
> ***3***: *the answer directly answer the question.*

To ensure a fair comparison in our setting, we use score-based evaluation criteria, which are the same as the rating score definitions we provided for human annotation in Section 3.3.

Inputs. Inputs include a given question and model-generated answer pair.

2.2 ELECT prompting.

The traditional CoT method usually breaks down complex problems into sub-problems or creates intermediate steps in line with the internal knowledge learned by the LLM. Specifically, we ask ChatGPT to create CoT reasoning steps for specific question-answer pairs based on the provided task instructions and evaluation criteria. We then refine the generated CoT reasoning steps and analytical results to add more detail and improve clarity using the System 2 thinking process. We present a real demonstration example of the ELECT prompting method as follows.

> ***Product Question***: *Is this difficult to clean?*
> ***model-generated Answer***: *it can be a little tedious to clean.*
> ***Chain of Thought***:
> *1. Understanding the Product Question*
> *The question asks, "Is this difficult to clean?" The user is seeking information about the ease or difficulty of cleaning the product. The expected answer should clearly address whether the product is easy or difficult to clean.*
> *2. Analyzing the model-generated Answer*
> *The model-generated answer says, "it can be a little tedious to clean."*
> *The model-generated answer indicates that cleaning the product may not be easy and can take some effort, which aligns with the user's inquiry about difficulty.*
> *The phrase 'a little tedious' implies some difficulty, though not extreme, and provides a direct answer to the question.*
> *3. Rating the model-generated Answer Based on the Evaluation Criteria*
> *Score 3: The model-generated answer directly answers the question by explaining that cleaning the product is somewhat difficult or tedious. It addresses the key aspect of the user's question.*
> *4. Conclusion*
> *The model-generated answer is relevant, clear, and directly addresses the question about difficulty in cleaning.*
> *Rating: 3 (Directly answers the question).*

In Step 2 of the 'Chain of Thought' in the above example, the originally generated analytical result was:

> *2. Analyzing the model-generated Answer*

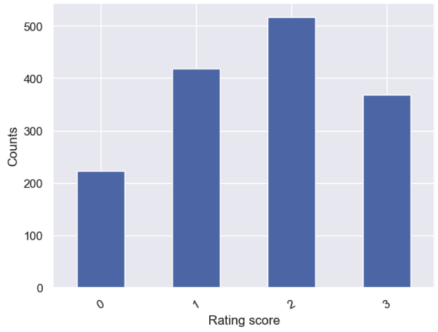

Figure 1: Rating score distribution of annotated data.

> *The model-generated answer says, "it can be a little tedious to clean."*
> *The phrase "a little tedious" suggests that cleaning is not extremely difficult but requires some effort.*

We then refine it with a more detailed thinking process, as shown in the above real demonstration example in orange. This is useful for the domain knowledge-intensive RPQA setting as the LLM can learn the domain-specific knowledge and latent patterns from in-context examples, and elicit its reasoning ability for RPQA evaluation.

2.3 Backbone models

We assess the performance of three backbone LLMs for RPQA evaluation - Prometheus [11], Llama [3] and GPT-4 [1]. **Prometheus** is a powerful evaluation-oriented open source LLM that is employed specifically to assess the quality of generated text by other language models. We use 'prometheus2-7b' model in our experiments. **Llama** is another popular general purpose open source LLM. We use 'llama3-8b' model in our experiments. **GPT-4** is a general purpose proprietary LLM. We use the GPT-4 version - 'gpt-4o' in our experiments.

3 Experiments and results

We adopt two widely used correlation measures: Spearman [21] coefficient ρ and Pearson [15] coefficient r in all our experiments. Spearman measures the monotonic relationships between two continuous variables without assuming linearity. Pearson measures the linear relationships between two continuous variables.

3.1 Datasets

Following the existing work - DGML [6], we use publicly available Amazon Community QA datasets (CQA) and review datasets[1] published by the existing work [14] for our evaluation tasks. We collect 1,800 questions in total from Amazon CQA dataset for 10 product categories with 180 questions for each product category.

We use BM25 [16] to compute and rank the relevant reviews against each question, and then pick top 100 relevant reviews for each question.

[1]https://cseweb.ucsd.edu/ jmcauley/datasets.html

3.2 RPQA model

UnifiedQA is a publicly available QA model that has been trained and evaluated on different QA formats (e.g., abstractive, yes/no, etc.) and publicly available datasets [9]. UnifiedQA shows strong generalization on unseen datasets in its experiments. We adopt UnifiedQA as our RPQA model to generate answers for CQA questions based on the top 100 ranked relevant reviews.

3.3 Human annotation

We conduct human annotation on CQA pairs, where the answer for each QA pair is generated by the UnifiedQA model, given the top 100 ranked reviews. We hire experienced crowd workers from the Amazon Mechanical Turk platform[2] to rate the model-generated answer on a scale of {0,1,2,3}, based on how well it answers the question, given three reference answers from the Amazon CQA dataset. We follow the same quality control methods and reliability measures for the crowd workers that were used in DGML [5, 6]. Specifically, we compute the Pearson correlation between workers' annotations and then remove annotations associated with workers who have a lower Pearson correlation score (inter-rater agreement score < 0.4). We also remove questions that have only one remaining annotation and those with two conflicting annotated scores (e.g., one rated as 0 and the other rated as 3). This results in 274 questions being eliminated from the test set. An average inter-rater agreement score of 0.76 is achieved. We use the rating scores of the remaining 1,526 questions as our human judgments. We plot the annotated rating score distribution in Figure 1.

3.4 RQ1: Can LLM-based metrics outperform traditional metrics in RPQA setting?

We construct a standard prompt template that includes task instructions, evaluation criteria, and a question-answer pair for evaluation. We call it as direct prompt (DP), without incorporating CoT and demonstrations into the prompt. We present a real direct prompt example as follows.

> *You are a product question answering task evaluator.*
> *You will be given a product question, and a model generated answer. Your task is to rate how well the model generated answer can answer the question based on the evaluation criteria. Please do not generate any opening, closing, and explanations. The response should be one word with a number between 0 and 3 only.*
> ***Evaluation Criteria: Rating Score (0-3)***
> *0: the answer was irrelevant to the question;*
> *1: the answer was related but didn't answer the question;*
> *2: the answer somehow answered or partially answered the question;*
> *3: the answer directly answer the question.*
> ***Product Question:***
> *I know that every review says they're great for close-up viewing, but how about far distance?*
> ***Model Generated Answer:***
> *for distance viewing, you can do much better.*

We then prompt the LLM to generate metric scores for the model-generated answers and compute the correlation between human judgments and LLM-generated scores. For the traditional metrics,

[2]www.mturk.com

Table 1: Results of correlation scores between human judgments and various evaluation metrics including LLM-based metrics and traditional metrics.

Product Category	GPT-4o		Prometheus		Llama		DGML		BERTScore	
	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r
Automotive	**0.698**	0.654	0.633	0.600	0.531	0.544	0.596	0.500	0.509	0.529
Cell phones	**0.747**	0.711	0.636	0.632	0.537	0.531	0.617	0.418	0.496	0.509
Electronics	**0.716**	0.729	0.609	0.617	0.497	0.529	0.540	0.471	0.527	0.578
Patio lawn and garden	**0.719**	0. 674	0.667	0.654	0.591	0.570	0.599	0.522	0.403	0.391
Office products	**0.837**	0.786	0.736	0.729	0.631	0.616	0.668	0.459	0.541	0.567
Pet supplies	0.649	0.621	0.653	0.638	0.612	0.606	**0.661**	0.530	0.429	0.479
Average	**0.728**	0.696	0.656	0.645	0.567	0.566	0.614	0.483	0.484	0.509

we follow these steps to generate metric scores for the same test sets. First, we input question and model-generated answer pairs into the DGML metric to generate the metric scores. Second, we input the question and its three reference answers from the CQA dataset into BERTScore, to compute metric scores, then average the three scores for each question to obtain the final metric score. Lastly, we compute the correlation between human judgments and the metric scores.

We report the experiment results of six product categories in Table 1. We observe that the general-purpose proprietary LLM, GPT-4o, statistically significantly achieves the best performance in five out of six product categories, while the evaluation-oriented LLM, Prometheus, ranks second. We also observe that the best traditional learned metric - DGML averagely outperformed Llama in terms of Spearman's ρ correlation. As the differences across product categories are not significant, Llama has the similar performance with DGML. However, Llama still statistically significantly outperforms DGML with 18% of relative difference in Pearson's r.

The overall results show that LLM-based metrics can be good evaluators in RPQA setting. LLM-based metrics show strong alignment with human adjustments. This indicates that the capability of LLM-based evaluator is not only distinguishing between good and bad model-generated answers, but also measuring the quality level of answers and reflecting the absolute differences in the scores.

3.5 RQ2: How does ELECT prompting help with LLM-based RPQA evaluation?

In this research question, we examine whether ELECT prompting could help with improving performance of LLMs. For the cost control purposes, we use an open-source LLM instead of a proprietary LLM in our experiments. As Prometheus runs slower than Llama, we prefer Llama over Prometheus as the base model for this research question. We construct the demonstration examples using our proposed ELECT prompting method as described in Section 2.2, and insert into the prompt. We use this prompt to ask Llama to score the model-generated answers for the same test set, and then compute and report the correlation between human judgments and the generated metric scores. For G-Eval method, we prompt ChatGPT to generate CoT steps given a task instruction, along with evaluation criteria as described in Section 2.1. The general CoT steps without example enhancement are generated as follows:

> ***1. Understanding the Product Question***
> *Clearly define the product-related question being asked. Identify any specific details or requirements the question is asking for (e.g., features, specifications, price, availability).*
> ***2. Analyzing the model-generated Answer***
> *Check whether the model-generated answer addresses the key aspects of the question.*
> *Determine if the model-generated answer provides the necessary information (or if something is missing).*
> *Verify if the model-generated answer is directly related to the product in question.*
> *Evaluate if the model-generated answer includes irrelevant or extraneous information.*
> ***3. Rating the model-generated Answer Based on the Evaluation Criteria***
> ***Score 0**: If the model-generated answer is entirely irrelevant or doesn't address any part of the question.*
> ***Score 1**: If the model-generated answer is somewhat related to the question, but it doesn't answer it.*
> ***Score 2**: If the model-generated answer partially answers the question but is incomplete or lacks clarity.*
> ***Score 3**: If the model-generated answer fully and directly answers the question with clear and relevant information.*
> ***4. Conclusion***
> *Based on the analysis, assign a score (0–3) to the model's answer.*

We use the same direct prompt (DP) as describe in RQ1. We insert a simple demonstration without example-enhanced CoT into DP prompt - 'One-shot DP' to examine whether it could help improve the performance without example-enhanced CoT. We present a simple demonstration used in 'One-shot DP' as follows.

> *Product Question: Is this difficult to clean?*
> *Model Generated Answer: it can be a little tedious to clean.*
> *Rating Score: 3*

In order to compare ELECT prompting against these three prompting strategies, we present them as input and ask Llama to score the model-generated answers for the same test set. We then compute and report the correlation between human judgments and the generated metric scores. As shown in Table 2, the 'ELECT' column shows statistically higher Spearman's ρ (numbers in bold) and Pearson's r (numbers with underscores) correlations than other three prompting strategies ('G-Eval', 'DP' and 'One-shot DP') in most of the product categories. By comparing 'ELECT' to 'G-Eval' with general CoT prompting, we observe that some product categories, 'Electronics', 'Patio Lawn and Garden' and 'Toys and Games', achieve more than a 10% relative difference in terms of both Spearman's ρ and Pearson's r correlations. This indicates that Example-enhanced CoT (ELECT) prompting method can help Llama understand more on the demonstration context and latent thinking process, and it further guides the Llama to generate more reliable metric scores aligned with human judgments.

We also compare 'ELECT' to two prompting strategies ('DP' and 'One-shot DP') without CoT applied, which are on the right side of the separator (||) in Table 2. Observe that some other categories - 'Home and Kitchen' and 'Patio Lawn and Garden' achieve large improvements, with relative differences of 37% and 24% respectively. However, there are not many differences among the prompting methods for the 'Cell Phones' category in terms of correlation

Table 2: Results of correlation scores between human judgments and various prompting strategies for Llama. 'G-Eval': direct prompt with general CoT; 'DP': direct prompt without CoT and demonstration; 'One-shot DP': direct prompt with one demonstration without example-enhanced CoT.

Product Category	ELECT		G-Eval		DP		One-shot DP	
	ρ	r	ρ	r	ρ	r	ρ	r
Automotive	**0.607**	0.589	0.549	0.556	0.531	0.544	0.574	0.552
Cell phones	0.539	0.548	0.527	0.512	0.537	0.531	**0.543**	0.521
Electronics	**0.614**	0.603	0.546	0.534	0.501	0.537	0.521	0.537
Patio lawn and garden	**0.682**	0.665	0.574	0.559	0.591	0.570	0.583	0.537
Home and kitchen	**0.484**	0.445	0.475	0.445	0.456	0.413	0.407	0.325
Office products	**0.661**	0.640	0.633	0.611	0.631	0.616	0.613	0.598
Pet supplies	**0.648**	0.643	0.604	0.592	0.612	0.606	0.635	0.615
Sports and outdoors	**0.676**	0.649	0.632	0.582	0.594	0.577	0.632	0.606
Tools and home impr.	**0.642**	0.636	0.599	0.581	0.592	0.595	0.619	0.614
Toys and games	**0.661**	0.630	0.575	0.545	0.581	0.563	0.624	0.595
Average	**0.621**	0.605	0.571	0.552	0.563	0.555	0.575	0.550

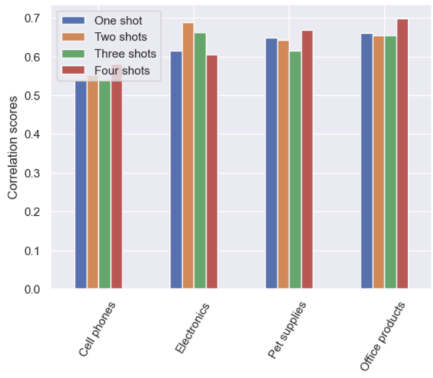

Figure 2: scores distribution of 'Automotive' product category with one-shot, two-shot, three-shot and four- shot ELECT demonstration for Llama.

scores. For example, 'ELECT' and 'One-shot DP' exhibit similar performance for the 'Cell Phones' category in terms of Spearman's ρ correlation. The reason behind this could be that products from the 'Cell Phones' category are well-known in the community and product knowledge may have been learned by LLMs during the pre-training or fine-tuning stage. We also observe that the differences among the 'G-Eval', 'DP' and 'One-shot DP' columns are not significant, indicating that they exhibit similar performance for all product categories.

We further investigate the impact of the number of examples provided with ELECT in the demonstration. As shown in Figure 2, the few-shot demonstration does not result in a more significant improvement than the one-shot demonstration. It indicates that Llama can learn from one-shot example-enhanced CoT demonstration effectively.

4 Conclusion

In this work, we experimentally study the capability of LLMs as evaluators for RPQA evaluation. In particular, we design an example-enhanced CoT (ELECT) prompting method that can be used in one-shot or few-shot in-context learning, where examples of demonstration are adapted with CoT reasoning steps using the System 2 thinking process. Our experimental results show that LLMs can be effective evaluators for RPQA. In addition, we find that the general CoT prompt strategy does not always improve performance in RPQA evaluation. However, our proposed ELECT prompting method, by integrating concrete CoT for each demonstration example in the prompt, can significantly improve the performance of LLM-based RPQA evaluation.

Acknowledgments

This research is supported in part by the Australian Research Council Discovery Project DP200101441.

References

[1] Josh Achiam et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
[2] Brown et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
[3] Hugo Touvron et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL]
[4] Jonathan St BT Evans and Keith E Stanovich. 2013. Dual-process theories of higher cognition: Advancing the debate. *Perspectives on psychological science* 8, 3 (2013), 223–241.
[5] Yvette Graham et al. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering* (2017).
[6] Tony Danhui Huang, Yongli Ren, Lifang Wu, and Xiuzhen Zhang. 2024. Reference-free review-based product question answering evaluation via distant contrastive learning. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
[7] Tony Danhui Huang, Xiuzhen Zhang, Youngli Ren, and Min Peng. 2020. Can helpful reviews help answer product questions?. In *PACIS 2020 Proceedings. 4.*
[8] Ehsan Kamalloo et al. 2023. Evaluating Open-Domain Question Answering in the Era of Large Language Models. *arXiv preprint arXiv:2305.06984* (2023).
[9] Daniel Khashabi et al. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700* (2020).
[10] Omar Khattab et al. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024* (2022).
[11] Seungone Kim et al. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535* (2024).
[12] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out.*
[13] Yang Liu et al. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634* (2023).
[14] Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web.*
[15] Mavuto M Mukaka. 2012. A guide to appropriate use of correlation coefficient in medical research. *Malawi medical journal* (2012).
[16] Stephen E Robertson et al. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94.*
[17] Cunxiang Wang et al. 2024. Evaluating open-qa evaluation. *Advances in Neural Information Processing Systems* (2024).
[18] Jiaan Wang et al. 2023. Is ChatGPT a Good NLG Evaluator? A Preliminary Study. In *Proceedings of the 4th New Frontiers in Summarization Workshop.*
[19] Jason Wei et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* (2022).
[20] Hanlu Wu et al. 2020. Unsupervised reference-free summary quality evaluation via contrastive learning. *arXiv preprint arXiv:2010.01781* (2020).
[21] Jerrold H Zar. 2005. Spearman rank correlation. *Encyclopedia of biostatistics* (2005).
[22] Tianyi Zhang et al. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).