Understanding Large Language Model Based Metrics for Text Summarization

Abhishek Pradhan[*#]
abhishek.pradhan2008@gmail.com

Ketan Kumar Todi[*+]
todiketan@hotmail.com

Abstract

This paper compares the two most widely used techniques for evaluating generative tasks with large language models (LLMs): prompt-based evaluation and log-likelihood evaluation as part of the Eval4NLP shared task. We focus on the summarization task and evaluate both small and large LLM models. We also study the impact of LLAMA and LLAMA 2 on summarization, using the same set of prompts and techniques. We used the Eval4NLP dataset for our comparison. This study provides evidence of the advantages of prompt-based evaluation techniques over log-likelihood based techniques, especially for large models and models with better reasoning power.

1 Introduction

Transformer-based language models have revolutionized the field of natural language processing (NLP), particularly in the area of language generation. However, the improved language generation capabilities of these models have also exposed the limitations of traditional lexical evaluation metrics, such as perplexity, BLEU (Papineni et al., 2002), and ROUGE (Lin, 2004). These metrics are often unable to accurately assess the quality of generated text, especially when it is creative or informative.

In response, researchers have developed a wide range of new automatic evaluation models, such as BLEURT (Sellam et al., 2020), BERTScore (Zhang et al., 2020), and BARTScore (Yuan et al., 2021). These models typically rely on a combination of lexical and semantic features to assess the quality of generated text, and some of them also take into account the golden reference annotation.

Recent large language models (LLMs) like PaLM (pal, 2022), GPT-3.5, and GPT-4 (OpenAI, 2023) have taken language generation capabilities to a new level, making it difficult to distinguish between machine-generated and human-written text. This has led to the use of LLMs for a variety of more complex tasks, such as summarizing entire research papers, even when the ground truth is not known. The increased complexity of these tasks has spurred interest in using LLMs themselves for model evaluation.

Prompt-based and log-likelihood-based evaluation are two widely used approaches for automatic evaluation of large language models (LLMs). However, it is unclear which approach works better with different model sizes, as previous studies have used these approaches on mutually exclusive sets of models.

In this paper, we evaluate multiple LLM models of different sizes using both prompt-based and log-likelihood-based evaluation on the Eval4NLP dataset (Leiter et al., 2023) as part of the Eval4NLP shared task (Leiter et al., 2023). We experiment with three models from the LLaMA (Touvron et al., 2023a) and LLaMA2 (Touvron et al., 2023b) family, which are allowed in the Eval4NLP 2023 shared task.

Our results show that prompt-based evaluation generally outperforms log-likelihood-based evaluation for all model sizes. This is likely because prompt-based evaluation is more directly aligned with the tasks that LLMs are typically used for, such as generating text, translating languages, and answering questions.

Our findings suggest that prompt-based evaluation is a more reliable and informative approach for evaluating LLMs of all sizes.

2 Dataset and Task Description

The summarization track of the Eval4NLP task involved predicting an overall score for a model-generated summary of a source text. The competition required participants to use only a limited set of models without fine-tuning, meaning that the

[*]These authors contributed equally to this work.
[+]Work done while author was at Google.
[#]Work done while author was at Incivus.

Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems, pages 149–155

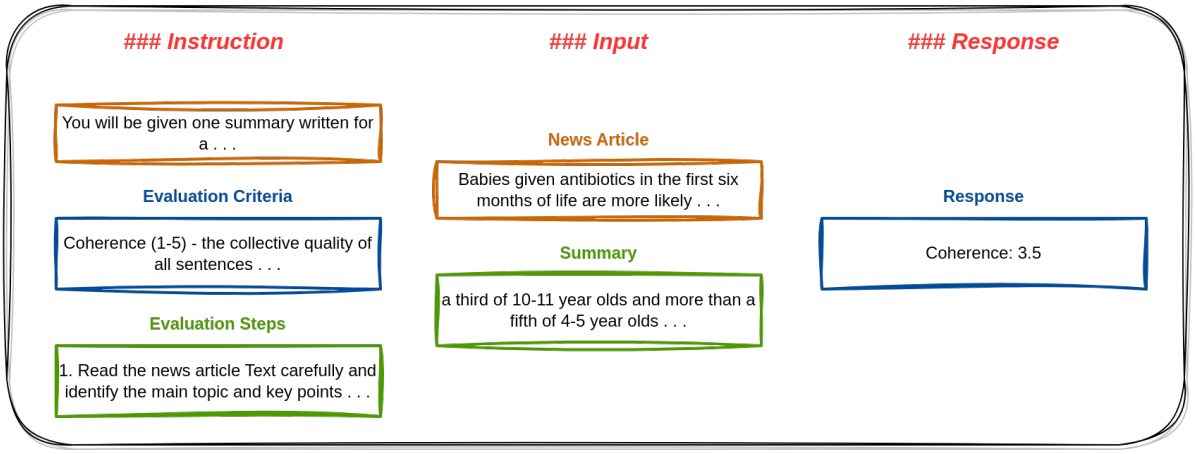

Figure 1: Prompt Design for Prompt Based Evaluation

	#of examples
Train	320
Dev	1280
Test	825

Table 1: Dataset Statistics

proposed approaches needed to determine different prompting strategies to improve model performance.

The dataset statistics are shown in Table 1

3 Related Work

Early work on NLG evaluation includes BLEURT, BERTScore and BARTScore to name a few. BLEURT and BERTScore both rely upon golden reference text to score the model generated text. Both these models propose finetuning the BERT model to predict a similarity score between the reference output and the model generated output. BARTScore leverages the natural language generation capability of BART model and proposes various different approaches of automated scoring some of which can be used even without knowing the reference output.

Similar to BARTScore, GPTScore (Fu et al., 2023) use the log-likelihood of the model generated output given the source text as a way of scoring the quality of the generated text. It carried out extensive experiments using different model sizes and different model types on a variety of different NLG evaluation tasks.

G-Eval (Liu et al., 2023) takes it a step further. It proposes to leverage the language generation capabilities of LLM to directly predict an evaluation score. As part of the prompt G-Eval provides the model with the metric definition and the model defined evaluation steps for each metric.

4 Experiments

The competition allowed only variants of the 13B LLaMA and LLaMA2 models, as well as quantized versions of LLaMA or LLaMA2 models with 60B+ parameters. Our main aim was to compare prompt-based evaluation and log-likelihood-based evaluation techniques across different model sizes. Therefore, we decided to work with the NousHermes-13B (Teknium, 2023) and Platypus-70B (Lee et al., 2023a) models. However, since these two models belong to different LLaMA families, we also included the results obtained using the Ocra-13B model (Lee et al., 2023b), which is based on LLaMA2, for a fair comparison.

We experimented with two different approaches as follows:

4.1 Prompt-based evaluation

Prompt-based evaluation involved providing the model with a prompt that contains an instruction to evaluate the summary and provide a score along with the original text, and the summary of the text (Liu et al., 2023).

Two types of prompt-based evaluation techniques were used to assess the quality of the summary of the provided text: 1) a single prompt for a final score and 2) four different prompts to evaluate coherence, consistency, fluency, and relevance. The scores from the four prompts were averaged to produce the final score for the technique. The intuition behind this approach was to reduce the

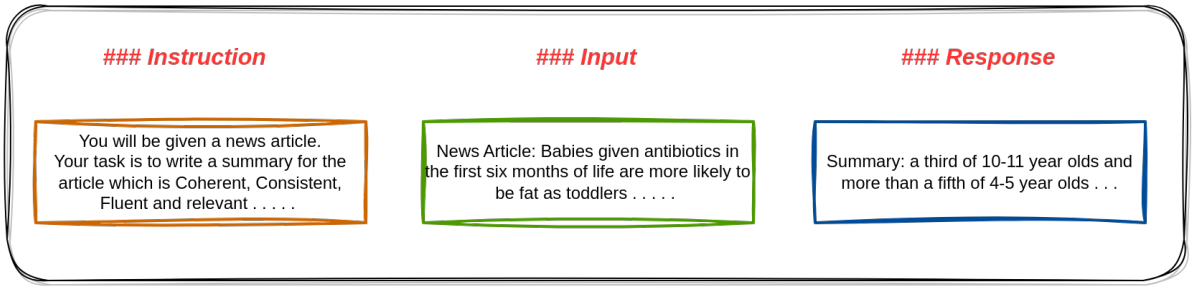

Figure 2: Prompt Design for Log-Likelihood Based Evaluation

complexity of the task and make the model focus on individual aspects, before we average it out.

The prompts used for the two settings are shown in Appendix B and Appendix C respectively. For the second setting of calculating four scores on four different aspects we modified the task description and evaluations steps in the same way as G-Eval (Liu et al., 2023). The prompt design for Prompt Based Evaluation is shown in Figure 1.

For both the prompt settings mentioned in the above paragraph we used sampling to sample 10 output scores for each input example, and then averaged it out to generate a single prediction score.

4.2 Log-Likelihood-based evaluation

Log-Likelihood-based evaluation involved providing the model with a prompt that contains an instruction to generate the summary along with the original text, and the summary of the text. The final score is calculated by multiplying log-likelihood of the tokens of the summary. This method helps to evaluate the likelihood of LLM generating the given summary. If summary is good according to the evaluating LLM, the summary gets a high log-likelihood. This method was used in both GPTScore (Fu et al., 2023) and BARTScore (Yuan et al., 2021).

We adopted a similar strategy as above for likelihood based approaches as well, i.e. a prompt to generate a single likelihood score and four different prompts to obtain four different likelihood values, which are then averaged out. The prompt design for log-likelihood based evaluation is shown in Figure 2. In addition we experimented with two different sets of prompt

- the first set of the prompts is similar to the one we used for prompt-based evaluation. The associated prompt has been shown in Appendix D.
- the second set of the prompts are the ones proposed in GPTScore.

	Nous-Hermes	Ocra	Platypus
Single			
Likelihood	0.314	0.292	0.292
Prompt-based	0.192	0.310	0.398
Average			
Likelihood (Our Prompts)	0.317	0.297	0.298
Likelihood (Original Prompts)	0.320	0.295	0.296
Prompt-based	0.296	0.376	0.463

Table 2: Performance on Dev Set

5 Results

Comparing the likelihood based scores for the Platypus-13B model across the single scoring and the 2 different prompts sets for average scoring from Table 2 we can see that the co-relation values remains the same. Same is the case for the other two models as well. This shows that the prompts are not too relevant for likelihood based approaches.

The likelihood performance of LLaMA2 based models is consistently worse than those of LLaMA based models across all settings. The performance of Ocra-13B model is similar to the NousHermes-13B model in case of likelihood based approach. But considering that prompt based scores are reversed for the two, it seems LLaMA2 based models are generally worse than LLaMA based models in the case of likelihood. We believe that one of the reasons for this could be that LLaMA2 based model's generation distribution might be different. i.e. it might consider most of the summaries to be average in nature resulting in low likelihood. Fur-

ther analysis and experiments with other instruction tuned model might be required to understand if other LLaMA2 based models also have similar results.

For the prompt based evaluations we can see that using a single prompt to get a score led to performance degradation across all the three models. This shows that the use of a complex prompt makes the reasoning process difficult for the model.

The performance of LLaMA2 based Ocra-13B model is much better than the LLaMA based NousHermes model. The performance different between the two models is vastly different. The two reasons for this could be (a) Ocra is a LLaMA2 based model or (b) different instrucion tuning data used for the two models. We believe the first to be true as it is eviden from the huggingface leaderboards, where LLaMA2 based models are consistentl ranked higher than LLaMA based models.

Lastly the quantized Platypus-70B model surpasses the performance of Ocra-13b model in the scoring based approach showing that bigger models tend to improve performance, even if it has been quantized down to 4-bits.

We tested the best models across both the settings i.e. the likelihood and the prompt based approach on test dataset. All the submission were made under the team name of *Beginners*. NousHermes-13b model achieved the best results using the likelihood based approach with a score 0.38 on test data. A single prompt was used as shown in Appendix D with the submission ID *20138*. The Platypus-70B model achieved the best score score in the prompt based approach. It got a score of 0.44 on test data by averaging the scores obtained using four different prompts for four different aspects (consistency, fluency, relevance, coherence) with submission ID *20254*.

6 Conclusion

Prompt-based evaluation technique outperforms log-likelihood-based evaluation technique in text summarization evaluation. However, evaluating single summaries is challenging, as there are many different aspects to consider, and some aspects may be more important than others. Averaging scores from different aspects improves performance, suggesting that there are other evaluation aspects that we did not consider. LLaMA2 based models seem better at reasoning and making decisions, even with low likelihood scores. Therefore, combining Prompt-based evaluation with LLaMA2 based models may further improve text summarization evaluation results.

Limitations

This experiment used smaller open-source models (13B or quantized 70B), but the inference hardware requirements for most of the models used in this paper are still high. For example, both the 13B and quantized 70B models took 24 hours to run on two 48GB A6000 GPU machines for the prompt scoring based approach, making it expensive and time-consuming to iterate through different ideas.

References

2022. Palm: Scaling language modeling with pathways.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2023a. Platypus: Quick, cheap, and powerful refinement of llms.

Ariel N. Lee, Cole J. Hunter, Nataniel Ruiz, Bleys Goodson, Wing Lian, Guan Wang, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023b. Openorcaplatypus: Llama2-13b model instruct-tuned on filtered openorcav1 gpt-4 dataset and merged with divergent stem and logic dataset model. https://huggingface.co/Open-Orca/OpenOrca-Platypus2-13B.

Christoph Leiter, Juri Opitz, Daniel Deutsch, Yang Gao, Rotem Dror, and Steffen Eger. 2023. The eval4nlp 2023 shared task on prompting large language models as explainable metrics. In *Proceedings of the 4th Workshop on Evaluation and Comparison for NLP systems*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation.

Teknium. 2023. Noushermes13b. https://huggingface.co/NousResearch/Nous-Hermes-13b.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

A System Settings

All the experiments were run of A6000 40GB GPUs. We used pytorch-2.0.1 and transformers=4.32.0 and nvidia-cuda-11.7.

B Single Scoring Prompt

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction:

You will be given a news article.
Your task is to rate the generated summary with a score of 1-5.
To rate the summary evaluate it on 4 different aspects Coherent, Consistent, Fluent and relevant.
Please make sure you read and understand the definitions carefully. Please keep this document open while reviewing, and refer to it as needed.

Coherence - the collective quality of all sentences. The summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to a coherent body of information about a topic.
Consistency - the factual alignment between the summary and the news article. A factually consistent summary contains only statements that are entailed by the news article. Annotators were also asked to penalize summaries that contained hallucinated facts.
Fluency - the quality of the summary in terms of grammar, spelling, punctuation, word choice, and sentence structure.
Relevance - selection of important content from the news article. The summary should include only important information from the news article. Annotators were instructed to penalize summaries which contained redundancies and excess information.

Input:

News Article: source_text

Summary: summary

Evaluation Form (scores ONLY):

Response: Score

C Scoring Prompt

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction:

You will be given one summary written for a news article.
Your task is to rate the summary on one metric.
Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:
Coherence (1-5) - the collective quality of all sentences. We align this dimension with the DUC quality question of structure and coherence whereby "the summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to a coherent body of information about a topic."

Evaluation Steps:
1. Read the news article Text carefully and identify the main topic and key points.
2. Read the Summary and compare it to the news article Text. Check if the Summary covers the main topic and key points of the news article Text, and if it presents them in a clear and logical order.
3. Assign a score for coherence on a scale of 1 to 5 (score can be decimal or integer), where 1 is the lowest and 5 is the highest based on the Evaluation Criteria.

Input:

news article Text: source_text

Summary: summary

Evaluation Form (scores ONLY):

Response: *Coherence:*

D Likelihood Prompt

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction:
You will be given a news article.
Your task is to write a summary for the article which is Coherent, Consistent, Fluent and relevant.
Please make sure you read and understand the definitions carefully. Please keep this document open while reviewing, and refer to it as needed.

Coherence - the collective quality of all sentences. The summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to a coherent body of information about a topic.
Consistency - the factual alignment between the summary and the news article. A factually consistent summary contains only statements that are entailed by the news article. Annotators were also asked to penalize summaries that contained hallucinated facts.
Fluency - the quality of the summary in terms of grammar, spelling, punctuation, word choice, and sentence structure.
Relevance - selection of important content from the news article. The summary should include only important information from the news article. Annotators were instructed to penalize summaries which contained redundancies and excess information.

Input:

News Article: source_text

Response: summary