CERET: Cost-Effective Extrinsic Refinement for Text Generation

Jason Cai, Hang Su, Monica Sunkara, Igor Shalyminov, Saab Mansour
AWS AI Labs
{cjinglun, shawnsu, sunkaral, shalymin, saabm}@amazon.com

Abstract

Large Language Models (LLMs) are powerful models for generation tasks, but they may not generate good quality outputs in their first attempt. Apart from model fine-tuning, existing approaches to improve prediction accuracy and quality typically involve LLM self-improvement / self-reflection that incorporate feedback from models themselves. Despite their effectiveness, these methods are hindered by their high computational cost and lack of scalability. In this work, we propose CERET, a method for refining text generations by considering semantic stability, entailment and inter-sample uncertainty measures. Experimental results show that CERET outperforms Self-consistency and Self-rerank baselines consistently under various task setups, by 1.6% in Rouge-1 for abstractive summarization and 3.5% in hit rate for question answering. Compared to LLM Self-rerank method, our approach only requires 9.4% of its latency and is more cost-effective. [1]

1 Introduction

Large Language Models (LLMs) like GPT (Brown et al., 2020), Claude, PaLM (Chowdhery et al., 2022; Anil et al., 2023), and Llama (Touvron et al., 2023) have showcased unprecedented capabilities in natural language understanding and generation. These models, with parameter counts reaching into the hundreds of billions, have become pivotal in advancing the frontier of natural language processing (NLP). Despite their impressive fluency and coherence, language models frequently generate content that is incomplete, biased, or misleading in their initial attempts across a variety of language generation tasks.

The key challenge is that while pre-training equips base models with broad linguistic knowledge, it does not necessarily impart the specialized skills needed for particular downstream tasks. Current methodologies for enhancing LLM generation largely involve resource-intensive approaches such as supervised fine-tuning (SFT), which relies heavily on domain-specific training data, or reinforcement learning from human feedback (RLHF), which necessitates extensive human annotations. However, curating large volumes of high-quality domain-specific data and human feedback often proves prohibitively expensive and time-consuming in practice, severely limiting the applicability of SFT and RLHF. By integrating feedback derived from the generated outputs, self-improvement / self-reflection approaches enhance generations in an iterative manner (Madaan et al., 2023; Yao et al., 2023a). These approaches empower the LLM to adapt to specific tasks and domains by learning from its own mistakes and successes. Nevertheless, the substantial cost linked to iterative inference poses challenges for scalability and applicability real-time systems.

This paper introduces CERET, a novel method designed to refine text generation in a rapid, low-resource manner to reduce the need for domain-specific training data or expensive human annotations. The cornerstone of CERET lies in its ability to enhance generated content by holistically considering three key scoring dimensions - semantic stability, entailment, and inter-sample uncertainty measures.

Semantic stability scoring quantifies the linguistic invariance among multiple candidate outputs generated by the base model for the same input, indicating higher confidence for more stable candidates. Entailment scoring leverages natural language inference (NLI) models to quantify the logical entailment relations between candidate outputs, preferring candidates that maximally entail others. Inter-sample uncertainty scoring penalizes candidates that are semantically similar to outputs for

[1] The source code and data samples are released at https://github.com/amazon-science/CERET-LLM-refine.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7377–7390

different inputs, a signal of greater uncertainty.

Our approach operates in a rapid, zero-shot manner without any domain-specific training data, reward modeling, or human feedback. The proposed scoring and refinement process encapsulates an efficient way to improve text generation across a diverse spectrum of NLP tasks, including abstractive summarization, dialogue response generation, and open-domain question answering. Through a rigorous series of experiments on standard datasets, CERET is empirically validated to significantly outperform baseline methods such as Self-consistency and Self-reranking across both summarization and QA tasks. Beyond its superior performance, CERET stands out for its practicality and cost-effectiveness, making it a promising solution for real-world applications where domain-specific resources and annotations are limited or unavailable. This paper not only presents CERET as a valuable novel contribution to the growing field of NLP but also underscores its potential impact on advancing the practical deployment of text generation across a myriad of domains. The main contributions are summarized as follows:

- CERET is proposed as a holistic framework for enhancing generation quality, encompassing semantic stability, entailment, and inter-sample uncertainty measures.
- The refinement process is data efficient and cost-effective, without the requirement for domain-specific training data or expensive annotations.
- The proposed approach can be applied across various natural language processing tasks, such as text summarization, dialogue response generation and question-answering systems.
- CERET is highlighted for its practicality and efficiency, presenting only a minor fraction of the usual latency associated with a single generation call, which positions it as a feasible solution for real-world applications.

2 Approach

2.1 System Architecture

CERET consists of three scoring methods, namely Semantic Stability Scoring, Entailment Scoring and Inter-sample Uncertainty Scoring, for calibrating the quality of LLM predictions. The overview of the proposed system is illustrated in Figure 1. Firstly, a diverse set of candidates are sampled from LLMs. Then each individual scoring method will produce a separate score from a certain perspective. Based on the scores in three dimensions, a linear weighted final confidence score is computed to measure the quality of each prediction. The prediction with the highest confidence score is selected as the final model prediction.

2.2 Semantic Stability Scoring

We first introduce an intra-sample scoring method, Semantic Stability Scoring, which is motivated by the need to enhance the confidence and reliability of sample generations produced by LLMs. The scientific rationale is inspired by Kuhn et al. (2023) and Yin et al. (2022), where it was shown that a sample generation exhibits higher confidence when it demonstrates considerable semantic stability or linguistic invariance among other generations. However, the semantic stability measured in Kuhn et al. (2023) involves clustering sampled generations for each sample, which is computationally expensive for real world applications at large scale.

In contrast, we propose a cluster-free method for semantic stability modeling. Specifically, Semantic Stability Scoring is formulated as the following: Given input data sample x, the model generates k predictions $(y_1, ..., y_k)$. For each y_i, a fixed pre-trained language model produces its corresponding embedding $e(y_i)$. In practice, we leverage RoBERTa (A Robustly Optimized BERT Pretraining Approach) (Liu et al., 2019) as the pre-trained language model, and the final hidden representation of "<s>" token from RoBERTa, is regarded as $e(y_i)$. To aggregate all intra-sample representations, we treat the average-pooled embedding $\bar{e}$ as a stability reference point:

$$\bar{e} = mean(e(y_1), ..., e(y_k)) \quad (1)$$

A lower distance between an embedding and the reference point implies a higher stability. We can employ Euclidean distance or cosine distance as the distance metric $||\cdot||$. The stability score s^i_{sta} is defined as the negative distance between $e(y_i)$ and the stability reference point $\bar{e}$:

$$s^i_{sta} = -||e(y_i) - \bar{e}|| \quad (2)$$

2.3 Entailment Scoring

Entailment scoring is another intra-sample scoring method, fully powered by entailment relation: "p

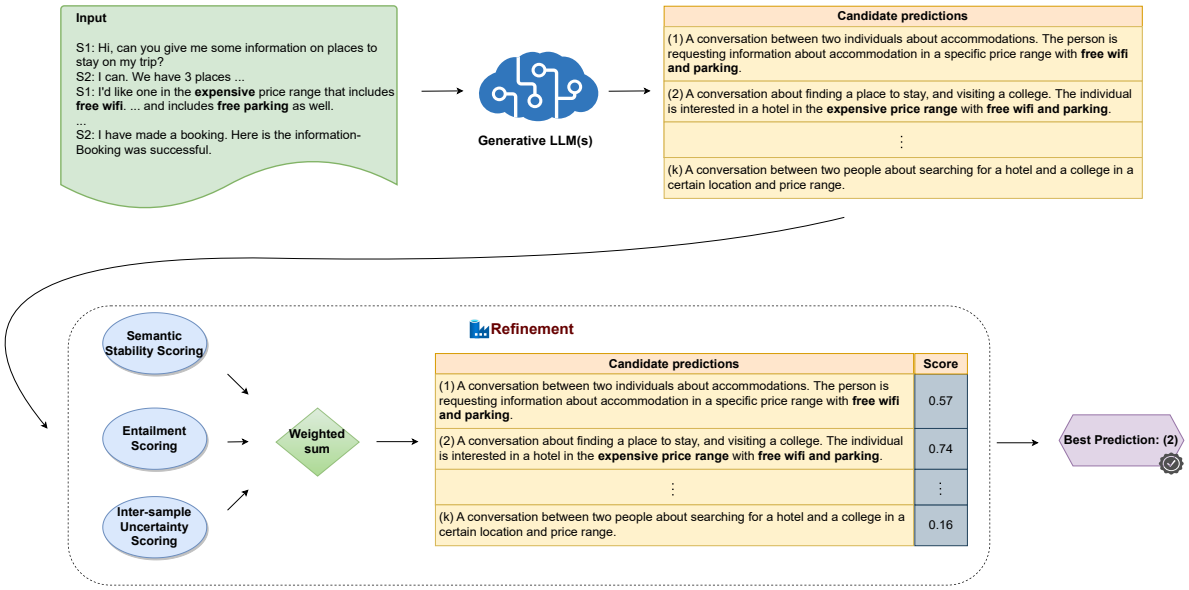

Figure 1: CERET overview

entails h" ($p \Rightarrow h$), if a human reading premise p would infer that hypothesis h is most likely true. This intrinsic connection to human inference aligns closely with the objective that language models should have the capacity to generate content that is not only syntactically accurate but also semantically meaningful. In the entailment scoring process, when an LLM generates k predictions $(y_1, ..., y_k)$, each prediction's entailment relation to others is quantified by a Natural Language Inference (NLI) model. The scalar value s_j^i reflects the degree to which the content of y_i logically entails y_j.

$$s_j^i = \text{ENT}(y_i, y_j) \tag{3}$$

Although the scalar function for entailment can be evaluated by the base LLM itself, such an approach leads to a higher computational cost. Hence, we resort to a more efficient and lightweight NLI model. Specifically, we adopt DeBERTa (Decoding-enhanced BERT with disentangled attention) (He et al., 2021) for this work. The NLI task is treated as a sequence classification problem: The texts y_i, y_j are concatenated, with special tokens as separators, to form the input to DeBERTa. The final hidden representations of pretrained DeBERTa are passed to a pooling layer and a classifier, to obtain softmax probability for three categories, namely Neutral, Entailment and Contradiction. The softmax probability for Entailment is used as $\text{ENT}(y_i, y_j)$.

A generation is plausible if it entails as many other sampled generations as possible. With the top k sampled model predictions, the entailment score for sample y_i is computed as follows:

$$s_{ent}^i = \frac{1}{k} \sum_{1 \le j \ne i \le k} s_j^i + LP(y_i) \tag{4}$$

Note that the preferred prediction will likely have rich information, and may be lengthy in certain situations. A length penalty $LP(y_i)$ is applied to this entailment score, in case lengthy outputs harm the expected conciseness.

$$LP(y_i) = 1 - (1 + q \cdot \text{len}(y_i))^p \tag{5}$$

where $0 \le q < 1$ and $p > 1$ are hyperparameters[2].

2.4 Inter-sample Uncertainty Scoring

In contrast to the methods above, the following is an inter-sample scoring method, which is inspired by uncertain region analysis. We first build an embedding space for all sampled predictions with a standalone model, e.g., RoBERTa. The rationale behind this inter-sample scoring method is that when a sampled prediction is located near predictions from different input data samples in

[2]In our practice, we chose q = 0 (i.e. no penalty). We found that our beam search sampled predictions generally have very comparable lengths. Nevertheless, the length penalty may benefit other datasets or decoding settings.

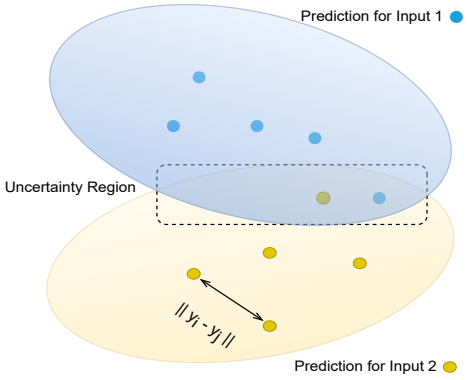

Figure 2: Inter-sample uncertainty region

the embedding space, this prediction is likely to be uncertain, illustrated in Figure 2. Uncertain predictions are down-weighted by a lower uncertainty score s_{unc}.

Suppose dataset D has size N. For each input x, top k predictions are generated by LLM(s), resulting in $k \cdot N$ predictions in total: $\{y_i\}_{1 \leq i \leq kN}$. The Euclidean distance of all possible prediction pairs $\|y_i - y_j\|, i \neq j$ are computed and cached. According to Euclidean distance, the nearest neighbor set $\mathcal{N}(i)$ is constructed for each prediction y_i. The inter-sample uncertainty score s^i_{unc} is computed as follows:

$$s^i_{unc} = -\sum_{j \in \mathcal{N}(i)} \frac{\mathbb{I}(\hat{x}_i \neq \hat{x}_j)}{(1 + \|y_i - y_j\|)} \quad (6)$$

where $\mathbb{I}(\cdot)$ is the indicator function. $\hat{x}_i$ denotes the input sample x for prediction y_i. Note that possibly $\hat{x}_i = \hat{x}_j$ when $i \neq j$. $||y_i - y_j||$ in denominator of Equation 6 is a regularization term, ensuring a further y_j is assigned with a lower weight for uncertainty. A negative sign is added to ensure that a higher score is better. In case when the dataset is large, the computation cost for obtaining pairwise Euclidean distances and nearest neighbors can be mitigated by limiting data size N to a certain batch (e.g., 1000). Additionally, the LLM generations in practice mostly have the number of sampled predictions $k \leq 20$. Thus the efficiency of this method can be maintained.

2.5 Computation of Final Score

All separate scores s_{sta}, s_{ent}, s_{unc} are transformed to the interval $(0, 1)$ by applying sigmoid function

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-ux}} \quad (7)$$

where $u > 0$ is an additional scaling factor [3]. Since three scores have distinct ranges, u is applied to ensure their scaled ranges are comparable. The final confidence score is a linear weighted score based on three dimensions.

$$s = \alpha \cdot s_{sta} + \beta \cdot s_{ent} + \gamma \cdot s_{unc} \quad (8)$$

The coefficients α, β, γ are tuned on validation datasets. To mimic the properties of probability for intuitive interpretation, the following constraints are imposed:

$$\begin{cases} \alpha + \beta + \gamma = 1 \\ \alpha, \beta, \gamma \geq 0 \end{cases} \quad (9)$$

3 Experimental Setup

3.1 Datasets

We evaluate the proposed approach CERET on Abstractive Summarization and Question Answering (QA) tasks. For summarization, we consider two dialogue summarization datasets: TodSum (Zhao et al., 2021) and DialogSum (Chen et al., 2021), as dialogue summarization has been a challenging summaization use case due to its multi-speaker nature and varying structures. TodSum is a dialogue summarization dataset based on MultiWoz (Budzianowski et al., 2018). Out of 7 MultiWoz domains, it contains 5 and totals 9,906 dialogues. DialogSum is a multi-domain dataset, mostly consisting of casual/spoken style daily conversations. It is based on top of the existing datasets around English practice conversations and English listening comprehension exams. For TodSum and DialogSum, official validation/test sets are used throughout this work.

For QA, we use TriviaQA (Joshi et al., 2017) and Natural Questions (Lee et al., 2019, the NQ-Open version) datasets. TriviaQA contains 95,956 QA pairs with 40,478 unique answers and 662,659 evidence documents. It contains question-answer pairs from 14 trivia and quiz-league websites, with the associated Wikipedia pages as evidence sources. NQ-Open is an open-domain question answering benchmark, a subset of Natural Questions (Kwiatkowski et al., 2019) with short answers and with evidence documents discarded. It contains 91,535 QA pairs. For TriviaQA and Natural Questions, the official test set is only available for online

[3] For each scoring dimension, there is a dedicated value of u.

benchmarking. We split the official validation sets into validation/test sets with an 1:1 ratio for our experiments.

3.2 Baselines and Evaluation Metrics

We choose Vicuna v1.3 (Chiang et al., 2023) and Llama 2 chat (Touvron et al., 2023) as our base LLMs. Vicuna is an open-source chatbot, fine-tuned from Llama with supervised instruction fine-tuning using around 125K conversations collected from ShareGPT. Llama 2 was pretrained on publicly available online data sources and trained on 2 trillion tokens, and was initially created through supervised fine-tuning and then iteratively refined using Reinforcement Learning from Human Feedback (RLHF). Both Vicuna v1.3 and Llama 2 were released in mid 2023.

Given each input prompt, we generate k LLM predictions by beam search sampling (Vijayakumar et al., 2016), while setting a high temperature to encourage diversity and increase the scope for improvement. The beam search sampled predictions are considered as **No-refinement** baseline. We further considered two baselines. (1) **Self-rerank:** In the Self-rerank approach, all predictions generated by the base LLM and the task context are fed back into the base LLM itself. The model is then instructed to select the single best prediction from the candidate set. The Self-rerank baseline provides insight into the capabilities of the base LLM to refine its own output as a straightforward reranking task. (Prompt templates in Appendix D) (2) **Self-consistency:** The Self-consistency (Wang et al., 2023) approach determines the best prediction through a majority vote among all generated predictions, after marginalizing out reasoning paths. This is a cost-effective approach for refinement, but it can only be applied to tasks with fixed answers. Hence, it is included as a baseline for open domain QA tasks. Furthermore, we also report **Oracle** scores, which represent the *upper bound* of refinement/re-ranking performance: Given an input x, we obtain a candidate prediction set $\{y_i\}_i$, out of these y_i's, we choose the best one according to certain evaluation metric (E.g., Rouge, Exact Match, etc) to compute oracle performance .

On the QA tasks with a closed set of answers, we evaluate the models against **Hit Rate**: A prediction receives score 1 if it exactly matches one of multiple target answers, otherwise score 0 is assigned. On the summarization tasks assuming more open-ended model outputs, we evaluate the models against **Rouge-1/2/L** (Lin, 2004) and **BERTScore** (Zhang et al., 2020). Rouge[4] is a series of metrics counting the number of overlapping word n-grams in the reference and the generated summary, working on top of 1-/2-grams (as the index in the metric name denotes). Rouge-L is a variant of the metric based on the Longest Common Subsequence between the reference and the generated summary. BERTScore[5] is a semantic similarity metric working in the BERT (Devlin et al., 2019) embedding space by computing pairwise cosine similarities between each predicted summary's token and each reference summary's token.

3.3 Implementation Details

For the purpose of experimentation, we opt for 13B models for both Vicuna v1.3 and Llama 2. In the LLM beam search sampling phase, we set the temperature parameter t to 0.7, and for each input sample, we accumulate the top $k = 5$ LLM predictions for subsequent refinement. We activate the half-precision mode to enhance the efficiency of LLM generation. In order to preserve generation quality, quantization is not applied to the LLMs. The entirety of our experiments is performed with NVIDIA A100 GPUs, conducted in a single run. For TodSum dataset, the LLM generation time (from input to the end of beam search sampling) is 2.38/2.85 sec for Vicuna 1.3 and Llama 2 respectively.

Regarding the BERT models integrated into the CERET pipeline, we select base-sized models for efficiency, namely RoBERTa-base (125M) and DeBERTa-v3-base-mnli (184M). The coefficients α, β, γ for final weighted scoring are tuned on separate validation sets, where a grid search is conducted with step size of 0.1. In uncertainty scoring, we found the size of nearest neighborhood $s = 3, 5$ generally lead to satisfactory performance in validation sets, and it is finally set to 5 in all test settings. Note that after post-processing, the duplicate predictions are merged. A neighborhood of size 5 may represent more than 5 raw predictions.

4 Results and Analysis

4.1 Effectiveness and Efficiency

Abstractive Summarization. The experimental results for dialogue summarization are presented in

[4] https://github.com/pltrdy/rouge
[5] https://github.com/Tiiiger/bert_score

Base LLM	Refinement	TodSum				DialogSum			
		Rouge-1	Rouge-2	Rouge-L	BERTScore F1	Rouge-1	Rouge-2	Rouge-L	BERTScore F1
Vicuna v1.3	No	38.8	9.9	25.8	22.9	32.7	10.3	26.3	27.6
	Self-rerank	39.1	10.2	25.6	22.9	32.7	10.4	26.1	27.3
	CERET	**40.7**	**11.1**	**25.9**	**24.5**	**34.0**	**10.9**	**27.2**	**28.5**
	Oracle	45.7	13.0	29.6	28.6	41.7	15.4	33.3	33.8
Llama 2 chat	No	40.1	10.3	26.4	23.2	30.3	9.4	24.5	26.5
	Self-rerank	40.5	10.5	26.5	23.2	30.4	9.7	24.6	26.3
	CERET	**41.4**	**11.2**	**26.7**	**23.8**	**30.8**	**9.8**	**25.0**	**27.2**
	Oracle	46.0	13.1	29.6	27.3	38.0	13.2	30.6	31.2

Table 1: Comparison of refinement methods for Abstractive Summarization tasks

Base LLM	Refinement	TriviaQA	Natual Questions
Vicuna v1.3	No	57.8	18.6
	Self-rerank	60.0	19.4
	Self-consistency	59.7	20.0
	CERET	**62.0**	**21.2**
	Oracle	70.0	27.7
Llama 2 chat	No	51.0	15.3
	Self-rerank	51.4	15.4
	Self-consistency	51.6	15.7
	CERET	**55.1**	**17.2**
	Oracle	66.7	24.5

Table 2: Comparison of refinement methods for Question Answering tasks

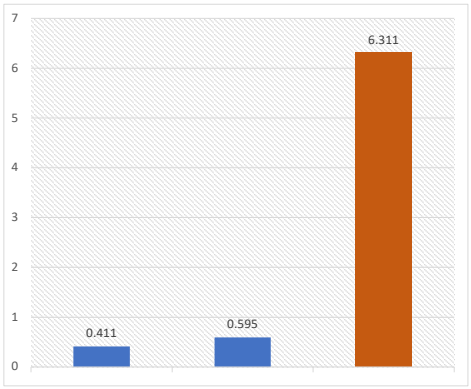

Figure 3: Latency (sec) per input sample. From left to right: *BERT inference, CERET, and LLM self-rerank.

Table 1. The initial performance of Vicuna v1.3 and Llama 2 chat in TodSum and DialogSum only result in moderate quality in generated summaries. However, the introduction of CERET brings obvious benefits into the enhancement of summarization outputs. Specifically, CERET achieves a decent improvement of 0.5-1.9 in Rouge-1 scores and 0.6-1.6 in BERTScore F1.

The method consistently outperforms Self-rerank, emphasizing the significance of leveraging semantic stability, entailment, and inter-sample uncertainty measures in refining large language model generations.

Question Answering. As shown in Table 2, the baseline performance of the LLM on TriviaQA and Natural Questions reflects a gap between the difficulty of these two tasks. Despite the fact that they are both evaluated in closed-book setting, Natural Questions dataset has lower hit rate as it contains various challenging open-ended questions (e.g., Q: "Philadelphia is known as the city of what?". A: "City of Brotherly Love") Regardless of the challenges, CERET is able to improve upon no-refinement baseline for 1.6-4.2 points in hit rate, which consistently surpasses the both Self-rerank and Self-consistency approaches, indicating its effectiveness across diverse knowledge domains.

Inference Efficiency. Efficient inference is a crucial aspect of deploying language models in real-world applications. We analyze and compare the inference efficiency of the proposed CERET method against LLM Self-rerank. We use latency (in seconds) per input sample as a metric for assessing the efficiency of different inference pipelines. We report results on the TodSum dataset. Since the validation/test sets have size ≤ 1000, we use the entire sets instead of small batches for Inter-Sample Uncertainty scoring.

As shown in Figure 3, CERET exhibits remarkable efficiency advantages compared to LLM Self-rerank. The latency required by CERET is only 9.4% of the latency observed in LLM Self-rerank[6]. The majority of the latency in the CERET pipeline is attributed to *BERT inference, where *BERT refers to RoBERTa and DeBERTa models. The efficient integration of these models within the CERET framework contributes to its overall effectiveness while maintaining a significantly reduced latency compared to Self-rerank approaches. The efficiency improvement is particularly noteworthy, especially considering the demands of real-time applications where low latency is imperative.

Overall Observations. Figure 4 provides a com-

[6]Both CERET and Self-rerank deal with predictions after LLM generation, and hence they don't include the beam search sampling time.

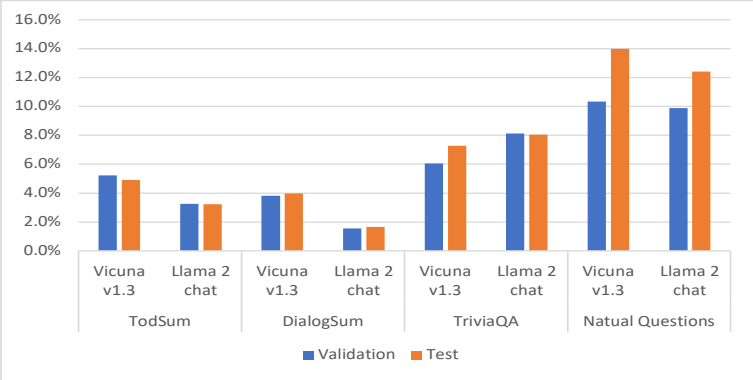

Figure 4: Relative performance gains on validation and test sets. The best coefficient combination is tuned on validation sets. Evaluation metrics: Rouge-1 for TodSum and DialogSum, and hit rate for Trivia QA and Natural Questions.

		Summarization - Rouge-1		QA - Hit rate	
Base LLM	**Refinement**	**TodSum**	**DialogSum**	**TriviaQA**	**Natual Questions**
Vicuna v1.3	No	38.78	32.67	57.82	18.61
	Semantic stability only	40.27	34.17	61.96	21.19
	Entailment only	40.67	32.30	57.66	18.51
	Uncertainty only	39.27	32.60	59.90	19.46
	CERET	40.69	34.27	61.96	21.19

Table 3: Ablation study of individual scoring dimensions

prehensive overview of the relative performance improvement achieved on both validation and test sets. The test performance gains observed are generally on par with the validation settings and, in certain instances, even surpass them, as in the case of Natural Questions. This suggests that the weight tuning strategy employed during validation exhibits robustness and generalizability when applied to test sets. The potential explanation for larger gains in certain test cases could be attributed to the random split of test sets, providing certain sets with more room for improvement.

The overall theme in the observed results is the consistently superior performance of CERET across all evaluated tasks. Furthermore, the consistency in performance gains between validation and test settings showcases the reliability and adaptability of the proposed CERET method across various settings in natural language processing tasks.

4.2 Ablations and Hyperparameter Analysis

Ablations. We systematically evaluate the impact of individual components within the CERET on both summarization and QA tasks, and present the findings in Table 3. The results indicate that all three scoring dimensions have positive contributions in certain task scenario, compared to no-refinement baseline. Notably, semantic stability alone improves summarization Rouge-1 scores from 38.78 to 40.27 and 32.67 to 34.17 for TodSum and DialogSum respectively. Similarly, for question-answering, semantic stability increases the hit rate from 57.82 to 61.96, and 18.61.67 to 21.19, which are promising improvements.

Various NLP tasks have their own unique characteristics, suggesting that effectiveness of specific refinement dimensions might vary. For example, when considering the TodSum task, the nuances of entailment play a pivotal role in summarization quality for task oriented dialogues, where entailment scoring leads to the most significant gains. We further observed uncertainty scoring exhibits the best improvement in Appendix A.

These insights underscore the synergies between semantic stability, entailment and uncertainty measures, highlighting their complementary roles in refining language model outputs. The comprehensive integration of these aspects in the CERET method showcases their collective impact, providing a flexible and contextually relevant refinement framework for various base LLMs and natural language processing tasks.

Hyperparameter Analysis. In Equation 8, coefficients α, β, γ are weights for three scoring dimensions respectively. To further investigate the sensitivity of non-trivial coefficients (when all co-

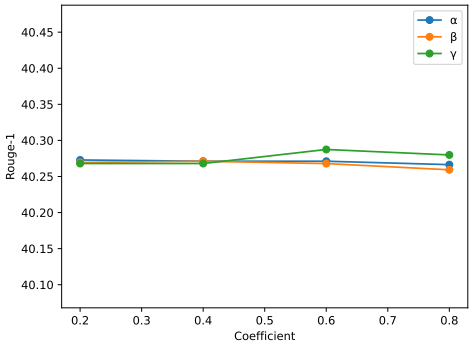

Figure 5: Sensitivity analysis of coefficients for TodSum

efficients are non-zero), a systematic approach was employed to assess the impact of individual coefficients on the overall model performance (Figure 5). A sensitivity analysis was conducted by keeping one coefficient, denoted as x, in a state of flux, while concurrently setting the other two coefficients, y and z, to be

$$y = z = \frac{1-x}{2}$$

Consider the green line in Figure 5: γ is set as the variable, the relationship $\alpha = \beta = \frac{1-\gamma}{2}$ is maintained. This variation allows for an in-depth exploration of the model's sensitivity to changes in each specific coefficient. illustrates that when all coefficients are non-zero, the model's performance remains relatively stable, with a fluctuation of Rouge-1 within 0.1, indicating the robustness to variations in individual coefficient values.

5 Related Work

Prompting strategy for LLM improvement/refinement. Improving LLM outputs by achieving behaviors close to reasoning has been explored before (Wei et al., 2022; Wang et al., 2023; Yao et al., 2023a,c; Madaan et al., 2023; Yao et al., 2023b; Gou et al., 2023; Akyurek et al., 2023). Wei et al. (2022) introduce a specific technique of formulating prompts for the model dubbed *Chain-of-Thought*. Essentially a series of intermediate reasoning steps that the model is asked to explicitly output, Chain of Thought significantly improves the ability of LLMs to perform complex reasoning.

Wang et al. (2023) propose a decoding strategy dubbed *Self-consistency* — under which the model, prompted in a chain-of-thought way, generates a set of sample predictions, or reasoning paths. The paths are then marginalized out, and the most consistent answer (the one which the most reasoning paths lead to) is selected as the final one. Huang et al. (2022) use this approach to improve LLMs without annotated data — they select the most consistent answer from the candidates pool, collect all the reasoning paths leading to that answer, and augment the trainset of the target model with the resulting data points. In contrast to Self-consistency, the *Self-Refine* approach of Madaan et al. (2023) assumes that the model iteratively provides verbal feedback on its own outputs, and incorporates it in the next generation round. Moreover, CRITIC (Gou et al., 2023) empowers Language Models (LLMs) to independently verify and improve their own outputs with external toolkits, similar to the way humans make use of tools. All the approaches above require prompt engineering, while we tackle the problem from another perspective.

Model confidence/uncertainty without self-feedback. A parallel line of work in improving LLM generation outputs is related to assessing the model confidence and the uncertainty of its predictions without iterative language model calls (Jiang et al., 2021; Lang et al., 2022; Wang et al., 2022; Kuhn et al., 2023; Ge et al., 2023; Jiang et al., 2023; Vernikos et al., 2023). Kuhn et al. (2023) define *semantic entropy*, a metric that incorporates linguistic invariance of the individual output candidates sharing identical meanings. This metric helps identify the correct model's predictions as evaluated on question answering task. LLM-Blender Jiang et al. (2023) adopts a two-stage design, rank-and-fuse, to generate highly confident and superior candidate outputs. Ge et al. (2023) use uncertainty estimation in order to create modified pseudolabels, and define uncertainty of a pseudolabel (obtained using stochastic dropout-based model inference) as its proximity to other different pseudolabels for the same data point. Training on the selected pseudolabels increases performance in binary and multiclass classification, as well as Natural Language Understanding tasks. Selecting high-confidence pseudolabels is also a key aspect of the co-training technique proposed by Lang et al. (2022), where both partial access and full access settings are studied. All these methods explore uncertainty/confidence from a certain perspective, while our approach combines the uncertainty/confidence with semantic stability and entailment, and we further proposed a framework

for these three dimensions and investigated their synergies.

6 Conclusions

Our proposed CERET is an efficient framework to enhance text generation without the need for domain-specific training data or expensive annotations. By considering semantic stability, entailment, and inter-sample uncertainty measures, our approach significantly improves the quality of text generation across multiple natural language processing tasks. The efficiency and cost-effectiveness of our approach suggest its potential for wide adoption in real-world applications.

7 Limitations

CERET can be potentially applied to a wide range of NLP problems, including dialogue response generation, open-ended common sense reasoning, and Natural Language Understanding (NLU) by text filling for text continuation. These topics require dedicated investigation and are not yet covered by this paper.

Our experiments show that beam search sampling almost always provides sufficient room for refinement, according to the oracle performance in Table 1 and Table 2. Nevertheless, in certain task or data scenarios, performances of no-refinement baseline and oracle prediction may be close to each other. In that case, the performance of CERET will be limited by oracle results.

References

Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, Toronto, Canada. Association for Computational Linguistics.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

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

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Jiaxin Ge, Hongyin Luo, Yoon Kim, and James R. Glass. 2023. Entailment as robust self-learner. 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 13803–13817. Association for Computational Linguistics.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *CoRR*, abs/2210.11610.

Siddhartha Jain, Xiaofei Ma, Anoop Deoras, and Bing Xiang. 2023. Self-consistency for open-ended generations. *CoRR*, abs/2307.06857.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Hunter Lang, Monica N. Agrawal, Yoon Kim, and David A. Sontag. 2022. Co-training improves prompt-based learning for large language models. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 11985–12003. PMLR.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

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.

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*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *CoRR*, abs/2303.17651.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2023. Follow the wisdom of the crowd: Effective text generation via minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4265–4293, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Giorgos Vernikos, Arthur Bražinskas, Jakub Adamek, Jonathan Mallinson, Aliaksei Severyn, and Eric Malmi. 2023. Small language models improve giants by rewriting their outputs.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yuxia Wang, Daniel Beck, Timothy Baldwin, and Karin Verspoor. 2022. Uncertainty estimation and reduction of pre-trained models for text regression. *Transactions of the Association for Computational Linguistics*, 10:680–696.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Liyan Xu, Yile Gu, Jari Kolehmainen, Haidar Khan, Ankur Gandhe, Ariya Rastrow, Andreas Stolcke, and Ivan Bulyko. 2022. Rescorebert: Discriminative speech recognition rescoring with bert. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6117–6121.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Yao Yao, Zuchao Li, and Hai Zhao. 2023c. Beyond chain-of-thought, effective graph-of-thought reasoning in large language models. *arXiv preprint arXiv:2305.16582*.

Fan Yin, Zhouxing Shi, Cho-Jui Hsieh, and Kai-Wei Chang. 2022. On the sensitivity and stability of model interpretations in NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2631–2647, Dublin, Ireland. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lulu Zhao, Fujia Zheng, Keqing He, Weihao Zeng, Yuejie Lei, Huixing Jiang, Wei Wu, Weiran Xu, Jun Guo, and Fanyu Meng. 2021. Todsum: Task-oriented dialogue summarization with state tracking. *CoRR*, abs/2110.12680.

Appendix

A Ethical Considerations

We have reviewed all licenses of public datasets, which allow the usage for research and paper publication. All datasets are sets are de-identified to ensure anonymity.

Our proposed method has a potential for substantial reductions in both the financial and environmental burdens associated with large language model improvement/refinement. Through minimizing the reliance on extensive data collection and human labeling, our approach serves as an effective safeguard for user and data privacy, mitigating the risk of information leakage during the construction of training corpora.

During the paper writing process, Generative AI was only used for language checking, paraphrasing and polishing.

B Additional Sensitivity Analysis

As described in 4.2 Hyperparameter Analysis, the sensitivity analysis is conducted by keeping one coefficient, x in a state of flux, while concurrently setting the other two coefficients, y and z, to be $y = z = \frac{1-x}{2}$. Empirical results on four datasets show that the performance variations are very limited, with fluctuations of Rouge-1 within 0.2, and hit rate within 0.4. This indicates the stability of the method to variations in individual coefficient values.

C Qualitative Analysis

Selected qualitative examples from both TodSum and TriviaQA datasets are presented in Table 4 and Table 5 respectively. Although the top 5 beam search sampled candidates are considered in experiments, in Table 4 and Table 5 we only present 3 most representative predictions.

In the example from TodSum, CERET is able to select the summary that contains key information "All Saints Church" and also mentions the phone number was provided. In fact, the prediction has the highest entailment score, which means it mostly implies other summaries. Regarding the examples from TriviaQA, the final score is largely determined by semantic stability: "Dotheboys Hall" and "Dotheboys" are close in semantic representation, and "Sir Cloudesley Shovell" (after parsing) actually appears 3 times in top 5 predictions.

D Prompt Templates

The relevant prompt templates for Abstractive Summarization, QA and Self-rerank are presented in Table 6, Table 7 and Table 8 respectively. We adopt Chain-of-Thought (CoT) prompting for open domain QA; while for Abstractive Summarization datasets, the key information is usually straightforward, hence we only include a length specification. Regarding LLM Self-rerank, we tested multiple additional instructions related to semantic stability, entailment/implication and uncertainty, and finally chose to include semantic stability only, as it produces most robust outcomes.

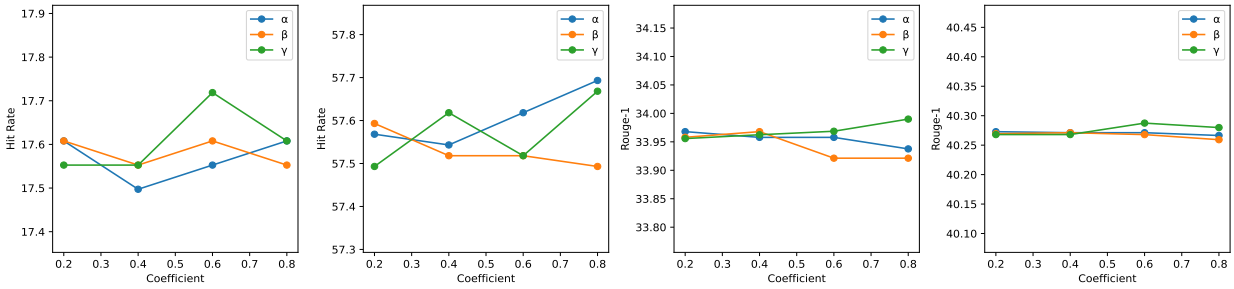

Figure 6: Sensitivity analysis of coefficients for Natural Questions, TriviaQA, DialogSum and TodSum (from left to right in order)

Input dialogue	
Hi there. Can you help me find a train from Norwich to Cambridge? There are several trains on that route. What time would you like to leave/depart? I would like to leave after 12:15 on a Sunday. I have a 12:16 departure from Norwich. Would you like me to book this for you? I'm not sure if I need to book just yet. Can you tell me the travel time for this please? Sure, TR5225 is a 79 minute trip and each seat costs 14.08 pounds. Can I help you with anything else today? Is there an All Saint's Church? I would like to visit it as well. Yes, all saints church is located t city cenre. It has free entrance. That sounds wonderful. Could I get the phone number along with the attraction type? It is listed as architecture, and their phone number is 012345678 Thank you so much! That is all the help I need for now. You are very welcome! Have a great day and I hope you enjoy your visit! Call for help anytime.	
LLM prediction	**score**
A person is seeking help to book a train ticket from Norwich to Cambridge on a Sunday at 12:16, and the agent provides information on the travel time, ticket cost, and the location of All Saints Church in the city center, with free entrance and a listed phone number for architecture enthusiasts.	0.83
A person is seeking information about trains from Norwich to Cambridge and asks about travel time, seat prices, and the location of All Saints Church, and the customer service representative provides the necessary information and offers to book the ticket.	0.61
The text is a conversation between a person looking for train information from Norwich to Cambridge and a train information provider, who provides information on train schedules, fares, and nearby attractions.	0.12
Ground truth	
The user asks duration of the train. This train leaves at 12:15 on sunday, departure is norwich and destination is cambridge. The user wants to know the phone number and type of the attraction. the attraction's name is All Saints Church.	

Table 4: Qualitative example: TodSum

Input question	
In the novel, "Nicholas Nickelby", by Charles Dickens, what was the name of the school, run by Wackford Squeers?	
LLM prediction	**Score**
Dotheboys Hall	0.55
Dotheboys	0.41
Squeers School	0.29
Ground Truth	
Dotheboys Hall	
Input question	
Who was the British Admiral who died in 1707 when four of his ships were wrecked in the Scilly Isles?	
LLM prediction	**Score**
Sir Cloudesley Shovell	0.68
Russell	0.45
Viscount Nicholas Boyle	0.39
Ground Truth	
[Cloudesley Shovell, Sir Cloudesley Shovell]	

Table 5: Qualitative examples: TriviaQA

The following is a conversation between two individuals. Provide a brief summary in [LENGTH] sentence(s). Output the summary only. Input example: [Input example] Output example: [Output example] Input: [INPUT]

Table 6: Prompt template for Abstractive Summarization

The following input is a question from an open domain Question-and-Answering task. Provide a succinct answer to the question in a single phrase (1-3 words). In addition, provide supporting reasons step by step in the following format: Input example: Who was the first man to walk on the Moon? Output example: Answer: Niel Armstrong. Reasoning: Neil Armstrong became the first human to walk on the moon during NASA's Apollo 11 mission on July 20, 1969. This historic event is well-documented through photographs, videos, audio recordings, and historical records, providing irrefutable evidence of his achievement. Input: [INPUT]

Table 7: Prompt template for QA

The following input consists of generated predictions from a Large Language Model(LLM). Besides standard criteria like correctness and helpfulness, take semantic stability into account: We prefer the candidate that is semantically closer to the majority of predictions. Please choose exactly one best prediction, and output the item number (For example "(8)"). If there are multiple identical best answers, choose a random one. Input: [<TASK CONTEXT> Candidates: (1) ... (2) ... (k) ...]

Table 8: Prompt template for LLM Self-rerank