Detecting Errors through Ensembling Prompts (DEEP): An End-to-End LLM Framework for Detecting Factual Errors

Anonymous ACL submission

Abstract

Accurate text summarization is one of the most common and important tasks performed by Large Language Models, where the costs of human review for an entire document may be high, but the costs of errors in summarization may be even greater. We propose *Detecting Errors through Ensembling Prompts (DEEP)* - an end-to-end large language model framework for detecting factual errors in text summarization. Our framework uses a diverse set of LLM prompts to identify factual inconsistencies, treating their outputs as binary features, which are then fed into ensembling models. We then calibrate the ensembled models to produce empirically accurate probabilities that a text is factually consistent or free of hallucination. We demonstrate that prior models for detecting factual errors in summaries perform significantly worse without optimizing the thresholds on subsets of the evaluated dataset. Our framework achieves state-of-the-art (SOTA) balanced accuracy on the AggreFact-XSUM FTSOTA, TofuEval Summary-Level, and HaluEval Summarization benchmarks in detecting factual errors within transformer-generated text summaries. It does so without any fine-tuning of the language model or reliance on thresholding techniques not available in practical settings.[1]

1 Introduction

The advancement of cutting-edge Large Language Models (LLMs) like GPT-4, Claude 3, LLaMA-2, and Gemini variants introduces a significant challenge: despite producing content that is linguistically coherent, their outputs frequently contain misleading or false information, often referred to as hallucinations or factual inconsistencies. Hallucinations in Large Language Models refer to instances where the model generates usually plausible but entirely fabricated information. Factual inconsistencies, a specific type of hallucination, occur when generated text contradicts the source material or other well-established facts not explicitly mentioned in the source.

Traditional automatic evaluation methodologies like ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and BLEU (Papineni et al., 2002) have been instrumental in assessing Natural Language Generation tasks. However, numerous studies demonstrate the lack of correlation between initial automatic evaluation models and human judgment in tasks such as machine translation (Callison-Burch et al., 2006; Bhattacharyya et al., 2007), image captioning (Cui et al., 2018), and notably, factuality (Fu et al., 2023; Mao et al., 2023). In particular, these models struggle to capture semantic equivalence when there are significant discrepancies in length, syntax, and wording between two texts (Guo and Vosoughi, 2023; Stent et al., 2005). Consequently, specialized models (Laban et al., 2021; Kryściński et al., 2019; Goyal and Durrett, 2021) have been developed to assess textual factual consistency, verifying the truthfulness of a claim or summary based on given ground truth textual content.

However, existing models, often fine-tuned variants of RoBERTa (Liu et al., 2019) for assessing factual consistency, exhibit significant limitations. As highlighted in Tang et al. (2023), these models show reduced effectiveness in detecting factual inconsistencies in content produced by recent state-of-the-art text-generating models. Ensemble learning is the practice of merging the outputs of multiple models to produce a more accurate prediction (Dietterich, 2000). Forbes et al. (2023) demonstrated that ensembling factual consistency models by calculating their weighted mean surpassed the performance of individual models in detecting hallucinations within a small dataset of GPT-3-generated Wikipedia abstractive summaries.

In light of these limitations, this study evaluates benchmarks exclusively featuring summaries from

[1] Code and data are available on GitHub.

recent transformer-based language models. This approach more accurately reflects actual usage scenarios, where users commonly need to validate texts generated by newer LLMs rather than texts from older text generation models. We assess factual consistency using the AggreFact-XSUM FTSOTA, TofuEval Summary-Level, and HaluEval Summarization (Tang et al., 2023, 2024; Li et al., 2023) benchmarks, consisting of transformer-generated abstractive summaries featuring hallucinations that existing models struggle to identify.

Existing factual consistency encoder models output numerical scores, requiring thresholding to map the scores to binary labels. However, Tang et al. (2023) demonstrates that the optimal threshold for factual consistency models varies depending on the recentness of the summarization model within the AggreFact dataset. Previous studies that report the performance of these factual consistency models have fine-tuned each model's threshold using the development subset of the same dataset under evaluation (Laban et al., 2021; Fabbri et al., 2022; Tang et al., 2023, 2024).

This approach is problematic and unrealistic, as it assumes access to labeled data from the target dataset, which may not be available in real-world scenarios. In this study, we benchmark five popular state-of-the-art factual consistency models and demonstrate a substantial decline in performance when thresholds are learned from different datasets or set to their default midpoint.

We extend the findings of Tang et al. (2023) and discover that the optimal threshold for each factual consistency model varies widely across different datasets, even when evaluating text generated solely from recent summarization models. Figure 1 reveals that, even when considering only datasets with summaries from recent transformer models, the optimal linear threshold for maximizing balanced accuracy differs widely for each factual consistency model, covering a broad range of their possible output scores.[2] Moreover, Figure 2 demonstrates that optimizing thresholds on non-test data or setting them to the midpoint of each model's score range significantly reduces balanced accuracy compared to test-set optimization. The reliance of these models on dataset-specific thresholds, as demonstrated by our findings, limits their practical utility in evaluating factual consistency across a diverse range of text without further fine-tuning or adjustments.

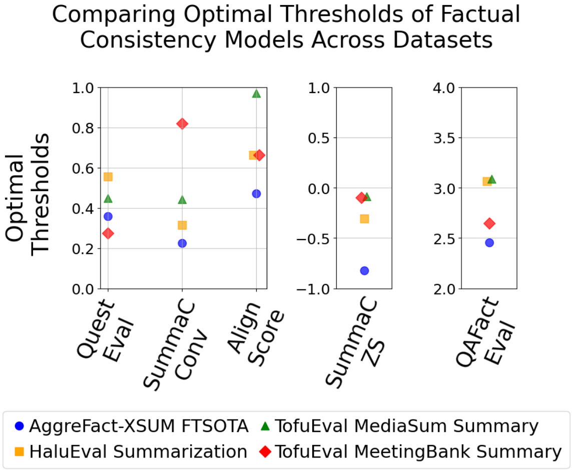

Figure 1: Optimal thresholds of factual consistency models when set to maximize balanced accuracy on each test dataset.

Previous efforts to use LLMs for identifying factual inconsistencies include Wang et al. (2023), who prompted ChatGPT to return numerical factuality scores for summarization, and Luo et al. (2023), who used ChatGPT to produce binary factuality judgments. However, Tang et al. (2023) showed that these existing prompts had shown poor performance on detecting factual consistencies over the AggreFact FTSOTA benchmark of transformer-generated summaries.

Existing LLM solutions are frequently overconfident in their assessments of a text's factual consistency. Tang et al. (2024) showed that their summary-level binary factual consistency prompts, when used with GPT-4, frequently failed to identify sentences with factual errors, resulting in False Positive Rates of 69% and 46% on the MediaSum and MeetingBank subsets of the TofuEval Summary-Level dataset, respectively. This issue of overconfidence is not unique to their approach but rather a general problem that neural networks, including LLMs, tend to be overconfident in their predictions (Guo et al., 2017; Minderer et al., 2021; Jiang et al., 2021; Xiong et al., 2023).[3]

Confidence elicitation, an increasingly popular method, involves prompting the LLM to output its uncertainty along with its prediction (Lin et al., 2022; Xiong et al., 2023; Tian et al., 2023). Despite

[2] QuestEval, SummaC-Conv, and AlignScore (with mode nli_sp) generate scores from 0 to 1. SummaC-ZS scores range from -1 to 1. QAFactEval scores range from 0 to 5.

[3] Binary prompts, by their very nature, force models to reduce nuance and uncertainty into a single decision, contributing to the overconfidence observed in their responses.

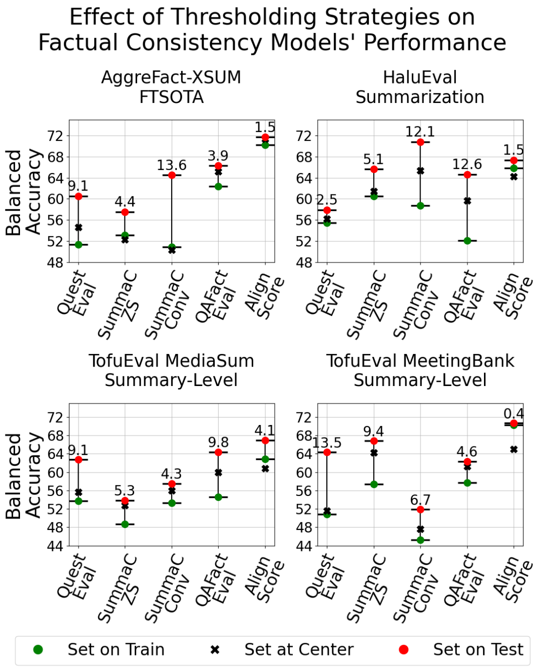

Figure 2: Factual consistency models' performance varies significantly based on threshold optimization strategy. The bars show balanced accuracy for factual consistency models under three threshold optimization strategies: optimizing the thresholds on the test dataset ("Optimizing on Test"), setting thresholds to the midpoint of each model's score range ("Optimizing at Center"), or optimizing on all datasets except the test set ("Optimizing on Train"), which reflects a realistic scenario of applying the model to unseen data. Numbers above bars quantify the decrease in balanced accuracy when thresholds are optimized on non-test data compared to the test dataset itself, underscoring the difficulty of effectively applying these models to unseen data in practice.

its potential, no confidence elicitation approach has yet consistently yielded accurate confidence estimates across diverse LLMs and tasks, limiting its current practical utility. Calibration, the process of adjusting model-predicted probabilities to match their empirical accuracies, has remained the leading solution to correct model overconfidence (Guo et al., 2017).

We propose a novel approach - crafting a diverse set of LLM prompts that each output a binary score, indicating each prompt's belief of whether the summaries contain any factual errors. These binary features are then fed into ensembling models, which integrate the multiple perspectives of each prompt to produce a single probability. Finally, we calibrate the ensemble models to obtain empirically accurate probabilities that a given summary is factually consistent or free of hallucination.[4] The full pipeline of our framework can be seen in Figure 3.

The primary contributions of our work can be outlined as follows: **1:** We demonstrate that prior methods for detecting factual errors in summarizations perform significantly worse without the common practice of an optimized threshold on subsets of the dataset under test. **2:** We introduce a large and diverse set of prompts, each employing unique methods and evaluation protocols to detect hallucinations and factual inconsistencies in generated summaries. **3:** Our end-to-end framework achieves state-of-the-art balanced accuracy on the AggreFact-XSUM FTSOTA, TofuEval Summary-Level, and HaluEval Summarization benchmarks in detecting factual errors in transformer-generated text summaries, all without fine-tuning the language model or relying on impractical thresholding techniques.

2 Datasets

The AggreFact FTSOTA benchmark (Tang et al., 2023) tests a model's ability to identify factual inconsistencies in summaries produced by fine-tuned transformer-based summarization models. The dataset combines nine existing annotated factuality datasets, converting all factual consistency scores to binary. AggreFact is categorized based on the development timeline of the underlying summarization models into FTSOTA, EXFORMER, and OLD categories, and is divided into AggreFact CNN/DM and AggreFact-XSUM subsets. We benchmark AggreFact-XSUM FTSOTA and exclude its CNN/DM counterpart due to its insufficient number of factual inconsistencies and its more extractive summary style, compared to the more abstractive style of current SOTA LLMs.

HaluEval (Li et al., 2023), a comprehensive benchmark for assessing hallucination in LLMs, uses a ChatGPT-based two-step sampling-then-filtering framework to create the dataset. We focus on the summarization subset of HaluEval, which pairs each of the 10,000 sampled document contexts with two summaries: one with hallucination and one without.

[4] We craft prompts that aim to detect either factual inconsistencies or hallucinations, allowing the ensembling model to determine the importance of each prompt for the datasets it is trained on. This approach enables us to sidestep the subtle distinction between factual inconsistencies and hallucinations, and instead let the ensembling model decide what prompts are most relevant for the training data.

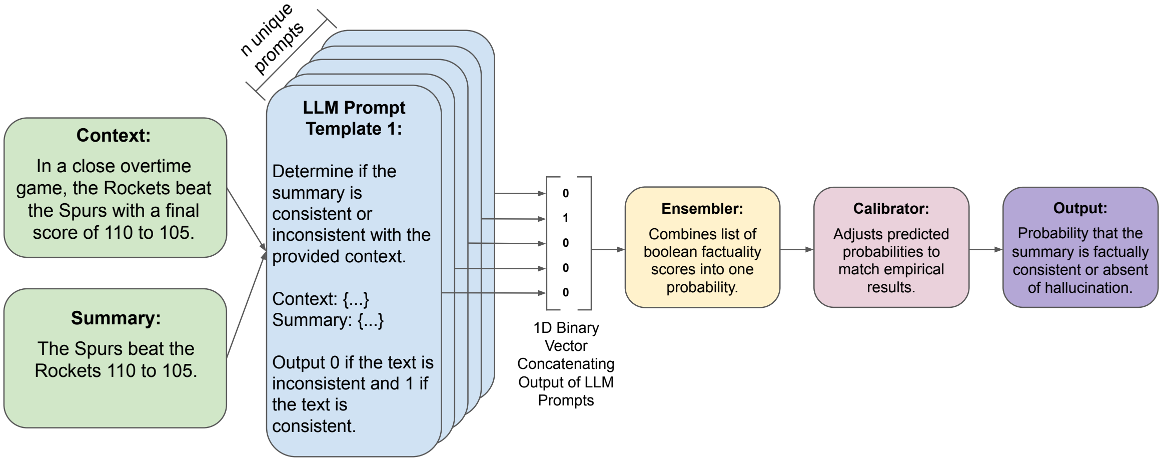
n unique prompts
Context:
In a close overtime game, the Rockets beat the Spurs with a final score of 110 to 105.
Summary:
The Spurs beat the Rockets 110 to 105.
LLM Prompt Template 1:
Determine if the summary is consistent or inconsistent with the provided context.
Context: {...}
Summary: {...}
Output 0 if the text is inconsistent and 1 if the text is consistent.
0
1
0
0
0
1D Binary Vector Concatenating Output of LLM Prompts
Ensembler:
Combines list of boolean factuality scores into one probability.
Calibrator:
Adjusts predicted probabilities to match empirical results.
Output:
Probability that the summary is factually consistent or absent of hallucination.

Figure 3: Diagram of our end-to-end framework.

TofuEval (Tang et al., 2024) is a topic-focused dialogue summarization benchmark containing 1.5K LLM-generated summaries from the MediaSum (Zhu et al., 2021) and MeetingBank (Hu et al., 2023) datasets. TofuEval uses a two-stage annotation process, where two expert linguists independently assess the binary relevance, completeness, and factual consistency of each sentence in the summaries. We focus solely on the factual consistency annotations of main topic summaries, merging summarization sentences into one paragraph deemed consistent if all summarization sentences are individually marked as consistent.[5] We thus refer to this concatenation interchangeably as "TofuEval Summary-Level" or simply "TofuEval Summary".

We use the provided test subsets of each dataset, except for HaluEval Summarization, which lacks a train-test split. For HaluEval Summarization, we use a balanced random sample of 3,000 summaries as our test set.

3 Methodology

3.1 Ensembling Methods

We train and evaluate 16 ensembling methods. When applicable, parameters for all relevant methods are determined using a grid search over the parameter feature space. Below is a listing of these methods:

- **Linear Models**: LogisticRegression, LDA
- **Tree-Based Methods**: RandomForest, GradientBoosting, AdaBoost, DecisionTree, CatBoost, XGB, LGBM
- **Ensemble Voting:** MajorityLabelVoter, WeightedMajorityLabelVoter
- **Label Aggregation Models**: LabelModel, Dawid-Skene[6]
- **Other Methods**: Support Vector Machines, Nearest Neighbors, Naive Bayes (BernboulliNB)

3.2 Metrics Used

Following Tang et al. (2023) and Laban et al. (2021), we use balanced accuracy to assess each model's proficiency in detecting factual errors. Balanced accuracy helps provide a less biased evaluation by balancing the importance of sensitivity and specificity, ensuring that the predominance of the majority class does not skew the results.

We measure the reliability of predicted probabilities for factual consistency using Expected Calibration Error (ECE). Expected Calibration Error (ECE) is calculated by partitioning predictions into n bins based on their confidence levels, computing

[5] Tang et al. (2024) separated summaries into Main and Marginal, with the majority of sentences categorized as Main. We focus on the Main summaries because the Marginal dataset in TofuEval is insufficiently small for reliable analysis.

[6] The LabelModel, as delineated in Ratner et al. (2017), is particularly effective in learning the conditional probabilities of labeling functions, adeptly reweighting their outputs in semi-supervised contexts. The LabelModel represents an evolution in semi-supervised learning, encompassing techniques such as those in FlyingSquid (Fu et al., 2020), Dawid-Skene (Dawid and Skene, 2018), Data Programming (Ratner et al., 2016), and MeTaL (Ratner et al., 2019), all of which were originally included as part of the Wrench benchmark (Zhang et al., 2021).

the absolute difference between the actual accuracy and the predicted probability in each bin, and then taking the weighted average of these differences across all bins. ECE effectively measures the discrepancy between a model's confidence in its predictions and its actual performance, with lower values indicating better calibration.

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right| \quad (1)$$

Where: M is the number of bins, N is the total number of samples, B_m is the set of samples in bin m, $|B_m|$ is the number of samples in bin m, $\mathrm{acc}(B_m)$ is the accuracy of predictions in bin m, and $\mathrm{conf}(B_m)$ is the average predicted probability in bin m.

3.3 Calibrating Ensembled Models for Reliable Probability Estimates

Histogram Binning (Zadrozny and Elkan, 2002), Bayesian Binning into Quantiles (BBQ) (Naeini et al., 2015), and Isotonic Regression (Zadrozny and Elkan, 2002) are non-parametric methods methods of calibration, while Temperature Scaling (Guo et al., 2017) and Platt Scaling (Platt, 1999) are parametric. Platt Scaling applies a sigmoid function to model outputs to calibrate them. We test and apply Platt Scaling, BBQ, Histogram Binning, and Isotonic Regression, to obtain reliable probability estimates from our ensembled models.

3.4 Methodology for Prompt Creation

Our LLM prompts, created using GPT-4 in the OpenAI playground, employs various Chain of Thought (CoT) (Wei et al., 2023) approaches to guide models through a structured evaluation of factual consistency. Most prompts have various explicit evaluation criteria, requiring the LLM to determine if each claim in the summary can be inferred directly from the context.

3.5 Selection of Prompt Pool

Prompt selection for each prompt pool size was determined using Recursive Feature Elimination (RFE) and Minimum Redundancy Maximum Relevance (MRMR) methods (Guyon et al., 2002; Ding and Peng, 2005). We select the best-performing subset of prompts of these two methods for each prompt size.

4 Results

4.1 Individual Prompt Results

Table 1 displays the top five prompts' performance, highlighting an average 2.5% increase in balanced accuracy when using GPT-4-Turbo over GPT-3.5-Turbo.

4.2 Ensemble Benchmarking and Comparing to Existing Methods

For each test dataset, we train ensemble models on the binary LLM prompt outputs using only the three remaining non-test datasets. Table 2 shows the impact of different LLM prompt sizes and ensembling methods on balanced accuracy for the AggreFact-XSUM FTSOTA, HaluEval Summarization, and TofuEval Summary-Level test datasets. The LabelModel (Ratner et al., 2017) is the most frequent top performer among the ensemble models across dataset and prompt size combinations. Snorkel's LabelModel emerges as the top performer among the ensemble models, likely because it is designed to effectively combine weak and noisy classifiers, which accurately describe the individual performance of the LLM prompts.

Table 2 demonstrates that ensembling binary outputs from multiple prompts significantly improves performance compared to the best individual LLM prompts, shown in Table 1. Increasing the number of ensembled prompts from 5 to 9 does not consistently improve performance, likely because the top-5 prompts use GPT-4 while the remaining prompts use GPT-3.5. The limited training data may hinder the ensembling models' ability to benefit from the less reliable GPT-3.5 prompts.

Table 3 compares the balanced accuracy of state-of-the-art encoder-based factual consistency models and LLM solutions to our ensemble methods in identifying factual inconsistencies and hallucinations. As shown in Table 3, our proposed ensemble approach outperforms all existing methods across the benchmark datasets.[7]

4.3 Calibration Analysis

We investigate the effectiveness of applying calibration to obtain reliable probability estimates from our ensembled models. Platt Scaling (Platt,

[7] The QAFactEval and QuestEval results for HaluEval Summarization were obtained using a balanced random sample of 2,000 context-summary pairs. LLM Solution method results on HaluEval were obtained using a slightly larger balanced random sample of 3,000 context-summary pairs. All other results use the full datasets.

Prompt	**LLM**	**AggreFact-XSUM FTSOTA**	**HaluEval Summarization**	**TofuEval MediaSum Summary-Level**	**TofuEval MeetingBank Summary-Level**
Prompt 1	GPT-3.5	67.4	69.7	57.6	68.0
	GPT-4	**69.1**	71.5	60.2	**73.2**
Prompt 2	GPT-3.5	66.4	67.2	58.4	65.0
	GPT-4	68.4	68.7	61.4	70.3
Prompt 3	GPT-3.5	65.0	70.2	63.2	60.0
	GPT-4	66.8	**72.7**	63.9	64.3
Prompt 4	GPT-3.5	61.7	63.7	65.4	68.7
	GPT-4	62.8	66.4	**66.6**	72.0
Prompt 5	GPT-3.5	60.2	66.4	61.0	63.3
	GPT-4	61.8	68.2	61.8	67.6

Table 1: Balanced accuracy of the top-5 performing prompts across all test datasets. GPT-4 refers to GPT-4-Turbo, while GPT-3.5 refers to GPT-3.5-Turbo for all datasets except HaluEval Summarization, where GPT-3.5-Turbo-16K is used due to the longer context length. The best performance for each dataset is shown in bold.

Ensembling Method	**AggreFact-XSUM FTSOTA**			**HaluEval Summarization**			**TofuEval MediaSum Summary-Level**			**TofuEval MeetingBank Summary-Level**		
	Number of Prompts			**Number of Prompts**			**Number of Prompts**			**Number of Prompts**		
	3	**5**	**9**	**3**	**5**	**9**	**3**	**5**	**9**	**3**	**5**	**9**
Baseline	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
AdaBoost	67.04	67.04	70.21	71.47	73.73	72.67	**65.80**	63.80	64.74	71.17	72.89	71.97
BernoulliNB	67.04	71.02	68.07	73.27	72.33	68.57	64.24	65.96	64.80	74.07	76.38	77.16
CatBoost	67.04	67.04	68.73	71.47	71.80	71.53	**65.80**	63.80	64.74	71.17	72.89	73.80
DawidSkene	69.17	69.62	70.28	73.27	70.53	67.37	64.17	64.33	63.47	74.07	73.70	71.91
DecisionTree	67.04	67.04	66.11	73.27	68.67	71.50	60.16	64.90	63.31	73.16	73.91	67.00
GradientBoosting	67.04	66.53	68.91	73.90	71.07	67.63	**65.80**	64.67	64.30	71.17	73.91	73.80
KNeighbors	67.04	67.04	67.39	**74.87**	71.43	68.77	64.24	64.17	63.28	74.07	69.55	71.08
LDA	67.04	67.04	69.79	73.27	73.83	70.60	**65.80**	63.80	64.44	71.17	72.89	71.03
LabelModel	**69.40**	**71.92**	67.93	73.90	71.67	68.50	64.24	**66.33**	**65.53**	**74.10**	**79.38**	**79.74**
LGBM	67.04	67.04	68.73	73.90	**74.07**	**73.43**	**65.80**	64.67	65.17	74.07	73.91	73.58
LogisticRegression	67.04	67.04	70.17	73.27	72.53	70.40	**65.80**	63.80	63.77	71.17	72.89	70.74
MajorityLabelVoter	69.17	69.62	**71.05**	73.27	70.37	67.87	64.17	64.33	63.50	**74.10**	73.70	70.67
RandomForest	67.04	66.72	67.91	73.90	70.87	70.77	**65.80**	64.90	63.54	71.17	73.99	73.58
SVC	67.04	66.72	67.61	73.90	72.87	72.03	**65.80**	64.67	64.30	71.17	73.91	72.27
WeightedMajorityVoting	67.04	67.04	70.17	73.27	73.73	72.60	**65.80**	63.80	63.77	71.17	72.89	70.74
XGB	67.04	66.72	70.39	73.90	71.73	66.73	**65.80**	64.90	63.87	71.17	73.99	72.14

Table 2: Exploring the impact of various LLM prompt sizes and ensembling methods on balanced accuracy across all test datasets. The top performing ensemble method for each prompt size-dataset combination is shown in bold.

1999) outperforms other methods, achieving low Expected Calibration Error (ECE) scores generally below 7%. Applying Platt Scaling to ensemble models in both HaluEval Summarization and TofuEval significantly reduces Expected Calibration Error (ECE) across various ensembling techniques.

Reliability diagrams are tools for assessing the accuracy of probability estimates from both uncalibrated and calibrated ensemble models. The reliability diagram in Figure 4 shows that Platt Scaling, when applied to an ensemble model, can significantly reduce overconfidence in predicting a text's factual consistency.

4.4 Statistical Testing

We assess if the performance gains in LLM ensembling over existing methods in models is statistically significant. For the AggreFact-XSUM FTSOTA, TofuEval, and HaluEval Summarization dataset, we execute distinct evaluations using bootstrap resampling techniques (Efron, 1982), comparing the best Ensembling Method against the previous top performing model for each dataset. Following Laban et al. (2021), we conduct comparisons at a statistical significance level of $p = 0.01$, incorporating the Bonferroni adjustment (Bonferroni, 1935) due to the multiple tests conducted on the datasets. Our LLM ensembling methods demonstrate statistically meaningful advancements

Method Type	Method	AggreFact-XSUM FTSOTA	HaluEval Summarization	TofuEval MediaSum Summary-Level	TofuEval MeetingBank Summary-Level
Encoder Models	AlignScore	70.2 ± 3.8	65.8 ± 0.7	62.3 ± 5.8	70.1 ± 5.4
	QuestEval	51.3 ± 4.2	55.4 ± 2.2	53.7 ± 6.0	50.8 ± 5.9
	SummaC-ZS	53.0 ± 4.2	60.5 ± 0.7	48.6 ± 6.0	57.4 ± 5.9
	SummaC-Cv	50.9 ± 4.2	58.7 ± 0.7	53.3 ± 5.9	45.2 ± 6.1
	QAFactEval	62.4 ± 4.0	52.0 ± 2.2	54.6 ± 5.9	57.8 ± 5.7
LLM Solution	ChatGPT-ZS (GPT-3.5)	62.1 ± 4.0	64.3 ± 1.8	63.6 ± 5.8	69.0 ± 5.3
	ChatGPT-COT (GPT-3.5)	55.4 ± 4.1	62.5 ± 1.8	63.1 ± 5.8	66.8 ± 5.2
	ChatGPT-DA (GPT-3.5)	56.4 ± 4.1	59.6 ± 1.8	52.7 ± 5.9	52.3 ± 5.8
	ChatGPT-Star (GPT-3.5)	55.4 ± 4.1	61.5 ± 1.8	57.4 ± 5.8	55.7 ± 5.7
	Tang2024-Summary (GPT-3.5)	62.4 ± 4.0	64.0 ± 0.9	61.9 ± 3.4	71.9 ± 3.1
	Tang2024-Summary (GPT-4)	62.9 ± 4.0	66.1 ± 0.9	62.3 ± 3.4	72.9 ± 3.1
	Ensemble-Top-3 (GPT-4)	69.4 ± 3.8	**74.9** ± 1.6	65.8 ± 5.7	74.1 ± 5.1
	Ensembled-Top-5 (GPT-4)	**71.9**± 3.8	74.1 ± 1.6	**66.3** ± 5.7	79.4 ± 5.1
	Ensemble-Top-9 (Mixed)	71.1 ± 3.8	73.4 ± 1.6	65.5 ± 5.7	**79.7** ± 4.9

Table 3: A chart comparing the balanced accuracy of encoder-based models and LLM solutions in identifying factual inconsistencies and hallucinations. For each test dataset, the encoder-based model scores are obtained using linear thresholds optimized on the other three datasets, ensuring that neither the test data nor its validation set is used for threshold tuning. Lower performance in encoder models compared to existing studies is due to evaluating without fine-tuning each model's threshold using the test dataset's development subset. 95% confidence intervals are shown, with the highest performing method for each dataset in bold. Mixed refers to ensembling the binary LLM outputs from both GPT-3.5 and GPT-4.

Model	AggreFact-XSUM FTSOTA					
	3 Prompts		5 Prompts		9 Prompts	
	Uncal. (%)	Platt (%)	Uncal. (%)	Platt (%)	Uncal. (%)	Platt (%)
AdaBoost	5.1	2.1	12.0	5.7	18.7	4.3
BernoulliNB	9.2	5.1	15.9	5.0	21.5	7.2
CatBoost	8.2	2.5	8.1	5.2	7.7	6.7
DecisionTree	8.9	6.4	7.2	5.3	9.3	4.9
GradientBoosting	7.5	4.6	6.1	5.2	8.6	5.4
KNeighbors	11.0	4.9	10.4	4.7	9.8	4.5
LabelModel	14.7	4.1	23.8	4.7	22.8	5.3
LDA	6.9	4.5	7.4	6.1	7.1	4.1
LGBM	15.5	4.9	20.0	7.7	15.9	6.1
LogisticRegression	13.4	4.9	6.5	5.8	6.6	3.9
MultinomialNB	8.3	8.1	4.2	**2.6**	4.4	**2.6**
RandomForest	9.8	6.3	6.0	4.1	9.8	4.5
SVC	9.3	**0.9**	8.4	6.5	9.5	5.7
XGB	7.2	6.4	6.9	4.8	16.0	6.9

Table 4: Comparison of the Expected Calibration Error (ECE) for ensembling models before (uncalibrated) and after calibration using Platt Scaling, across various models and prompt pools. The calibration model was trained only on the three non-test datasets.

on the HaluEval dataset, with statistically significant p-values of under .01 in both the original and Bonferroni-adjusted analyses. However, the statistical tests do not confirm a statistically significant advantage, with p-values below .01, compared to the second-best methods across the remaining three datasets. The limited number of samples in the AggreFact-XSUM FTSOTA and TofuEval datasets, in contrast to the larger HaluEval dataset, necessitates a significantly greater improvement in performance to achieve statistical significance. [8]

[8] To eliminate any biases caused by varying sample sizes across models, we exclusively conduct statistical testing on the overlapping subsets of all samples under consideration, thereby ensuring that differing sample sizes do not affect our statistical significance testing.

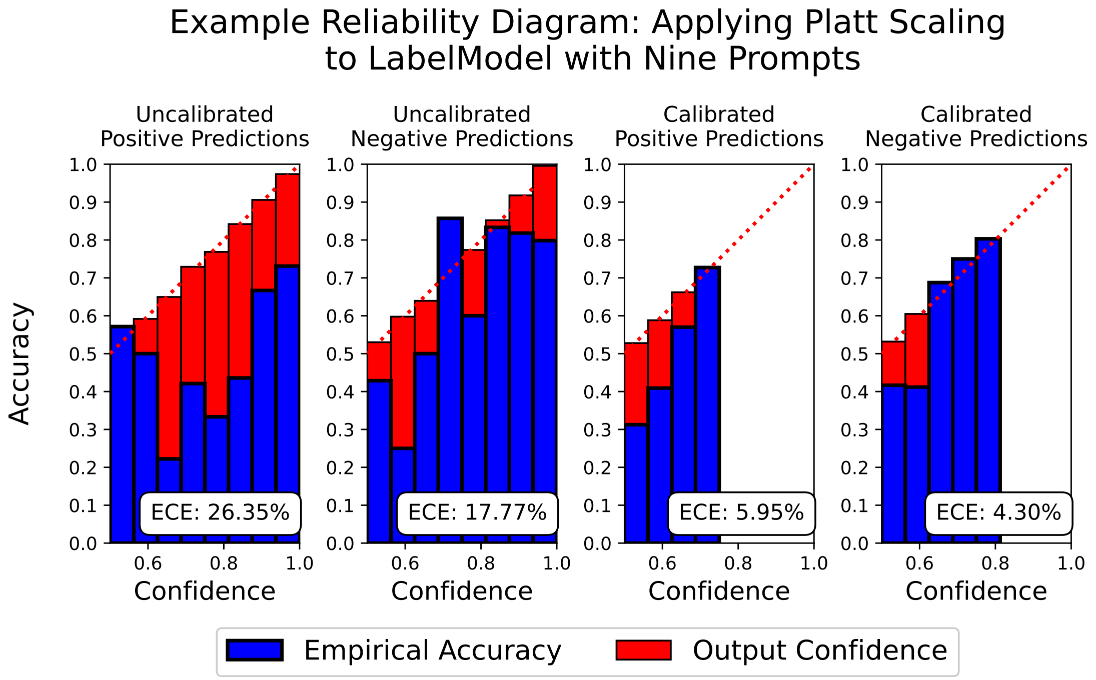

Figure 4: An example reliability diagram highlighting the difference in reliability between the predicted probabilities before and after calibration. The reliability diagram is separated by positive and negative predictions, highlighting the contrast in model confidences between predicting whether a summary is factually consistent with the source context versus inconsistent. In this visualization, LabelModel was trained with the output of nine LLM Prompts on the non-test datasets and tested on the AggreFact-XSUM FTSOTA test dataset.

5 Conclusion

We introduce *Detecting Errors through Ensembling Prompts (DEEP)*, a state-of-the-art LLM-based method for detecting factual consistencies and hallucinations in summaries. Our findings reveal that factual consistency encoder models exhibit a pronounced sensitivity to threshold settings, with their performance markedly declining if the threshold is not adjusted based on the test datasets. We demonstrate that *DEEP* surpasses the performance of existing methods and models in evaluating the factual consistency of summaries produced by recent transformer models. Finally, we show that by calibrating ensemble models using binary input features derived from LLM prompts, *DEEP* achieves reliable probabilities indicating a text's factual consistency or presence of errors.

Limitations

LLM approaches for identifying factual errors are significantly more resource-intensive than existing fine-tuned encoder models, requiring up to three orders of magnitude more parameters. Our method requires more computation; however, this cost may be justified in high-stakes situations where failing to identify factual errors is costly.

Future research should explore the performance of these prompts on more powerful language models as they become available. Additionally, future work should create a dataset of quality examples of chain-of-thought reasoning to identify factual errors, enabling few-shot learning to boost LLM performance. Future work should consider comparing the performance of ensembling factual consistency model scores to LLM prompt ensembling.

Future work should investigate why encoder models for factual consistency evaluation require dataset-specific linear thresholding for optimal performance. Possible reasons include: (1) the dependence of optimal binary classification thresholds on the average number of sentences per summary, as most models calculate summary factuality scores by averaging sentence-level scores; (2) dataset imbalances in the ratio of summaries with factual errors, with thresholding balancing specificity and sensitivity; and (3) inherent differences in identifying factual errors across datasets, even when controlling for summary length and annotation imbalances, due to factors like summarization models or the context's source. However, these reasons are speculative, and further experiments are needed to test these hypotheses and explore approaches that reduce model sensitivity to dataset characteristics.

It is uncertain how many labeling errors exist in the AggreFact-XSUM-SOTA and HaluEval Summarization datasets. Future datasets should create summarization error datasets, where, like TofuEval, ground truth labels are generated by combining multiple humans annotations. Additionally, the datasets under test contained solely English Summaries. To evaluate LLMs' ability to identify errors across languages, there is a need for multi-lingual summarization error datasets.

Fine-tuning LLMs for error detection and applying our end-to-end pipeline for spotting wider ranges of language generation errors, including those in QA and machine translation, should be explored.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Pushpak Bhattacharyya, Sasikumar Mukundan, and Ritesh Shah. 2007. Some issues in automatic evaluation of english-hindi mt: more blues for bleu.

Carlo E. Bonferroni. 1935. Il calcolo delle assicurazioni su gruppi di teste. *Studi in onore del professore Salvatore Ortu Carboni*, pages 13–60.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy. Association for Computational Linguistics.

Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge Belongie. 2018. Learning to evaluate image captioning.

A. P. Dawid and A. M. Skene. 2018. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 28(1):20–28.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Chris Ding and Hanchuan Peng. 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205.

Bradley Efron. 1982. *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics.

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.

Grant C Forbes, Parth Katlana, and Zeydy Ortiz. 2023. Metric ensembles for hallucination detection. *arXiv preprint arXiv:2310.10495*.

Daniel Y. Fu, Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. 2020. Fast and three-rious: Speeding up weak supervision with triplet methods. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*.

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

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*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks.

Xiaobo Guo and Soroush Vosoughi. 2023. Length does matter: Summary length can bias summarization metrics. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15869–15879, Singapore. Association for Computational Linguistics.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine learning*, 46:389–422.

Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. Meetingbank: A benchmark dataset for meeting summarization. *Preprint*, arXiv:2305.17529.

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.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2021. Summac: Re-visiting nli-based models for inconsistency detection in summarization.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models.

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.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words.

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.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for text summarization.

Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2023. Gpteval: A survey on assessments of chatgpt and gpt-4.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2901–2907.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282.

Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4763–4771.

Alexander J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and Christopher R'e. 2016. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *Computational Linguistics and Intelligent Text Processing*, pages 341–351, Berlin, Heidelberg. Springer Berlin Heidelberg.

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.

Liyan Tang, Igor Shalyminov, Amy Wing mei Wong, Jon Burnsky, Jake W. Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024. Tofueval: Evaluating hallucinations of llms on topic-focused dialogue summarization. *Preprint*, arXiv:2402.13249.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2024)*.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD '02 International Conference on Knowledge Discovery and Data Mining*, pages 694–699.

Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021. Wrench: A comprehensive benchmark for weak supervision.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. Mediasum: A large-scale media interview dataset for dialogue summarization. *Preprint*, arXiv:2103.06410.

A Our Prompts

Prompt 1

Assess the factual consistency of the following claim based solely on the information provided in the summary. Do not make inferences or assumptions beyond the provided information.
Summary: {...}
Claim: {...}
Assessment: Think step by step. Use the following chain of thought.
Step 1: Remember the claim.
Step 2: Remember the summary.
Step 3: Break down all the lines in the claim and remember them. Pay close attention to numbers associated with the events and store them separately.
Step 4: Follow the given chain of thought.
After following these steps, decide if the claim is SUPPORTED or NOT SUPPORTED. If you are not sure, output NOT SUPPORTED. Only output SUPPORTED if you are 100 percent confident that the claim is fully supported by the summary. But regardless, be sure that you output your answer.

Prompt 2

Assess the factual consistency of the following claim based solely on the information provided in the summary. Do not make inferences or assumptions beyond the provided information.
Summary: {...}
Claim: {...}
Assessment: Think step by step. Use the following chain of thought.
Step 1: Remember the claim.
Step 2: Remember the summary.
Step 3: Break down all the lines in the claim and remember them. Pay close attention to numbers associated with the events and store them separately.
Step 4: Use multiple factuality metrics (e.g., ChatGPT-based metrics, FEQA, QAGS) to evaluate the factuality of every line. Compare and contrast the results from each metric, discussing any variations in their effectiveness in detecting factual errors. Explain which metric appears to be most effective and why, based on the specific characteristics of the summary.
Step 5: Summarize results.
Step 6: Be strict. Even if one line is factually inconsistent, mark as UNSUPPORTED; else mark as SUPPORTED.

Prompt 3

In this task, you are required to analyze the factual consistency of a summary against the original article by directly comparing key points.
Procedure:
1. Identify Key Points: - List the key points and claims made in the summary and the corresponding points in the article.
2. Comparative Analysis: - Create a side-by-side comparison for each key point between the summary and the article. - Note any discrepancies, no matter how minor, in each comparison.
3. Detailed Discrepancy Evaluation: - For each noted discrepancy, determine whether it falls under any specific error type (e.g., Negation, Adjective, Coreference, etc.). - Evaluate the impact of each discrepancy on the overall factual consistency. - For each noted discrepancy, determine whether it falls under any specific error type (e.g., Adjective Error, Coreference Error, Number Error, Entity Error, Attribute Error, Pronoun Error, Commonsense Error, Temporal Error, Predicate Error, Discourse Link Error, Relation Error, Quantity Error, Event Error, Noun Phrase Error, Circumstance Error, Hallucination Error). - ENSURE THAT YOU THOROUGHLY AND COMPREHENSIVELY CHECK FOR ANY Adjective Error, Coreference Error, Number Error, Entity Error, Attribute Error, Pronoun Error, Commonsense Error, Temporal Error, Predicate Error, Discourse Link Error,

Relation Error, Quantity Error, Event Error, Noun Phrase Error, Circumstance Error, Hallucination Error - If any of the following discrepancies fall under the listed error types, output NOT SUPPORTED.
4. Strict Criteria for Comparative Support:
- Classify the summary as 'SUPPORTED' only if there are no discrepancies in the comparative analysis. - If any discrepancy is found, classify the summary as 'NOT SUPPORTED'.
Article: {...}
Summary: {...}
Answer (SUPPORTED or NOT SUPPORTED):

B Existing Prompts

Li et al. (2023) Prompt For Identifying Factual Consistencies

Follow the instructions. Instruction: Determine if the text is consistent or inconsistent with the provided knowledge and dialogue history. If there is a logical conflict, respond with 'inconsistent'. If there is no conflict, respond with 'consistent'.
Input: Text: {...}.
Dialogue History: {...}: Please summarize the given knowledge.
Response:

Luo et al. (2023) Zero-Shot

Decide if the following summary is consistent with the corresponding article. Note that consistency means all information in the summary is supported by the article.
Article: [...]
Summary: [...]
Answer (yes or no):

Luo et al. (2023) Chain-of-Thought

Decide if the following summary is consistent with the corresponding article. Note that consistency means all information in the summary is supported by the article.
Article: [...]
Summary: [...]
Explain your reasoning step by step then answer (yes or no) the question:

Wang et al. (2023) ChatGPT-DA

Score the following news summarization given the corresponding news with respect to consistency on a continuous scale from 0 to 100, where a score of zero means "inconsistency" and score of one hundred means "perfect consistency". Note that consistency measures whether the facts in the summary are consistent with the facts in the original article. Consider whether the summary does reproduce all facts accurately and does not make up untrue information.
Article: [...]
Summary: [...]
Scores:

Wang et al. (2023) ChatGPT-Star

Score the following news summarization given the corresponding news with respect to consistency with one to five stars, where one star means "inconsistency" and five stars means "perfect consistency". Note that consistency measures whether the facts in the summary are consistent with the facts in the original article. Consider whether the summary does reproduce all facts accurately and does not make up untrue information.
Article: [...]
Summary: [...]
Stars: