Identifying Factual Inconsistency in Summaries: Large Language Model is Ready to Help

Anonymous ACL submission

Abstract

Factual inconsistency poses a significant hurdle for the commercial deployment of abstractive summarizers. Under this new era of Large Language Model (LLM), this work focuses around two important questions: what is the best way to leverage LLM for factual inconsistency detection, and how could we distill a smaller LLM with both high efficiency and efficacy? Three zero-shot paradigms are firstly proposed and evaluated across five diverse datasets: direct inference on the entire summary or each summary window; entity verification through question generation and answering. Our experiments suggest that LLM itself is capable to resolve this task directly under the correct paradigm design, which surpasses the baselines by up to 4.7% on average. To further promote efficiency for practical use, we then propose training strategies to distill smaller open-source LLM that learns to score the entire summary at once with high accuracy, which outperforms the zero-shot approaches by much larger LLM, serving as an effective ready-to-use scorer.

1 Introduction

Pretrained generative models such as BART (Lewis et al., 2020) have boosted the fundamental development of abstractive summarization ever since. Nevertheless, it is well aware that summaries from those systems could be prone to factual inconsistency, where certain facts presented in the summary are not mentioned in or not consistent with the original document (Maynez et al., 2020; Kryscinski et al., 2020). Previous works to detect factual inconsistency mostly encompass BERT-variants (Devlin et al., 2019) to perform reasoning. Particularly, two main directions arise with state-of-the-art performance: approaches represented by SummaC (Laban et al., 2022a) that adopt trained Natural Language Inference (NLI) models to score the entailment between the document and summary; approaches represented by QuestEval (Scialom et al., 2021) and QAFactEval (Fabbri et al., 2022), which first select entities in the summary to be verified, then utilize Question Generation (QG) models on the summary to generate questions for each entity, and finally employ Question Answering (QA) models on the document to verify whether their answers match the corresponding entities in the summary.

As indicated by previous works, this task is positioned heavily towards document understanding and reasoning, requiring models with strong capabilities. It is naturally occurring under this new Large Language Model (LLM) era: (1) how could we leverage LLM's powerful reasoning abilities for this task, and how good it can be? (2) can we have a smaller and practical LLM model for this task with both efficiency and efficacy in mind?

For the first question, Sec. 3 adapts the core ideas of previous works, and proposes three paradigms to perform zero-shot reasoning by LLM. Specifically, two of the paradigms, Summ-NLI and Sent-NLI, resemble NLI-based approaches that directly reason on a (document, summary) pair, where Summ-NLI determines their factual consistency at once, but Sent-NLI is applied on each summary window then aggregates the final judgement. The third paradigm QG-QA adopts the explicit entity verification, performing zero-shot QG and QA, based on our unique design of entity types, question forms, verification criteria, and decomposed reasoning steps.

The three zero-shot paradigms are evaluated on five diverse datasets using LLM of different models and sizes (Sec. 3.4). Empirical results suggest that, LLM itself is capable enough to identify factual errors directly, and highlight the importance of the correct paradigm design. Particularly, Sent-NLI and QG-QA by ChatGPT leads up to 4.7% upon baselines with sophisticated components. The best open-source LLM can also come close to ChatGPT, with only 1.7% gap by Sent-NLI. Impressively, adopting GPT-4 directly boosts more than 10% improvement on each paradigm, which substantiates that more powerful LLM with prompt designs may become the simple and effective task solution in near future (Sec. 3.5).

Our aforementioned second question is motivated by two remaining problems. Firstly, although our zero-shot approaches achieve strong results, the best performance is obtained by Sent-NLI and QG-QA, which operates on summary windows and then aggregates, being a less efficient and practical option compared to Summ-NLI that scores at once. Secondly, the best zero-shot result requires either the closed-source OpenAI, or large open-source LLM that is also not efficient nor practical to use.

To resolve the second question, Sec. 4 seeks to enable smaller open-source LLM that scores in the same efficient way as Summ-NLI, while maintaining relatively high accuracy. To this end, we propose strategies to train Llama-2 7B models (Touvron et al., 2023) that learn from gold labels, as well as distilling from the available reasoning of a more capable model. Based on our strategies, the trained model successfully outperforms both Summ-NLI and Sent-NLI by ChatGPT with large margins, while being much more efficient. Distilling from reasoning in training is also shown 2% robust improvement for both in-domain and out-of-domain evaluation (Sec. 4.2). Overall, our contributions can be summarized as follows:

- Three zero-shot paradigms are proposed and evaluated that leverage LLM to directly identify factual inconsistency in summaries.
- Experiments on five datasets with multiple models and sizes corroborate that LLM itself is capable to tackle this task, while also highlighting the importance of the paradigm design.
- We further present smaller open-source models of both high efficiency and efficacy through our proposed training strategies, serving as an independent and practical substitution of large LLM.

2 Related Work

Multiple datasets to evaluate factual inconsistency detection in summaries have been independently introduced in recent years, e.g. Goyal and Durrett (2021), SummEval (Fabbri et al., 2021), FRANK (Pagnoni et al., 2021), CLIFF (Cao and Wang, 2021), DiaSumFact (Zhu et al., 2023), LongEval (Krishna et al., 2023), etc. Each dataset may focus on its own types of factual errors, thus they are usually not completely comparable. Recent work has also attempted to unify those factual error types by defining a fine-grained schema (Tang et al., 2023). In this work, we choose five datasets that focus on a similar set of explicit error types (Table 1), such as entity errors, coreference errors, predicate errors, etc. Our experimental datasets include multiple domains: news, dialogues, official documents, stories, and their summary lengths also vary significantly.

Previous state-of-the-art works for this task can be mainly categorized into two directions. NLI-based approaches, such as Falke et al. (2019), SummaC (Laban et al., 2022b), utilize existing NLI models to score the level of entailment between the document and summary to determine their factual consistency. While for QA-based approaches, such as QuestEval (Scialom et al., 2021) and QaFactEval (Fabbri et al., 2022), take explicit entities appeared in summaries, and verify their context on the document through separate QG and QA steps. Apart from these two directions, other works have also explored to recognize factual errors through information extraction (Nan et al., 2021) and syntactic dependencies (Goyal and Durrett, 2021). In this work, our zero-shot paradigms are proposed and established on the ideas of two main directions.

3 Approach: LLM Zero-Shot

Figure 1 illustrates three zero-shot paradigms we propose to identify factual errors in summaries, including wrong entities or predicates, coreference or logical errors, etc. We adopt zero-shot prompting, as we found few-shot examples could introduce bias and do not contribute consistently. Instead, certain efforts are spent on refining prompts for each paradigm, aiming to outline full criteria, clear instructions and thought process to guide the LLM reasoning. Chain-of-Thought (CoT) (Wei et al., 2022) is enforced for all prompts. Full prompts of each paradigm are provided in Appx. D.

3.1 Summ-NLI

The most straightforward way to integrate LLM is to directly ask it to score or classify the given document and summary pair according to their factual consistency. This resembles previous works employing NLI models to determine their level of entailment. As LLM takes the entire (document, summary) pair and scores at once, regardless of the summary length, we dub this paradigm as Summ-NLI (Summary-level NLI).

One pilot work (Wu et al., 2023) has also conducted a study similar to Summ-NLI, which asks

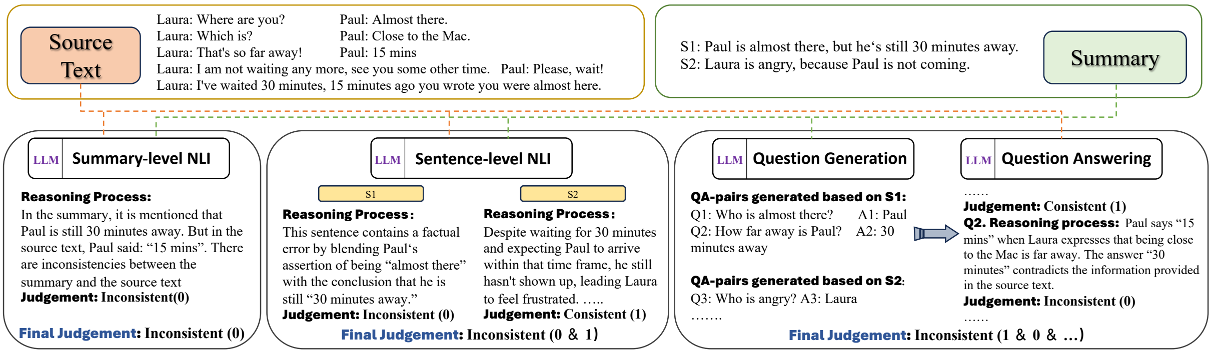

Figure 1: Illustration of our three zero-shot paradigms: Summ-NLI, Sent-NLI, QG-QA (Sec. 3).

GPT-4 to directly score the entire summary. For easy comparison, we keep our Summ-NLI in a similar fashion by laying out the factual error types and reasoning steps in the prompt, effectively regarding Summ-NLI as an enhanced baseline.

We instruct Summ-NLI to either yield a binary label in the end (whether the summary has any factual errors), or to produce a consistency score within a range, according to the specific dataset. Let S_i be the ith summary, D_i be its corresponding document. The output y_i can be denoted as:

$$y_i^{\text{Summ-NLI}} = \text{LLM}\left(D_i, S_i\right) \quad (1)$$

3.2 Sent-NLI

Apart from scoring the entire summary at once, one could also score by each summary sentence, then aggregate. The intuition is simple: when the summary gets longer, LLM might overlook certain errors scattered across many sentences, which is indeed a mistake humans can make. Thus, we adjust Summ-NLI to operate on a window each time that consists of one or a few summary sentences. For brevity, we dub as Sent-NLI (Sentence-level NLI) .

To aggregate each window, we consider the entire summary having factual errors if any of its windows has errors. Let W_{ij} be the jth window of summary S_i that has $\xi(S_i)$ windows in total, the output y_i can be denoted as:

$$y_i^{\text{Sent-NLI}} = \text{any}_{j=1}^{\xi(S_i)}\left(\text{LLM}\left(D_i, W_{ij}\right)\right) \quad (2)$$

3.3 QG-QA

Distinct from NLI-based approaches that directly evaluate the factual consistency, previous QA-based approaches such as QAFactEval (Fabbri et al., 2022) take a more fine-grained way that utilize existing QG and QA models to verify explicit entities in summaries. However, they require careful tuning of multiple pipeline components, including answer selection and answer checking, in addition to the question generation and answering modules. We design our QG-QA paradigm on LLM as follows, integrating answer selection and checking into the following two phases.

Question Generation For this phase, LLM is only conditioned on a summary window without accessing the document; the primary goal is to create (question, entity) pairs from the summary window, to be verified on the document later.

Entities include named entities, e.g. person, locations, products, as well as general noun phrases. We confine the corresponding question to be *wh*-question in a *subject-verb-object* structure, such that the answer to the question is the entity itself. To induce high-quality pairs, we facilitate LLM reasoning by a three-step decomposition:

1. As commonly there are pronouns in the summary window, coreference resolution is firstly performed by LLM on the entire summary context, thereby (question, entity) pairs will always present explicit entities with no ambiguity. By contrast, most previous approaches neglect this step, e.g. QAFactEval produces pronouns as entities, which could be troublesome during verification due to its ambiguity.
2. Important entities within this summary window are then explicitly listed, where each entity is unique, and can be compound entity such as "Mike and Amy".
3. Generate (question, entity) pairs based on the listed entities, according to certain rules, such as one question per entity, no pronouns as entities, no open-ended question, etc.

Above steps are written as instructions in the prompt that LLM is expected to follow. After we gather all pairs from all summary windows, we move to the next phase.

	# Summaries	# Tokens	Domain	Label	Metric	Error Types
FRANK	175	59.0	News	Binary (85.7%)	Balanced Accuracy	*Relation Error, Entity Error, Coreference Error, Circumstance Error, Out-of-Article Error*
DiaSumFact	475	43.7	Dialogues	Binary (43.0%)	Balanced Accuracy	*Entity Error, Coreference Error, Circumstance Error, Predicate Error*
CONFIT	600	17.8	Dialogues	Binary (62.3%)	Balanced Accuracy	*Circumstantial Error, Negation Error, Object Error, Wrong Reference Error*
GovReport	204	397.2	Official Reports	Score	Pearson Correlation	*Entity Error, Coreference Error, Circumstance Error, Out-of-Article Error*
SQuALITY	40	347.3	Stories	Score	Pearson Correlation	*Correctness*

Table 1: Statistics and evaluation metrics for our five experimented datasets, including the number of summaries, averaged number of tokens per summary, and their label types and error types. The ratios of positive labels (no factual errors) are shown in parenthesis. For FRANK, we use summaries generated from BART for CNN/DM.

Question Answering With all pairs obtained, this phase prompts LLM the second time to verify each pair based on the document. Note that this should be a new LLM session rather than continuing from QG, so that the QA phase is free from any interference by the original summary.

By providing the document and listing the (question, entity) pairs, we instruct LLM to do the following for each pair: first reason the question based on the document, and then check whether its reasoned answer is consistent with the provided entity from the summary. In contrast to previous works that use lexical overlap or cosine similarity for answer checking, we can now simply shift it to LLM to judge their consistency.

Overall, QG-QA guides LLM to recognize factual errors through explicit entities. In this process, predicate errors and logical errors will be detected as well: a question with the wrong predicate or logic should not have an answer by the document, thus not consistent with the provided entity.

Let summary S_i have $\zeta(S_i)$ pairs, and (q_k, e_k) be its kth pair, QG-QA can be represented as:

$$\{(q_k, e_k)\} = \text{LLM-QG}\ (S_i) \quad (3)$$

$$y_i^{\text{QG-QA}} = \text{any}_{k=1}^{\zeta(S_i)} \left(\text{LLM-QA}\ (D_i, q_k, e_k)\right) \quad (4)$$

3.4 Zero-Shot Experiments

Our experiments are conducted on five datasets of diverse domains and summary length. Table 1 provides an overall summary of all the datasets.

Three of the datasets have regular-length documents with binary labels for factual errors:

- **FRANK** (Pagnoni et al., 2021) consists of news articles and their summaries. Here we evaluate a subset of summaries generated by BART for CNN/DM (Hermann et al., 2015).
- **DiaSumFact** (Zhu et al., 2023) consists of dialogue documents and their system-generated summaries, including daily conversations for SAMSum (Gliwa et al., 2019), and meeting transcripts for QMSum (Zhong et al., 2021).
- **CONFIT** (Tang et al., 2022) contains factual annotations for SAMSum summaries as well. We use summaries generated by all six models.

Two of the datasets have long document length with factual consistency scores as labels:

- **GovReport** (Koh et al., 2022) consists of official reports and summaries for Huang et al. (2021), and each report has on-average 3.8k words.
- **SQuALITY** (Krishna et al., 2023) consists of stories and long articles with summaries for Wang et al. (2022), with ∼5k words per document.

Evaluation Protocol Though each dataset annotates its own factual error types, these types are largely shared (Tang et al., 2023), and Table 1 lists the corresponding error types our experiments considered. We exclude grammar errors as factual errors, as they still largely convey consistent factual information. For the first three datasets, each summary has a binary label indicating whether it has any factual errors, and we report **Balanced Accuracy** for evaluation. For the latter two, each label is a score within a range, and we report **Pearson Correlation** following previous works.

Baselines We adopt two QA-based approaches as our baselines: QuestEval (Scialom et al., 2021) and QaFactEval (Fabbri et al., 2022), where QaFactEval is considered the state-of-the-art prior to LLM (Tang et al., 2023). We use the code released

	Balanced Accuracy			Pearson Correlation		
	FRANK	**DiaSumFact**	**CONFIT**	**GovReport**	**SQuALITY**	*Macro-Average*
QuestEval	62.67	56.03 / 56.91*	59.50 / 59.78*	26.90	42.21	49.46
QaFactEval	53.00	**67.29** / 68.79*	56.34 / 59.90*	40.59	44.79	52.40
Summ-NLI	51.33	58.89	65.59	19.95	35.46	46.24
Sent-NLI	61.42	65.83	62.12	46.46	**49.76**	57.12
QG-QA	**63.35**	64.57	**67.51**	**50.17**	40.03	**57.13**

Table 2: Evaluation results on the five datasets, along with their macro-average scores as the overall evaluation metric. Sent-NLI and QG-QA outperform other approaches through our designed zero-shot prompting paradigms. *: since two baselines QuestEval and QaFactEval require a threshold to convert the scores to classification labels, while DiaSumFact and CONFIT do not have a development set to tune the threshold, we adopt the same threshold tuned upon FRANK for the two datasets. As the original paper (Zhu et al., 2023) tunes against the test set itself, which could become over-optimistic, we also provide scores tuned upon the test set for comparison (marked by *).

by their authors for the implementation. Besides, Summ-NLI can be regarded as a baseline with LLM, as it is straightforward and has been explored by previous works (Wu et al., 2023).

LLM We use *gpt-3.5-turbo-0613* (ChatGPT) for our main experiments. For analysis, we additionally run GPT-4 (*gpt-4-1106-preview*), Llama-2[1] models (Touvron et al., 2023), Vicuna[2] models (Zheng et al., 2023) on DiaSumFact. For GovReport and SQuALITY, we follow Wu et al. (2023) that for each question, top sentences that maximize ROUGE scores towards each summary are retrieved as the context for factual evaluation, up to 1k tokens per document (details in Appx. A).

Results Table 2 shows the evaluation results on all five datasets, and we use their macro-average scores as the main evaluation metric. Sent-NLI and QG-QA are shown to obtain similar results, and both outperform the two non-LLM baselines by up to 7.6%, which demonstrates that **LLM itself is capable enough to identify factual errors directly**, ascribed to its superior understanding and reasoning ability. Nevertheless, Summ-NLI that adopts the same LLM as well underperforms those two baselines, indicating **the importance of the correct zero-shot paradigm design**, which can play a significant role for the task performance.

Comparing the three paradigms, the gap between Summ-NLI and the other two gets larger for GovReport and SQuALITY that have longer summaries (Table 1). This observation may not be surprising, as Summ-NLI only scores once regardless the length of the summary, being a more efficient option but potentially prone to more errors.

As both Sent-NLI and QG-QA achieve strong results, QG-QA obtains comparable or better performance than Sent-NLI on all datasets except for SQuALITY. Thus, the paradigm to explicitly verify entities not only leads to state-of-the-art performance for pre-LLM approaches, and is proved still valid under the LLM era. However, QG-QA does run LLM twice per summary window, bringing more overhead than Sent-NLI, which could make Sent-NLI more appealing in practice.

Despite the trivial trade-off between Sent-NLI and QG-QA, both paradigms are window-based approaches that are less efficient than Summ-NLI. In Sec. 4, we further seek to train open-source LLM with our proposed training strategies, combining both the efficiency from Summ-NLI and the efficacy from Sent-NLI.

3.5 Zero-Shot Analysis

We focus on DiaSumFact and perform further analysis over multiple dimensions as follows.

Varying LLMs and Sizes As models from OpenAI are known among the best models, Table 3 compares zero-shot results using different LLM, including GPT-4, and the open-source Llama-2 and Vicuna models of multiple sizes. Neither Llama-2 nor Vicuna could outperform ChatGPT; though, their largest models do come close by Sent-NLI. The results suggest that adopting Sent-NLI accompanied by Vicuna 13B can serve as a good zero-shot alternative to ChatGPT for this task.

Clearly, just by using GPT-4, there comes a direct performance boost upon ChatGPT by over 10% for each paradigm, which is quite impressive. This

[1]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[2]https://huggingface.co/lmsys/vicuna-7b-v1.5-16k

	Summ-NLI	Sent-NLI	QG-QA
Llama-2 7B	53.49	51.16	53.16
Llama-2 13B	52.13	54.64	53.13
Llama-2 70B	53.74	63.21	58.02
Vicuna 7B	53.15	53.27	50.12
Vicuna 13B	56.62	64.11	58.64
ChatGPT	58.89	65.83	64.57
GPT-4	**70.09**	**76.63**	**74.78**

Table 3: Evaluation results on DiaSumFact using LLMs of different models and sizes. Sent-NLI with Vicuna 13B achieves the best non-OpenAI performance.

	Summ-NLI	Sent-NLI	QG-QA
Llama-2 13B	± 3.16	± 1.60	± 2.01
Vicuna 13B	± 1.80	± 3.10	± 3.37
ChatGPT	± 1.34	± 0.12	± 1.04

Table 4: Standard deviation of different models and paradigms on DiaSumFact from three repeated runs.

indicates that a more powerful LLM with better reasoning abilities might be the simple and effective solution for the future of factual error detection.

Comparing Llama-2 and Vicuna, Vicuna 13B is able to outperform Llama-2 70B by each paradigm, proved as an outstanding open-source candidate for zero-shot evaluation. Nonetheless, increasing the size of LLMs lead to higher evaluation scores for both models. Especially, Sent-NLI and QG-QA benefit more than Summ-NLI, advocating again that evaluation by windows can be more effective than scoring the entire summary.

To inspect the zero-shot variation of different models, Table 4 further shows the standard deviation of three repeated runs for Llama-2 13B, Vicuna 13B, and ChatGPT on DiaSumFact. These variation comes from imperfect instruction following and inconsistent answers on ambiguous cases. ChatGPT exhibits smallest variation among three models. Hence, although open-source models can come close, ChatGPT still remains the preferable model owing to its stable performance.

Paradigm Comparison Figure 2 plots the performance curve on different lengths of documents and summaries with ChatGPT. Overall, Sent-NLI and QG-QA follow similar trend: both are quite robust to different document lengths. Understandably, Summ-NLI is shown to suffer degradation on longer documents or summaries, due to its length-agnostic scoring mechanism. Though, all

	None	Ent.	Circ.	Pred.	Coref.
Sent-NLI	80.96	47.97	54.26	16.50	26.67
QG-QA	77.28	42.08	41.49	43.50	30.84

Table 5: Recall of error types by Sent-NLI and QG-QA on DiaSumFact: No Errors, Entity Errors, Circumstantial Errors, Predicate Errors, Coreference Errors.

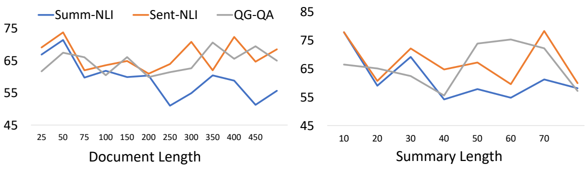

Figure 2: ChatGPT accuracy on the first three datasets of different lengths of documents and summaries.

paradigms struggle to keep up the performance on long summaries (length > 75).

For fine-grained analysis on error types, Table 5 discloses the recall of different types by Sent-NLI and QG-QA on DiaSumFact. Both paradigms recover entity, circumstantial and coreference errors similarly, while QG-QA recognizes more predicate errors than Sent-NLI, albeit they only constitute a small portion of factual errors. Overall, QG-QA is shown slightly more balanced than Sent-NLI.

Strengths and Limitations For qualitative analysis, more concrete examples are provided in Appx. B to illustrate the strengths of LLM paradigms, as well as their current limitations.

4 Approach: Distilling Efficient Scorers

As suggested by Table 2, Sent-NLI and QG-QA obtain strong performance and surpass other approaches by good margins. However, it does come with the overhead to evaluate on each summary window; ideally, one would prefer a model that only scores once per summary, just like Summ-NLI, and still achieves similar or better performance than Sent-NLI. Furthermore, the best performance requires either the closed-source OpenAI, or large LLM that is not efficient for practical use.

Motivated by above issues, we then resort to distill smaller open-source LLM models that learn to score in the same way as Summ-NLI, aiming to serve as an efficient, effective, and independent substitution of Sent-NLI or QG-QA, without being tied to any OpenAI models. To achieve this, we focus on the classification scenario, and regard the three datasets in Sec. 3.4 with binary labels as training resources. As all the open-source LLM presented in

Table 3 underperform ChatGPT by Summ-NLI, we further propose to leverage the available reasoning of ChatGPT from previous experiments into training, which could facilitate to distill useful knowledge from the more powerful ChatGPT, in addition to simply learning the task labels themselves. Concretely, the training data comprises the following two types of prompts.

4.1 Distilling Strategies

Prompts with reasoning If a (document, summary) pair has been processed by ChatGPT (or GPT-4) in existing experiments, and the classification given by ChatGPT is correct, then the prompt for this pair is the same as Summ-NLI, where it instructs the model to perform reasoning first and then give the final label. During training, the open-source LLM learns to generate the same reasoning as ChatGPT that leads to the correct label, then to produce the final label in the end. The reasoning to be learned may come from two sources: if ChatGPT answers correctly by Summ-NLI, we then extract its reasoning as partially the training output; otherwise, if ChatGPT could answer correctly by Sent-NLI, then we extract its reasoning of each summary window and concatenate them together, serving as the summary-level reasoning that is consistent with the final label.

Prompts without reasoning The prompt for this type is still largely similar to Summ-NLI, except that now it explicitly instructs to directly produce the classification label without any reasoning; the output to be learned is then the gold label from the dataset. This type of prompts resembles the conventional supervised classification paradigm, where the model learns to classify directly. It can apply to any (document, summary) pairs, whether it has been processed by ChatGPT or not.

Combining the above two types of prompts in training, the trained model is exposed to the reasoning process for this task from a more capable model. Moreover, for pairs processed by ChatGPT correctly, we provide both two prompts (with and without reasoning) in training, which provides contrastive examples to assist recognizing reasoning and the inference of labels. Overall, the model distills task knowledge and learns to detect factual errors more robustly, corroborated in Sec. 4.2.

The inference now also becomes flexible: the model could either opt to perform reasoning first, or to produce the label directly.

4.2 Training Experiments

For training, Llama-2 7B is used as the backbone model. We conduct experiments with three different strategies on reasoning:

- `T-wo-R` + `I-wo-R`: the model is **T**rained **w**ith**o**ut any **R**easoning; consequently, the **I**nference is also performed without reasoning.
- `T-w-R` + `I-w-R`: the training is assisted by reasoning, and inference also performs reasoning.
- `T-w-R` + `I-wo-R`: the model receives reasoning in training, but directly yields classification labels during inference (faster inference than `I-w-R`).

For each strategy, we further conduct two sets of experiments as follows, positioned to evaluate the task capability of trained models, as well as the transfer capability on unseen domains respectively.

In-Domain Evaluation We randomly split 80% documents in FRANK, DiaSumFact and CONFIT as the training set, and the other 20% for evaluation. We adopt common hyperparameters for LLM finetuning, described in Appx. A, without requiring a development set due to the limited data. Additionally, we add new documents and summaries from FRANK not evaluated by ChatGPT before into training, in the form of prompts without reasoning. Specifically for DiaSumFact, as reasoning of GPT-4 is available from Sec. 3.4, we use its reasoning instead of ChatGPT's. Based on the approach design in Sec. 4, prompts with reasoning constitute 28% in-domain training examples.

Out-of-Domain Evaluation In practical scenarios, the trained model may be used on domains that are much more diverse than those seen in training. To assess the performance under domain shift, we perform the out-of-domain (OOD) evaluation, where the model is trained on the entire DiaSumFact and CONFIT, which comprise dialogue documents, and then evaluated on FRANK that consists of news documents. For this setting, prompts with reasoning constitute 40% total training examples.

	# Train	Length	R-Ratio	# Test	Length
ID	2918	42.0	28%	246	35.3
OOD	1801	29.2	40%	175	59.0

Table 6: Statistics for In-Domain (ID) and Out-of-Domain (OOD) evaluation: number of training examples; averaged number of training summary tokens; ratio of prompts with reasoning; number of evaluation examples; averaged number of evaluation summary tokens.

		In-Domain				Out-of-Domain
		FRANK	**DiaSumFact**	**CONFIT**	*Average*	**FRANK**
Summ-NLI	ChatGPT (Zero-Shot)	46.43	57.55	63.33	55.77	51.33
	Llama (`T-wo-R` + `I-wo-R`)	58.93	**76.64**	64.23	66.60	52.67
	Llama (`T-w-R` + `I-w-R`)	57.14	63.80	58.37	59.77	50.67
	Llama (`T-w-R` + `I-wo-R`)	**62.50**	75.15	**68.36**	**68.67**	**54.67**
Sent-NLI	ChatGPT (Zero-Shot)	55.36	67.30	64.00	62.22	61.42

Table 7: Results of our trained Llama-2 7B models for both in-domain and out-of-domain evaluation. The **T**raining can be either assisted with or without **R**easoning from ChatGPT (`T-w-R` or `T-wo-R`); the **I**nference can also opt to perform **R**easoning or not (`I-w-R` or `I-wo-R`). Details of experimental settings are described in Sec. 4.2.

Table 6 shows a brief summary of our experimental data. Especially, the summary length almost doubles from training to testing in OOD evaluation. Though the training data does not seem plentiful, our objective is not to build a model with the best possible performance by scraping all existing resources for training; rather, we aim to examine the training strategies and propose an effective and robust direction to distill smaller models.

Results Table 7 shows the evaluation results on training Llama-2 7B by different strategies to directly perform Summ-NLI. For in-domain evaluation, all three trained models outperform Summ-NLI by ChatGPT, with the best setting leading up to 12.9%, after undergoing the training process. It is worth noting that two of the settings also outperform Sent-NLI by up to 6.5%, successfully fulfilling the goal to build **open-source models of both superior efficiency and efficacy** than the zero-shot paradigms of Sent-NLI and QG-QA. Our trained models will be openly released to researchers.

Comparing the three strategies, the best performance is achieved by `T-w-R` + `I-wo-R` for both ID and OOD evaluation. By receiving reasoning in training, it surpasses its counterpart (`T-wo-R` + `I-wo-R`) by 2% robustly for both ID and OOD, which validates our hypothesis to assist training through reasoning. By contrast, there is a noticeable degradation when performing reasoning during inference (`T-wo-R` + `I-wo-R`), which can be attributed by the fact that the available reasoning in training is still relatively sparse to get fully learned; when the model performs reasoning with lower quality, it could impair the inference of the labels and bring more negative impact than the positive. Overall, Table 7 suggests `T-w-R` + `I-wo-R` to be the best strategy, being the most performant and also the fastest option during inference.

Quantitative Comparison Observed in Sec. 3.4, Summ-NLI suffers more degradation when the summary gets longer in the zero-shot setting. For the trained Llama-2 models, we also plot the performance towards different summary lengths, comparing the zero-shot approaches and trained models, as shown in Figure 3. Similar to Figure 2, all approaches indeed still perform less for summary length > 75. However, both two trained models are able to keep up the performance till length 60. More importantly, they outperform Summ-NLI and Sent-NLI on almost all summary lengths, which demonstrates that open-source models can score the entire summary at once with high accuracy through our proposed training strategies.

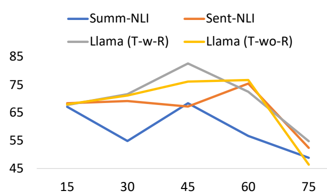

Figure 3: In-domain evaluation of different summary lengths, with approaches: Summ-NLI and Sent-NLI by zero-shot ChatGPT, Llama-2 w/ and w/o reasoning.

5 Conclusion

In this work, we propose and evaluate three zero-shot paradigms on leveraging LLM for factual error detection in summaries. Empirical results on five datasets suggest that LLM itself is capable to resolve this task directly, and highlight the importance of the correct paradigm design. To be more practical, we further propose effective training strategies to score the entire summary at once by smaller open-source LLM. Our models learn from both gold labels and reasoning, additionally outperform ChatGPT by large margins, combining both efficiency and efficacy for practical use.

Limitations

While our study demonstrates the potential of leveraging LLMs for detecting factual inconsistencies, it is important to acknowledge certain limitations. The zero-shot paradigms, though effective, may not fully capture the nuances of complex summaries. We exclusive list certain limitations and concrete qualitative examples on the failure cases by LLM; please see Appx. C. In summary, though LLM is indeed capable understanding and reasoning well, it cannot follow the instructions perfectly, resulting in occasional low-quality questions or entities, and wrong inference, which is especially more severe for smaller open-source LLM (such as 7B models).

For distilling smaller open-source models, though learning from reasoning is shown helpful, the training indeed depends on the availability of those labels and reasoning, which can be limited, thus confining its final performance, unlike the zero-shot paradigms which itself do not require any additional resources.

References

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Huan Yee Koh, Jiaxin Ju, He Zhang, Ming Liu, and Shirui Pan. 2022. How far are we from robust long abstractive summarization? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2682–2698, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. LongEval: Guidelines for human evaluation of faithfulness in long-form summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1650–1669, Dubrovnik, Croatia. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022a. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022b. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733, Online. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Xiangru Tang, Arjun Nair, Borui Wang, Bingyao Wang, Jai Desai, Aaron Wade, Haoran Li, Asli Celikyilmaz, Yashar Mehdad, and Dragomir Radev. 2022. CONFIT: Toward faithful dialogue summarization with linguistically-informed contrastive fine-tuning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5657–5668, Seattle, United States. Association for Computational Linguistics.

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. 2023. Llama 2: Open foundation and fine-tuned chat models.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022. SQuALITY: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Yunshu Wu, Hayate Iso, Pouya Pezeshkpour, Nikita Bhutani, and Estevam Hruschka. 2023. Less is more for long document summary evaluation by llms. *arXiv preprint arXiv:2309.07382*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli

Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Rongxin Zhu, Jianzhong Qi, and Jey Han Lau. 2023. Annotating and detecting fine-grained factual errors for dialogue summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6825–6845, Toronto, Canada. Association for Computational Linguistics.

A Experimental Settings

Long Document Alignment As documents in both GovReport and SQuALITY have long length of thousands of tokens, alignment is firstly performed, such that for each summary or summary window, related sentences from the document are retrieved, which will be used as a shorter context for factual error evaluation.

For Summ-NLI, top sentences from the document that maximize the recall of ROUGE-1 and ROUGE-2 towards the summary are retrieved, until the total length reaches a certain threshold. These sentences are concatenated as the new context, which is shorter but has higher information density than the original document.

For Sent-NLI and QG-QA that operates on summary windows, n important sentences are independently extracted to maximize the recall of ROUGE-1 and ROUGE-2 towards the summary. Table 8 shows the alignment thresholds we adopted for the two datasets.

	Summ-Alignment	Sent-Alignment (n=5)
GovReport	1024	102.31
SQuALITY	1024	28.50

Table 8: The maximum length of aligned context for Summ-NLI, and the averaged length of aligned context per summary window for Sent-NLI and QG-QA. n is the number of sentences extracted for each summary window. For SQuALITY, some of the retrieved sentences can be quite short.

Evaluation for Baselines Both two baselines QuestEval and QaFactEval produce a score for each summary, which requires a threshold to convert to the classification label. For FRANK, we use its development set provided by the dataset to tune the threshold for the baselines, where we perform 5-fold cross-validation to obtain the best threshold. For DiaSumFact and CONFIT, there is no dedicated development set provided. Hence, we use the thresholds tuned on FRANK and apply them on these two datasets. For GovReport and SQuALITY, the evaluation metric is Pearson Correlation, thus without requiring any thresholds.

Experimental Settings for Training We perform full finetuning of Llama-2 7B models with flash attention enabled on 6 Nvidia A100 GPUs with 40GB memory each. The training runs for 8 epochs, taking around 2 hours to finish for `T-wo-R` and 5 hours for `T-w-R`. We use a batch size of 1 per GPU, learning rate of 1×10^{-5}, and cosine learning rate schedule with warm up rate 0.05. Loss is only enabled for the completion during training.

B Qualitative Examples

Advantages of LLM LLM is able to perform QA with greater understanding than the traditional non-LLM baselines, leading to better performance.

[Part of the source text]
Adam: Have you talked to May?
Karen: Yes, yesterday, why?
Adam: I just talked to her and I must admit I worry about her.
Karen: Me too, May doesn't want to see a specialist.
Adam: No wonder...
Karen: I know, but I think this is serious. She's saying she's depressed, like everyone around, but in her case it may be true.
......
Karen: Honestly I don't know if we can help her, Adam. I suggested a specialist because these are very sensitive issues and I'm afraid we may unintentionally make it worse.
Adam: Yes, but she doesn't want to see a specialist. Basically, she doesn't want to see anyone.
Karen: Hm... I don't know... How about I call someone for advice? So we could know what to do. ......

[Part of the summary]
May doesn't want to see a specialist.

[Part of the response of QAFactEval]
"entity": "a specialist"
"question": "Who does May not want to see?"
"prediction": "I have a friend who's a psychologist"
"lerc_quip": 0
"Judgement": Inconsistent

[Part of the response of LLM (QG-QA)]
"entity": "May"
"question": "Who doesn't want to see a specialist?"
"Reasoning process": "Karen mentions that she suggested May should see a specialist, but May wasn't happy about it. It indicate that May is the one who doesn't want to see a specialist. Adam also mentions that May doesn't want to see a specialist.
Judgement: Consistent

QG-QA over Sent-NLI QG-QA performs more fine-grained reasoning, able to examine an explicit entity from multiple perspectives, which can be beneficial. In this example, LLM did not find the factual error in the first place, but by answering the second question from another angle, QG-QA identified the error successfully.

[Part of the source text]
Ollie: Hi , are you in Warsaw?
Jane: yes, just back! *Btw are you free for diner the 19th?*
Ollie: nope!
.....

[Part of the summary]
Ollie is in Warsaw.

[Part of the response of LLM (Sent-NLI)]
Reasoning process: 1. The summary states that Ollie is in Warsaw, which is consistent with the source text. 2. The summary states that Ollie will have lunch with Jane on Friday, which is consistent with the source text.
Therefore, the final judgement is consistent.
Judgement: Consistent

[Part of the response of LLM (QG-QA)]
Q1: Who is in Warsaw? E1: Ollie
Reasoning process: In the conversation, Ollie asks Jane if she is in Warsaw, to which she responds ÿes, just back!Ïhis implies that Ollie is the one who is currently in Warsaw.
Judgement: Consistent
Q2: Where is Ollie? E2: Warsaw
Reasoning process: In the conversation, Ollie asks Jane if she is in Warsaw, indicating that he is not in Warsaw. However, there is no explicit information provided about Ollie's current location.
Judgement: Inconsistent

Sent-NLI over QG-QA Sent-NLI is simpler to perform, while QG-QA may have error propagation due to its two separate phases, as shown by the following example.

[Part of the source text]
......
user interface: It's we call it a mushroom design.
user interface: It's looks like some mushroom , so we call it mushroom design.
industrial designer: Yeah. Uh.
user interface: So this is a introduction of our product.
project manager: Genetically modified mushroom I will say, but...
......

[Part of the summary]
the user interface *believed that this was the introduction of their product.*

[Part of the response of LLM (Sent-NLI)]
...... The user interface concludes their description by saying, "So this is an introduction of our product." The summary captures the essence of the user interface's role in the dialogue, which is to introduce the product. ......
Judgement: Consistent

[Part of the response of LLM (QG-QA)]
Q1: Who believed that this was the introduction of their product?
E1: ***The user interface team***
"Reasoning process: The user interface says, "So this is a introduction of our product." However, the dialogue does not mention a "team," only an individual referred to as "user interface.""
Judgement: Inconsistent

C Failure Cases by LLM

Low-quality questions may be generated by QG-QA As LLM itself is not perfect, questions not meeting up the criteria may be generated by QG-QA, which eventually results in failure cases.

[Summary]
industrial designer thought a modern remote control should control a device diverse subset of equipment. for instance, dvd players, cell phones, video and audio equipment. so one remote control for all your equipment. there should be some interoperability between them. industrial designer also suggested some new technical possibilities available and which might be interesting to implement in the product. project manager agreed and asked industrial designer to remind people of some new possibilities which were available and of the technical possibilities and impossibilities.

[Specified sentences]
industrial designer also suggested some new technical possibilities available and which might be interesting to implement in the product.

[Questions generated by LLM (QG)]
The questions generated by LLM:
Q1: Who thought a modern remote control should control a diverse subset of device?
E1: industrial designer ...

Low-quality entities selected by QG-QA LLM sometimes cannot completely follow the instruction, resulting in low-quality entities chosen by LLM.

[Summary]
Callan's Samsung S8 overheats and it takes a while for it to restart again. He's going to the store to have it checked. It's still under warranty.

[Specified sentences]
It's still under warranty.

[Questions generated by LLM (QG)]
The questions generated by LLMs:
Q1: Is the device still under warranty?
E1: Yes
...

Inconsistent reasoning with humans For some cases, even though LLM makes the correct label prediction, its reasoning does not correspond to what humans think.

[Part of the source text]
...
phd d: Uh working in the cepstral domain ? So there is one guy in Grenada
phd b: Yeah, in Grenada one of my friend.
phd d: and another in uh, Lucent that I met at ICASSP.
...

[Part of the summary]

The team said that the person in grenada was working in the cepstral domain.

[Human comment]

the team -> phd d

[Response of LLM]

*The summary states that **the person in Grenada was working in the cepstral domain.** However, there is no mention in the source text about the person in Grenada working in the cepstral domain. This indicates a predicate inconsistency.*

```
Summm-NLI

### Instruction:
Now you are one of the greatest English linguist experts and Logicians. Below is an instruction for evaluating the consistency of a generated summary to the source text. Consistency measures whether a summary is factually consistent with the source text. The goal is to find possible factual errors in the sentences to ensure that people can get the correct information from the summary.

Please consider the following six types of errors while conducting the evaluation:
i) Predicate inconsistency in the summary compared to the source text
ii) The core arguments or their attributes in a semantic frame are wrong, such as the subjects and object
iii) Incorrect circumstantial information in the predicate
iv) A pronoun or a reference has a wrong antecedent or has no antecedents
v) Errors that stray beyond the source text's content
vi) Logic Errors, such as Non Sequitur, Bait and Switch, sequence of events error, treating a possibility as a certainty.

### Evaluation Criteria:
Inconsistent: If the summary contains one of the errors listed above, please reply Inconsistent.
Consistent: If every information point in the sentence can be found in the source text and there are no factual errors, please reply Consistent.

### Evaluation Steps:
1. Read the Source Text: Familiarize yourself with the content and key points of the source text.
2. Analyze the Generated Summary: Carefully examine the generated summary. Look for factual accuracy of information points in summary to the source text.
3. Provide Justification: Support the judgement with specific examples of inconsistencies or alignment between the summary and the source text. This helps justify the evaluation.
4. Identify Factual Inconsistencies: Note any information in the summary that contradicts or misrepresents facts from the source text. This could include incorrect data, misinterpretation of events, wrong relationships between information points (possible-inevitable, cause-and-effect, etc.) or misleading statements.
5. Consider tolerable mistakes: Sometimes, the summary might differ in wording or focus but still convey the same information as the source. In this case, it should be considered that there is no error.
6. Based on your step-by-step reasoning process, reply Consistent or Inconsistent as your final judgement.

--- Your_Task---
### Source Text:
  {source text}

### Generated Summary:
  {summary}

### Reasoning process and final judgement:
  Reasoning process: < Reasoning process >
  Judgement: < Consistent or  Inconsistent >

Please refer to the instruction, then finish Your_Task by showing your reasoning process and final judgement.
```

Figure 4: Prompt for Summm-NLI.

D Full Prompts

Sent-NLI

Instruction:
Now you are one of the greatest English linguist experts and Logicians. Below is an instruction for evaluating the consistency of the specified sentences in a generated summary for the source text. Consistency measures whether a summary is factually consistent with the source. The goal is to find possible factual errors in the sentences to ensure that people can get the correct information from the summary.
.
Please consider the following six types of errors while conducting the evaluation:
i) Predicate inconsistency in the summary compared to the source text
ii) The core arguments or their attributes in a semantic frame are wrong, such as the subjects and object
iii) Incorrect circumstantial information in the predicate
iv) A pronoun or a reference has a wrong antecedent or has no antecedents
v) Errors that stray beyond the source text's content
vi) Logic Errors, such as Non Sequitur, Bait and Switch， sequence of events error， treating a possibility as a certainty.

Evaluation Criteria:
Inconsistent: If the summary contains one of the errors listed above, please reply Inconsistent.
Consistent: If every information point in the sentence can be found in the source text and there are no factual errors, please reply Consistent.

Evaluation Steps:
1. Read the Source Text: Familiarize yourself with the content and key points of the source text.
2. Analyze the Generated Summary: Carefully examine the generated summary. Look for factual accuracy of information points in summary to the source text.
3. Provide Justification: Support the judgement with specific examples of inconsistencies or alignment between the summary and the source text. This helps justify the evaluation.
4. Identify Factual Inconsistencies: Note any information in the summary that contradicts or misrepresents facts from the source text. This could include incorrect data, misinterpretation of events, wrong relationships between information points (possible-inevitable, cause-and-effect, etc.) or misleading statements.
5. Consider tolerable mistakes: First, the summary might differ in wording or focus but still convey the information as the source. In this case, it should be considered that there is no error. Second, it is permissible that the sentence does not contain some of the information points in the source text. All you need to judge is whether the content of the sentence itself is consistent or not.
6. Based on your step-by-step reasoning process, reply Consistent or Inconsistent as your final judgement.

--- Your_Task---
Source text:
{source text}

One of the specified sentence in summary:
{Sentence}

Reasoning process and final judgement:
Reasoning process: < Reasoning process >
Judgement: < Consistent or Inconsistent >

Please refer to the instruction, then finish Your_Task by showing your reasoning process and final judgement.

Figure 5: Prompt for Sent-NLI.

Questions Generation

Instruction:
Suppose you are one of the greatest linguistics professors and English teachers. Now you are asked to list entities and noun phrase chunks and generate question-answer pairs from the given specified sentences. The context around the specified sentences is also provided. Steps to generate questions are shown as follows:

1: Please perform coreference parsing on the "Specified sentences" and replace pronouns with appropriate entity names. If multiple pronouns refer to the same entity, be sure to replace them with the correct entity name.

2: Then list important and complete named entities or noun phrase chunk (such as a person, location, organization, product, etc.) in the specified sentences which are relevant to factual consistency evaluation. Multiple entities connected by a conjunction should be regarded as a complete entity.

3：Please design Specified-sentences-specific questions with Wh-Questions format and subject-verb-object structure for each listed important entity or noun phrase chunk in step 2.
Please ensure that:
(1) Don't generate follow-up question to other questions.
(2) Restrict the answer to a noun-phrase chunk or entity in sentence after coreference parsing, if entity and noun blocks have modifiers, include the modifiers as part of the answer.
(3) Don't generate open-end questions.
(4) Generate only one question that is the most relevant to factual error detection for each important entity or noun phrase chunk.
(5) The answer cannot be a restatement of the question.

4. Follow the example below and format your response as [Q1: <question>, A1: <answer>] for each question and entity pair as the final response.

---Example---

"Context": Gemma the pit bull was filmed at home in California being fed some treats. But in a bid to trick her, Mike throws a broccoli spear into the mix. Immediately the canine pulls a look of disgust as she chomps on the vegetable. She then proceeds to spit it out on her towel, Mike and Amy are laughing.

"Specified sentence" : she then proceeds to spit it out on her towel.

"Important named entities (or noun phrase chunks) in sentence after coreference parsing" : she (referring to gemma), it (referring to the vegetable), her towel (gemma's towel), Mike and Amy

"Response":
[Q1: Who then proceeds to spit broccoli out on the floor?, A1: she (gemma)]
[Q2: What is spit out on the floor by gemma ?, A2: it (the vegetable)]
[Q3: Where is gemma spit the vegetable?, A3: her towel (gemma's towel)]
[Q4: Who are laughing?, A4: Mike and Amy]

---Your Task---

"Context": {context}

"Specified sentences" : {sentence}

"Important named entities(or noun phrase chunk) in Sentence after coreference parsing" : < named entities or noun phrase chunks >

"Response": < Questions and answers >

Please refer to the instruction and example, then finish Your_Task. Please perform coreference parsing on the specified sentences, list the important entities and noun phrase chunks, and finally generate qualifying question-answer pairs. Make sure to follow the format as in the example.

Figure 6: Prompt for QG.

Questions Answering

Now you are one of the greatest English linguist experts and Logicians. I will show you part of the source text, question-answer pairs, respectively. For each question-answer pair, please first find out what is relevant to the question from the given source text, and then please judge whether the given answer matches the content of the source text. Instead of merely relying on a simple comparison between the answer and the source text, please THINK STEP BY STEP to fully engage your textual comprehension and reasoning abilities. The guideline and example are as below:

--- Reply Guideline---
Consistent: The question-answer pair aligns with or does not contradict the information provided in the text. Even if the answer is somewhat vague or generalized, it broadly corresponds to the theme or subject matter in the source text without directly conflicting with the information.
Inconsistent: The question-answer pair conflicts directly with or is contradicted by explicit information or assertions provided in the original text, or there is no sufficient basis for answering the question in the source text.

--- Example ---

Source Text:
GAO also examined and analyzed key acquisition documents including contractor monthly status reports, earned value management data, and Defense Contract Management Agency reports to determine the performance and cost status of the development effort. In December 2015, we reviewed the program's Integrated Master Schedule (IMS) and compared it against best practices criteria in the GAO Schedule Assessment Guide and discussed the results of our schedule assessment with VH-92A program officials.

Question-answer pairs

Q1: Who did GAO discuss the results of the schedule assessment with in December 2015? A1: Contract Management Agency
Q2: Who did GAO discuss the results of the schedule assessment with in December 2015? A2: program officials
……

Reasoning process and judgement:
Q1: Who did GAO discuss the results of the schedule assessment with? A1: Contract Management Agency
Reasoning process: The document mentions that the GAO discussed the results of the schedule assessment with VH-92A program officials. Answer "Contract Management Agency" is clearly contradicting the information in the text.
Judgement: Inconsistent
Q2: Who did GAO discuss the results of the schedule assessment with? A2: program officials
Reasoning process: The document mentions that the GAO discussed the results of the schedule assessment with VH-92A program officials. Although the answer "program officials" is a fairly generic response instead of a more specific answer "VH-92A program officials", it contains no error messages and does not contradict the information provided in the text.
Judgement: Consistent
…….

--- Your_Task ---
Source Text:
{source text}

Question-answer pairs:
{Question-answer pairs}

Please refer to the instruction and example, finish Your_Task by showing your reasoning process and final judgement in format (Reasoning progress: < Your reasoning progress >, Judgement: < Consistent or Inconsistent >) for each question-answer pair.

Figure 7: Prompt for QA.