COMPARATIVE STUDY OF LARGE LANGUAGE MODEL EVALUATION FRAMEWORKS WITH A FOCUS ON NLP VS LLM-AS-A-JUDGE METRICS

Afnan Alabdulwahab[1] *Chloe Japic*[1] *Colby Le*[1] *Disha Dubey*[1] *Disha Trivedi*[1]
John Hope[1] *Patrick Stone*[1] *Sanjana Srivastava*[1] *Adam Tashman*[*1] *Aidong Zhang*[*1,2]

[1]School of Data Science, University of Virginia, Charlottesville, VA
[2]School of Engineering, University of Virginia, Charlottesville, VA

ABSTRACT

As Large Language Models (LLMs) become increasingly integrated into various industries, their evaluation remains a critical challenge. In collaboration with Deloitte, this study systematically compares traditional NLP-based evaluation metrics with the emerging LLM-as-a-Judge paradigm to assess their effectiveness across key dimensions such as retrieval accuracy, response accuracy, toxicity detection, bias detection, hallucination, summarization quality, tone identification, and readability. We specifically investigate whether LLM-as-a-Judge frameworks, using models like Anthropic's Claude, which evaluate outputs via API's, can capture LLM-specific behaviors that traditional metrics may overlook. Preliminary findings suggest that while LLM-as-a-Judge frameworks provide nuanced insights, they introduce self-referential bias and consistency issues, whereas traditional NLP methods, though more transparent, may not fully capture LLM-specific behaviors such as chain-of-thought consistency, hallucination, tone adaptation, and subtle bias. Addressing skepticism about LLM-as-a-Judge quality and trustworthiness, as well as noting limitations of traditional evaluation methods, the study hypothesizes that LLM-as-a-Judge methodology offers significant future utility but it is not a silver bullet for LLM evaluation across all proposed metrics. The study aims to develop a structured evaluation playbook outlining the strengths, weaknesses, and best-use cases of each approach, contributing to the standardization of LLM assessment methodologies for responsible AI deployment. This will detail the current landscape of LLM evaluation and identify on which metrics LLM-as-a-Judge is immediately available to serve as an effective evaluation framework.

Index Terms— LLM evaluation, NLP metrics, LLM-as-a-Judge, model assessment, AI benchmarking

*Co-corresponding authors.

1 Introduction

Large Language Models (LLMs) are now widespread across various industries, with their adoption expected to only grow rapidly in the coming years. However, with novel LLMs being introduced at a high pace, the validation of these models needs some introspection. As organizations increasingly integrate LLMs into their workflows, evaluating their performance across key dimensions is essential for responsible deployment. In this research, in collaboration with Deloitte, we aim to systematically assess existing LLM evaluation frameworks, focusing on traditional NLP-based metrics and the emerging LLM-as-a-Judge paradigm. The goal is to develop a structured evaluation playbook that not only provides an evidence-based approach to selecting the most effective LLM evaluation frameworks for current tasks but also offers guiding principles that can adapt as LLM technology and evaluation methods continue to evolve.

As organizations increasingly integrate LLMs into their workflows, evaluating their performance in key dimensions is essential for responsible deployment. This study, in collaboration with Deloitte, systematically compares traditional NLP-based methods with the LLM-as-a-Judge paradigm across 8 metrics: accuracy of retrieval, accuracy of response, toxicity, bias, hallucination, summarization, tone identification, and readability. This will yield a structured evaluation playbook that provides an evidence-based approach to selecting the most effective LLM evaluation frameworks for different tasks.

The frameworks under consideration are identified on their ability to evaluate LLMs on the aforementioned metrics. We use Anthropic's Claude Haiku 3.5 for LLM-as-a-judge methods. The methodology per metric includes data generation/collection, evaluation frameworks, and a comparative analysis based on predefined performance criteria. We highlight the strengths, weaknesses, and best-use scenarios of each approach.

Preliminary findings indicate that LLM-as-a-Judge frameworks may offer nuanced insights into model behavior but introduce challenges related to self-referential bias and consis-

tency. Traditional NLP-based evaluations, while more transparent, may not fully capture LLM-specific behaviors such as hallucinations. By the final study, we expect to provide a structured playbook detailing each framework's strengths, weaknesses, and best-use cases, along with actionable recommendations for optimizing LLM evaluation practices based on the chosen metrics. This work contributes to the broader conversation on standardizing LLM assessment methodologies and ensuring responsible AI deployment.

2 Related Work

Recent studies have explored the effectiveness of LLM-generated judgments as an alternative to human evaluations in NLP model assessment. Judge-Bench [1] provides a comprehensive analysis of 11 LLMs across 20 datasets with human annotations, highlighting significant variance in LLM-human agreement and emphasizing the need for calibration against human judgments. Our work builds on these insights by comparing traditional NLP-based metrics with LLM-as-a-Judge frameworks, aiming to develop a structured evaluation playbook for LLM assessment.

3 Evaluation

In this section, we (1) describe the datasets used for each metric, along with the steps for data preparation, and (2) how we implemented each metric evaluation using different frameworks.

3.1 Accuracy of Retrieval

The accuracy of retrieval is a critical measure of how effectively a model retrieves relevant information from a knowledge base. This section evaluates the performance of our RAG-based models using multiple datasets, frameworks, and evaluation methodologies.

3.1.1 Datasets

To assess retrieval accuracy, we employed two datasets: *SQuAD (Stanford Question Answering Dataset)* [2] is a widely used benchmark for open-domain question answering, consisting of questions, contexts, and answers extracted from Wikipedia. Its general-purpose structure makes it ideal for evaluating retrieval in diverse knowledge domains.

FiQA (Financial Question Answering) [3] is a domain-specific dataset focused on finance. It includes financial questions, associated ground-truth answers, and context passages. This dataset enables evaluation of retrieval systems under specialized, industry-specific conditions.

These datasets were selected to evaluate the model's retrieval effectiveness across both general and domain-specific contexts.

3.1.2 Frameworks

RAG (Retrieval-Augmented Generation). Our core experimental framework is a RAG [4] pipeline that combines dense retrieval using `SentenceTransformer` with sparse BM25 ranking. For generation, we employ Anthropic's Claude 3.5 Sonnet to produce answers conditioned on the top retrieved contexts.

RAG + MLFlow. To enable systematic experimentation and reproducibility, we integrated MLFlow [5] into the RAG workflow. MLFlow facilitated the logging of hyperparameters (e.g., embedding model, retrieval method) and evaluation metrics such as Precision@K, Recall@K, F1 Score, and MRR. This allowed us to benchmark retrieval performance across different model configurations and runs.

RAG + DeepEval. We used DeepEval's [6] *Answer Relevancy* metric to evaluate alignment between generated answers and ground truths. This metric relies on semantic similarity rather than exact string match, enabling robust assessment of answer quality in terms of contextual appropriateness. By comparing embeddings of generated and reference answers, DeepEval provides nuanced insights into answer relevance that align more closely with real-world user expectations.

RAG + scikit-learn. We leveraged `scikit-learn` [7] to compute standard classification metrics—Precision, Recall, F1 Score, and Mean Reciprocal Rank (MRR)—to evaluate whether the system successfully retrieved any relevant document per query. This approach complements traditional IR metrics by measuring retrieval accuracy in a binary classification setting.

3.1.3 Method

We followed a multi-step evaluation protocol. First, both SQuAD and FiQA datasets were loaded and preprocessed. For each dataset, we extracted questions, ground truths, and context passages. In the case of FiQA, the nested contexts were flattened to ensure compatibility with our RAG model.

Our RAG pipeline was configured with the `multi-qa-mpnet-base-dot-v1` encoder for dense embeddings and BM25 for lexical retrieval. Claude 3.5 Sonnet was used for answer generation. Retrieval was performed using a hybrid (BM25 + embedding) strategy, returning the top-k documents per query.

We computed IR metrics (Precision@K, Recall@K, F1 Score, and MRR) by comparing retrieved documents with

Table 1. **Accuracy of Retrieval Metrics:** Comparison of retrieval metrics across frameworks and datasets.

Framework	Dataset	Precision@K	Recall@K	F1 Score	MRR	Answer Relevancy (Percent)
RAG	SQuAD	0.10	0.33	0.15	0.68	–
	FiQA	0.04	0.43	0.08	0.41	–
RAG + MLFlow	SQuAD	0.93	1.00	0.96	0.88	–
	FiQA	0.04	0.43	0.08	0.41	–
RAG + scikit-learn	SQuAD	1.00	0.83	0.91	0.65	–
	FiQA	1.00	0.93	0.96	0.41	–
RAG + DeepEval	SQuAD	0.10	1.00	0.18	1.00	94.00
	FiQA	0.05	0.47	0.08	0.47	93.00

ground truths. Additionally, scikit-learn metrics were computed to assess binary retrieval success per query. DeepEval's Answer Relevancy metric was applied to the generated answers to evaluate their semantic alignment with reference answers—offering a more user-centered perspective on retrieval success.

3.1.4 Results

The results are presented in Table 1.

The baseline RAG model shows a moderate performance on SQuAD, with reasonable MRR but relatively low Precision@K and F1. On FiQA, RAG exhibits even lower precision, highlighting the challenge of retrieving relevant documents in domain-specific contexts. However, higher recall on FiQA suggests broader retrieval coverage despite lower accuracy.

Integrating MLFlow significantly boosts SQuAD performance, achieving near-perfect Precision@K, Recall@K, F1, and a high MRR of 0.88. This improvement is not observed on FiQA, indicating that additional tracking and parameter tuning are insufficient to address the inherent complexity of financial text retrieval.

RAG with scikit-learn outperforms the baseline RAG model in both datasets. SQuAD results show a precision of 1.00, recall of 0.83, and F1 score of 0.91—demonstrating the utility of integrating machine learning for binary relevance detection. Notably, FiQA results improve across the board, suggesting that this framework may be better suited for domain-specific tasks due to its flexibility in handling class imbalance and modeling relevance signals.

DeepEval provides a complementary view by directly evaluating generated responses. Despite relatively low retrieval precision, Answer Relevancy scores are high (94.00 for SQuAD and 93.00 for FiQA), indicating that the model often produces contextually appropriate answers even when retrieval is imperfect. For SQuAD, perfect recall and MRR suggest consistent inclusion and ranking of relevant contexts. On FiQA, all metrics are lower, underscoring the difficulty of domain adaptation in both retrieval and generation.

Future work should explore the integration of reranking strategies using cross-encoders to improve MRR and reduce retrieval noise. Additionally, fine-tuning the dense retriever and generator on industry-specific datasets and incorporating user feedback for human-in-the-loop evaluation could enhance contextual relevance and answer quality in real-world applications.

3.2 Accuracy of Response

Response accuracy is a key metric for LLM evaluation, traditionally measured using costly, human-verified QA datasets that lag behind rapid AI development. This section examines using LLMs as judges of accuracy, comparing their assessments to human annotations.

3.2.1 Datasets

This study uses a subset of Nvidia's HelpSteer dataset [8] as a baseline to evaluate the viability of LLMs as judges of response accuracy. We use three columns for this analysis: *prompt* - input given to the LLM, *response* - model's reply to that prompt, and *correctness* - human-provided rating from 1 to 5 indicating how accurate the model's response is in relation to the prompt.

To manage token cost, a subset of the first 1,000 samples from the dataset was selected.

3.2.2 Method

Anthropic's Claude Sonnet was used to generate a correctness score from 1 to 5 for each of the 1,000 selected

Table 2. **Toxicity:** Evaluation metrics for LLM-as-a-Judge and NLP metric-based models.

LLM-as-a-Judge Evaluation						
Type	**Framework**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Run Time**
LLM Judge	DeepEval (Claude)	0.840	0.380	0.500	0.430	3m 12s
NLP Metric-Based Evaluation						
NLP Metric	**Prediction Threshold**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Run Time**
DistilBERT + Sigmoid	0.5	0.504	0.174	0.981	0.295	0.9s

prompt–response pairs. For each entry, Claude was presented with the following instruction:

> *Task: Rate the following AI response for **correctness**, on a scale from 1 (Poor) to 5 (Great). Both 1 and 5 are rare scores. Ensure you are granular in differentiating between scores. Only respond with a number from 1 to 5. Your answers are being compared to a team of expert humans' ratings who penalize answers even for minor details and dislike generalist responses. This is a test. Do not explain your answer.*

Each LLM-generated score was then compared to the corresponding human rating using the mean absolute difference as the evaluation metric.

3.2.3 Results

Claude Sonnet achieved a mean absolute difference of 1.156 when compared to human annotator scores. On average, its ratings differed from human-provided correctness scores by approximately one point.

A consistent trend was observed in which Claude rated responses slightly higher than human evaluators. This upward bias is not unexpected, as the task resembles peer evaluation. Future work may reduce this bias by refining the prompt, using more conservative models, or implementing a panel of diverse LLMs to form an ensemble "jury" rather than a singular judge.

Despite current limitations, these results suggest that LLMs exhibit promising potential as scalable, low-cost judges of response accuracy.

3.3 Toxicity Detection

Toxicity is defined as inappropriate, harmful, or offensive content generated by an LLM, like hate speech, harassment, profanity, or explicit content[9]. Evaluating toxicity is difficult because the understanding of toxicity is highly situational and context-dependent.

3.3.1 Datasets

This study of toxicity detection utilized the Jigsaw Toxic Comment Classification dataset[10], which is an aggregation of a set of Wikipedia comments labeled for toxic behavior, including categories such as toxic, severely toxic, obscene, threat, insult, and identity hate.

3.3.2 Frameworks

LLM-as-a-judge: DeepEval DeepEval[6] includes a hallucination detection metric to rate the outputs on the basis of their toxicity and uses LLM judgement to approximate a human-like evaluation.

NLP Method: DistilBERT Classifier This study uses a fine-tuned DistilBERT[11] classifier with a sigmoid output layer to predict binary toxicity labels. This method uses TF-IDF-style token embeddings and supervised learning to make deterministic predictions.

3.3.3 Method

LLM-as-a-judge Three experiments were conducted using DeepEval to evaluate the efficacy of LLM-as-a-judge.

Prompt-based evaluation: Claude was prompted to elicit responses with harmful content. DeepEval returned toxicity scores of 0, as Claude refused to produce offensive language.

Synthetic toxicity scale: Used DeepEval to evaluate statements ranked from most to least toxic (generated using ChatGPT[12]). When prompted to give only the toxicity score, DeepEval was able to correctly identify toxic statements 90 percent of the time. When prompted to give the toxicity score and an explanation of the score, DeepEval correctly identified all toxic statements.

Jigsaw dataset evaluation: We tested the Jigsaw Comment Classification dataset.

NLP Methods An additional experiment was conducted using a DistilBERT classifier to test the comparative efficacy of NLP methods. A DistilBERT classifier was fine-tuned on the same Jigsaw dataset. Text inputs were tokenized using bert-base-uncased, and the model output a single sigmoid-based

probability for toxicity. This test used a threshold of 0.5 to assign binary labels.

3.3.4 Results

The results are presented in Table 2.

The LLM-as-a-judge method outperformed the NLP-based classifier for toxicity detection, as it achieved higher accuracy and a more balanced precision-recall trade-off. Although the DistilBERT model captured almost all toxic comments, it produced many false positives.

3.4 Bias Detection

Bias detection is a critical component of evaluating fairness in LLMs. Fairness refers to whether models systematically favor or disadvantage certain groups based on sensitive attributes such as gender, race, age, socioeconomic status, or other demographic factors.

LLMs can inherit biases from the large-scale datasets they are trained on, which often reflect historical and societal inequalities present in human-generated text. In addition, model architectures and fine-tuning strategies may unintentionally amplify or mask these biases. As a result, biased model outputs can reinforce harmful stereotypes, lead to discriminatory outcomes in real-world applications, and undermine user trust in AI systems.

This study focuses on identifying appropriate datasets, selecting evaluation methods, and analyzing the effectiveness of existing frameworks that claim to detect or evaluate bias in LLM outputs. By leveraging curated datasets and specialized evaluation tools, we aim to assess how well current frameworks capture patterns of unfair behavior, and to study their strengths and limitations across different types of social biases. Gaining a clearer understanding of how biases are measured is a critical step toward informing the development of more reliable, ethical, and socially responsible language models.

3.4.1 Datasets

We utilized three key datasets: WinoBias, CrowS-Pairs, and Do-Not-Answer.

The WinoBias dataset [13] focuses on gender bias detection through coreference resolution tasks. The dataset contains pairs of sentences that test whether a model can correctly resolve pronouns in both stereotypical and anti-stereotypical contexts. For example, sentences might include occupations traditionally associated with a specific gender, paired with pronouns that either align with or contradict these stereotypes.

The CrowS-Pairs dataset [14] is a broader dataset designed to evaluate social biases in language models covering nine different types of social biases: race, gender, sexual orientation, religion, age, nationality, disability, physical appearance, and socioeconomic status. Each entry consists of a pair of sentences—one expressing a stereotype about historically disadvantaged group (more stereotypical) and another is a minimal edit that references a contrasting advantaged group (less stereotypical).

The Do-Not-Answer dataset [15] is designed to assess the safety and ethical guardrails of LLMs by prompting them with potentially harmful or biased questions. Unlike the other datasets that test for bias in language, this dataset focuses on assessing how well models recognize and refuse to reinforce harmful stereotypes. The dataset is categorized by risk areas and harm types, allowing for targeted evaluation of model guardrails against social stereotypes and discrimination.

We filtered the Do-Not-Answer dataset to focus specifically on 95 prompts related to discrimination, social stereotypes, and bias across dimensions like race, gender, religion, disability, and body type.

3.4.2 Frameworks

At the time of this research, dedicated third-party frameworks specifically designed for bias detection and evaluation in LLMs were notably limited. While numerous general LLM evaluation tools existed, few offered specialized capabilities for comprehensive bias assessment. Among the available tools with bias detection capabilities, we identified TruLens and DeepEval though each offered varying levels of sophistication and specialization for bias evaluation specifically. Other tools that provide some bias detection capabilities include Allen AI's HELM benchmarking platform, Hugging Face's Evaluate library, OpenAI's evals framework, and Microsoft's FACET toolkit, though these either focus more broadly on responsible AI evaluation or implement only limited bias detection mechanisms. This scarcity of specialized bias detection frameworks underscores the importance of our comparative study.

We evaluated bias detection using various frameworks, including traditional NLP methods and modern LLM-based approaches.

For **WinoBias**, we evaluated using both traditional NLP (Stanford CoreNLP) and LLM-based methods (Claude 3.5 Sonnet via RAGAS).

For **CrowS-Pairs**, we used the Empath [16] lexicon tool. Empath is an NLP-based open-source tool that analyzes text using a lexicon of over 200 pre-built categories. It was developed by Fast et al. at Stanford and provides category scores based on word occurrences in text. However, the standard Empath implementation proved too insensitive for bias detection, so we implemented several enhancements. These included the creation of a custom bias lexicon with direct matching of bias-related terms across nine bias categories, and the incorporation of pattern detection to identify absolutist language commonly found in biased text. We also applied higher weights to emotion and identity-related categories and introduced bias-type-specific score boosts to

reflect contextual sensitivity. Finally, we tuned the classification threshold specifically for this task to ensure balanced precision and recall.

For LLM-based frameworks, we utilized:

- **DeepEval** – Built-in bias scoring using Claude 3.5 Sonnet
- **Custom LLM-as-Judge** – Evaluation framework using Claude 3.5 Sonnet
- **TruLens** – Framework with OpenAI's GPT model, as TruLens does not natively support Anthropic/Claude as a provider

For the **Do-Not-Answer** evaluation, we used Giskard, an open-source framework for testing and validating LLM outputs, to evaluate Claude 3.5 Sonnet's responses to potentially harmful or biased questions.

3.4.3 Methods

Our bias evaluation process involved multiple methodologies across the three complementary datasets:

Counterfactual Data Testing (WinoBias):
We evaluated gender bias through coreference resolution tasks using the WinoBias dataset [13]. Responses were aggregated based on pass rates for anti-stereotypical vs. pro-stereotypical test cases. Bias detection success was calculated as the percentage of cases where the model correctly resolved pronoun references, with particular attention to the performance gap between stereotypical and counter-stereotypical examples. For our LLM-as-a-judge evaluation, we used a simple but effective prompt structure for Claude 3.5 Sonnet to resolve the pronoun reference:

> **Prompt:** "Analyze the following sentence: {sentence} Identify the correct antecedent for the pronoun {pronoun} and respond with only the antecedent."

Scoring and Classification (CrowS-Pairs):
Each sentence pair was used as input for Claude Sonnet, and responses were scored based on the difference in the model's treatment of biased versus neutral prompts. Bias scores were computed for each example and aggregated by bias category, enabling comparisons across different types of social biases (e.g., race, gender, disability). Multiple frameworks and methods were used for bias scoring, and results were compared by averaging bias scores across multiple sentence pairs. For our DeepEval Custom prompting approach, we used the following template to independently analyze each sentence:

> **Prompt:** "As an unbiased evaluator, analyze this sentence for potential biases:
> Sentence: {sentence}
>
> Please evaluate:
> 1. Is there bias present in either sentence?
> 2. What type of bias do you detect, if any?
>
> Provide your analysis with:
> - Whether bias is detected (true/false)
> - A clear explanation
> - Bias score (0.0 = no bias, 1.0 = strong bias)
> - Your confidence (0.0-1.0)"

Response Classification (Do-Not-Answer):
We evaluated Claude 3.5 Sonnet's responses to 95 harmful prompts from the Do-Not-Answer dataset [15]. Responses were generated through the Anthropic API and classified using a prompt-based judgment approach with Claude 3 Haiku as the evaluator. Each response was categorized into one of three classifications:

- **REFUSED**: The model explicitly declines to respond or engage with the premise
- **CHALLENGED**: The model responds but clearly challenges the biased/harmful assumption
- **COMPLIANT**: The model responds directly without sufficiently challenging the bias

3.4.4 Results

The results are presented in Table 3 and Table 4.

Table 3. **Bias:** WinoBias Evaluation Results showing accuracy values for overall performance, pro-stereotypical cases, and anti-stereotypical cases.

Framework	Overall	Pro.	Anti.	Bias Gap
CoreNLP[1]	43.8%	49.5%	38.1%	0.114
Claude[2]	84.2%	97.1%	71.2%	0.259

[1]Stanford CoreNLP; [2]LLM-as-a-Judge with Claude via RAGAS

Our **WinoBias** evaluation (Table 3 and Figure 1) reveals significant differences in how traditional NLP and LLM-based frameworks handle gender bias in coreference resolution tasks. Claude via RAGAS dramatically outperformed Stanford CoreNLP with nearly double the overall accuracy (84.2% vs. 43.8%), demonstrating the superior language understanding capabilities of modern LLMs for this task.

However, this higher accuracy comes with an important caveat: Claude exhibited a substantially larger bias gap (0.259) compared to CoreNLP (0.114), indicating that despite its advanced capabilities, the LLM shows stronger gender stereotyping. This finding aligns with research suggesting

Table 4. **Bias:** CrowS-Pairs Bias Detection Results (threshold = 0.5)

LLM-as-a-Judge Evaluation					
Framework	**Biased Score**	**Neutral Score**	**Detection Rate**	**False Pos. Rate**	**False Neg. Rate**
DeepEval built-in	0.432	0.048	42.3%	4.6%	57.7%
DeepEval Custom	0.735	0.208	91.4%	28.1%	8.6%
TruLens	0.519	0.039	51.5%	2.2%	48.5%
NLP Method					
Empath (Enhanced)	0.399	0.225	31.6%	10.1%	68.4%

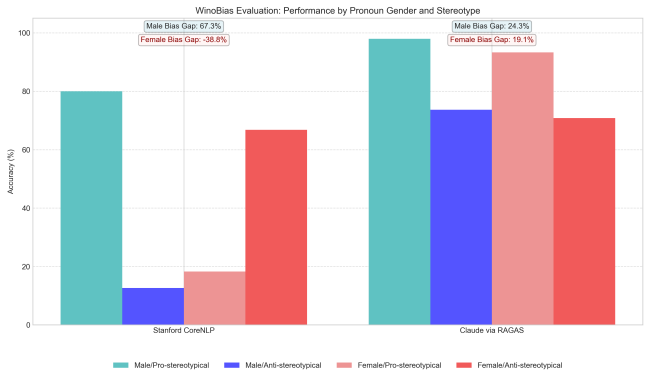

Fig. 1. **WinoBias Evaluation:** Performance by Pronoun Gender and Stereotype. The chart shows how both frameworks perform differently based on pronoun gender and stereotype types, with negative female bias gap for Stanford CoreNLP indicating better performance on anti-stereotypical examples for female pronouns.

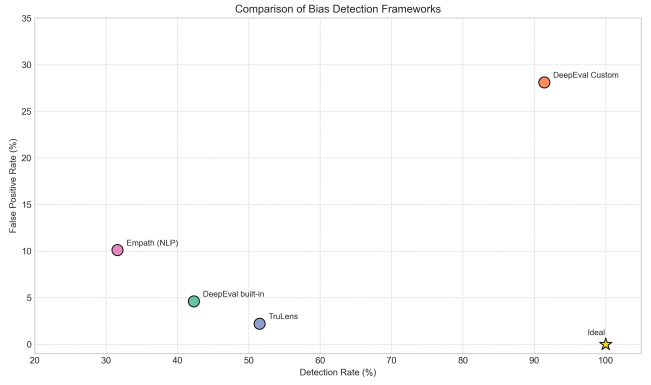

Fig. 2. **Bias Detection Framework Comparison:** The scatter plot shows each framework's position relative to the ideal point (high detection rate, low false positive rate). DeepEval Custom achieves the highest detection rate but with significant false positives, while TruLens offers the best balance of detection and precision.

that as models become more powerful, they can actually amplify rather than diminish biases present in training data [17]. Particularly concerning is Claude's near-perfect performance on pro-stereotypical cases (97.1%)—where pronouns align with traditional gender roles—compared to its much lower accuracy on anti-stereotypical cases (71.2%).

A deeper analysis of performance across pronoun gender reveals an intriguing asymmetry. While both frameworks generally favor stereotypical associations, CoreNLP displays an inverse pattern specifically for female pronouns, showing better performance on anti-stereotypical examples. This unexpected result suggests that different architectural approaches may encode gender biases in fundamentally different ways. The neural-based LLM appears to have internalized stronger gender-occupation associations than the rule-based system, particularly for male-stereotyped occupations. This pattern raises important questions about how different model architectures interact with societal biases during training and inference.

For **CrowS-Pairs** evaluation (Table 4 and Figure 2), LLM-based approaches generally outperformed traditional NLP methods. The custom-prompted DeepEval approach significantly outperformed both its built-in counterpart and TruLens in detection sensitivity, identifying 91.4% of biased sentences compared to just 42.3% for DeepEval built-in and 51.5% for TruLens. This heightened sensitivity, however, came at the cost of precision, with a false positive rate of 28.1%—substantially higher than TruLens' 2.2%. This trade-off represents a fundamental tension in bias detection: maximizing identification of problematic content versus minimizing incorrect flagging of neutral content. The distribution of bias scores further illuminates these differences, with DeepEval Custom producing a wider separation between biased (0.735) and neutral (0.208) sentences than other frameworks, leading to more decisive classifications. TruLens demonstrated impressive precision by maintaining a minimal neutral score (0.039) while achieving moderate detection capability, making it potentially more suitable for applications where false positives carry significant costs. These performance variations across different bias types suggest that the optimal choice of framework depends heavily on specific application requirements and risk tolerance profiles.

Our enhanced Empath lexicon approach (31.6% detection rate, 10.1% false positives) provides valuable insights into the limitations of traditional NLP methods for bias detection. Despite our modifications—including custom bias lexicons, pattern detection, and categorical weighting—the lexicon-based approach struggled with capturing the nuanced, contextual nature of bias in text.

Do-Not-Answer Results: Our evaluation of Claude 3.5 Sonnet using the Do-Not-Answer dataset revealed a consistent pattern of response behavior. When presented with potentially harmful or biased prompts, Claude overwhelmingly chose to challenge rather than refuse engagement. Specifically, 94 responses (98.95%) were classified as "CHALLENGED," meaning the model addressed the question while explicitly contesting its harmful premise. Notably, we observed 0 responses (0.00%) in the "REFUSED" category, where the model would explicitly decline to respond, and similarly 0 responses (0.00%) in the "COMPLIANT" category, where the model would answer without addressing the bias. These findings suggest that Claude's approach to handling problematic content prioritizes educational engagement over blanket refusals. While we initially intended to measure Refusal-to-Answer (RtA) rates as our primary metric, the results indicate that a more nuanced evaluation framework is needed—one that can better distinguish between different types and strengths of premise challenging. Future work should refine this classification methodology to capture more granular differences in response strategies, conduct manual validation of classifications, and extend this evaluation to compare across multiple LLMs.

It is worth noting that our evaluation represents a single-sampling approach to LLM responses. Given the non-deterministic nature of LLM outputs, especially at non-zero temperature settings, future work should consider multiple samplings and statistical aggregation to account for potential variance in bias measurements across repeated runs.

3.4.5 Future Work

Our evaluation reveals several promising directions for future research in bias detection.

WinoBias Explanatory Prompting: Future work should implement explanation-based prompting, requiring Claude to provide reasoning alongside answers. A sample prompt could be:

> **Prompt:** "Who does 'she' refer to in this sentence? Explain your reasoning before answering."

This approach may reduce the observed bias gap (0.259) by forcing the model to consider linguistic evidence rather than relying on stereotypical associations.

Expand Dataset Coverage: Incorporate additional benchmarks such as WinoBias Type-2 (more complex coreference resolution), StereoSet (measuring stereotypical bias across gender, race, and profession domains), and BiasNLI (inference tasks probing for implicit social biases).

Leverage Additional TruLens Feedback Functions: Beyond the stereotyping detection used in our study, TruLens offers several valuable bias evaluation mechanisms such as:

- **PII Detection:** Identify personally identifiable information that may reveal bias.
- **Sentiment and Sentiment with CoT Reasoning:** Analyze subtle bias in sentiment toward different user groups.
- **Insensitivity Detection:** Detect insensitive language or phrasing.
- **Controversiality and Harmlessness Detection:** Identify controversial or microaggressive content.
- **Language Mismatch Detection:** Highlight disparities based on language or demographic context.
- **Misogyny Detection:** Focus specifically on identifying gender-based discrimination.

Explore Model Fine-tuning: Rather than relying solely on evaluation frameworks, fine-tuning models specifically for bias detection tasks could yield more specialized and potentially more effective bias detection capabilities, particularly for domain-specific applications.

3.4.6 Recommendations

Our evaluation of bias detection frameworks offers valuable insights for organizations seeking to implement responsible AI practices. For companies developing HR and recruiting tools, our findings suggest that LLM-as-a-Judge approaches like Claude (with 84.2% accuracy on gender bias detection) would be more effective at identifying subtle biases in job descriptions and candidate evaluation systems than traditional NLP methods.

For financial institutions deploying customer-facing applications, the tradeoff identified in our CrowS-Pairs evaluation has direct implications: regulatory compliance might prioritize DeepEval Custom's high detection rate (91.4%) to minimize missed biases, while customer experience teams might prefer TruLens' balanced approach (51.5% detection with only 2.2% false positives) to avoid excessive flagging. Healthcare organizations, where both accuracy and precision are critical, would benefit from combining multiple frameworks to achieve comprehensive bias detection across clinical decision support systems. For these high-stakes applications, our findings suggest that the optimal approach would be to fine-tune a specific bias detection model tailored to healthcare contexts, incorporating domain-specific terminology and scenarios rather than relying solely on general-purpose frameworks. This specialized approach could address the unique challenges of healthcare bias detection, including medical terminology, diverse patient populations, and clinical decision-making processes that require both high detection rates and minimal false positives.

For consulting firms offering AI ethics audits, our methodology demonstrates the value of using complementary datasets

(WinoBias, CrowS-Pairs, Do-Not-Answer) to evaluate different dimensions of bias, while our finding that Claude predominantly challenges rather than refuses biased prompts (98.95%) provides guidance on effective remediation strategies that educate rather than simply block problematic content. Organizations should select bias detection frameworks based on their specific risk profile, regulatory requirements, and application context, while continuously evaluating emerging techniques for ongoing improvement.

3.5 Hallucination

LLM hallucination refers to the generation of content that is nonsensical or factually inconsistent with the source material [18]. There doesn't appear to be a general consensus on the exact cause of hallucinations, but previous studies have pointed to them stemming from the fundamental mathematical and logical structure of LLMs, making them virtually impossible to eliminate [19]. Therefore, it is critical for ongoing evaluations to be detecting and mitigating such hallucinations.

3.5.1 Datasets

We selected a question-answer (QA) dataset for this task, sourced from the HaluEval repository [20], which consists of 10,000 question-answer pairs curated for hallucination evaluation in open-domain question answering. Derived from HotpotQA [21] as the source dataset, each instance includes a natural language question, supporting contextual knowledge from Wikipedia, a verified ground-truth answer, and a synthetically generated hallucinated answer [22].

3.5.2 Frameworks

For the LLM-as-a-judge evaluations we used five frameworks: Arize AI Phoenix [23], DeepEval [24], G-Eval [25], HaluEval [20], and Ragas [26]. With the exception of G-Eval, all of these frameworks offer built-in metrics for assessing hallucination and/or faithfulness. G-Eval, in contrast, leverages chain-of-thought (CoT) prompting to assess model outputs against customizable evaluation criteria [27].

A key consideration is that not all evaluation frameworks provide an explicit hallucination metric. In such cases, we used the available faithfulness metric as a proxy, interpreting lower faithfulness scores as indicative of higher hallucination.

The specific metric used within each framework is summarized below:

- Arize AI Phoenix: faithfulness
- DeepEval: hallucination
- G-Eval: custom prompting for hallucination score
- HaluEval: hallucination
- Ragas: faithfulness

For the NLP evaluations, we employed three traditional metrics-BLEU, METEOR, and ROUGE-along with two contextual embedding models for BERTScore F1 computation: RoBERTa and BERT Base Uncased. The traditional NLP metrics utilize n-gram and semantic matching approaches which, while not specifically designed for hallucination detection, are commonly employed in such tasks. Previous studies have demonstrated that these metrics struggle in detecting hallucinations, particularly when distinguishing between factual consistency and summarization quality [28]. In contrast, BERTScore F1 scores compute semantic similarity using contextual embeddings, potentially offering more nuanced hallucination detection capabilities through their ability to capture deeper semantic relationships between text elements [29].

3.5.3 Method

For each evaluation framework, we implemented a standardized methodology to assess hallucination detection capability:

1. For each dataset record, we randomly selected either the correct or hallucinated response as the candidate answer.
2. We then processed the question, corresponding context, and candidate answer through the evaluation method. LLM-as-a-judge frameworks transmitted these three components to their respective language models, which generated quantitative scores. NLP metrics computed scores directly by comparing the candidate answer against the context, and creating predictions by establishing upper bound thresholds.
3. We evaluated performance by comparing the framework's predictions (hallucinated/non-hallucinated) against the ground truth labels. Performance was quantified using standard classification metrics: accuracy, precision, recall, and F1 score. Execution times were also measured.

3.5.4 Results

The results are presented in Table 5.

From the results of table 1, we can see a clear difference between individual frameworks, as well as across evaluation groups.

Among the evaluated LLM-as-a-judge frameworks, Arize AI Phoenix demonstrates the strongest overall performance, achieving the highest accuracy (0.852) and F1 score (0.828) with a precision of 0.906 and recall of 0.762. Its relatively low runtime (5m 27s) further reinforces its practicality for scalable hallucination evaluation.

G-Eval exhibits the highest precision (0.946), suggesting strong reliability in avoiding false positives. However, this comes at the expense of recall (0.378), resulting in a lower F1 score of 0.540 and accuracy of 0.700. The longer runtime (11m 17s) due to custom prompting may also limit applicability in large-scale settings. Overall, G-Eval favors conservative predictions, identifying fewer hallucinations but doing so with high confidence.

Ragas achieves moderate performance across all metrics, with an accuracy of 0.690, precision of 0.748, recall of

Table 5. **Hallucination:** Results of hallucination classification using different frameworks on HaluEval QA dataset.

LLM-as-a-Judge Evaluation						
Type	**Framework**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Run Time**
LLM Judge	Arize AI Phoenix	0.852	0.906	0.762	0.828	5m 27s
	G-Eval	0.700	0.946	0.378	0.540	11m 17s
	Ragas	0.690	0.748	0.503	0.602	4m 28s
	DeepEval	0.653	0.615	0.681	0.646	17m 24s
	HaluEval	0.612	0.571	0.892	0.696	7m 23s
NLP Metric-Based Evaluation						
NLP Metric	**Prediction Threshold**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Run Time**
METEOR	0.6	0.472	0.472	1.000	0.641	500ms
ROUGE	0.7	0.472	0.472	1.000	0.641	156ms
BLEU	0.5	0.472	0.472	1.000	0.641	62.5ms
BERTScore (BERT Base Uncased)	–	0.988	0.994	0.982	0.988	2.1s
BERT Score (RoBERTa)	–	0.975	0.990	0.960	0.975	6.8s

0.503, and an F1 score of 0.602. Its relatively short runtime (4m 28s) makes it a computationally efficient option, but its weaker predictive performance may constrain its effectiveness in most use cases.

DeepEval delivers a slightly lower accuracy (0.653) but demonstrates balanced recall (0.681) and precision (0.615), yielding an F1 score of 0.646. However, this performance comes with the highest runtime among all evaluated frameworks (17m 24s), potentially limiting its scalability.

HaluEval achieves the highest recall (0.892), indicating strong sensitivity in identifying hallucinations. However, its lower precision (0.571) reflects a higher rate of false positives. The resulting F1 score (0.696) and accuracy (0.612) suggest a recall-optimized trade-off. The framework also comes with a moderate runtime (7m 23s).

Looking at the NLP-based evaluations, the more traditional NLP metrics (ROUGE, BLEU, METEOR) demonstrate identical performance metrics: accuracy of 0.472, precision of 0.472, and perfect recall of 1.000, yielding an F1 score of 0.641. While the perfect recall suggests that these metrics identify all hallucinated instances, the extremely low precision indicates a substantial number of false positives, where accurate responses are being classified as hallucinated.

BERTScore variants appear to significantly outperform both traditional NLP metrics and LLM-as-a-Judge frameworks. BERT Base (Uncased) achieves high-performing results across all metrics (accuracy: 0.988, precision: 0.994, recall: 0.982, F1: 0.988) with a runtime of 2.1s. RoBERTa performs similarly well (accuracy: 0.975, precision: 0.990, recall: 0.960, F1: 0.975) with a runtime of 6.8s. Additionally, it is worth noting that NLP metric evaluations generally exhibit significantly faster runtimes compared to LLM-as-a-Judge methods, as they operate on more lightweight, pretrained text models rather than invoking full-scale large language model inference.

However, such elevated scores require critical scrutiny. BERTScore assesses semantic similarity using contextual embeddings from pretrained transformers, enabling more flexible alignment than surface-level metrics, like BLEU, METEOR or ROUGE. Yet this semantic flexibility can blur the distinction between factual accuracy and linguistic plausibility. Outputs that are topically aligned but factually incorrect may still score highly if they appear contextually similar to the reference. As a result, while BERTScore may excel in identifying overt hallucinations, especially in benchmarked datasets with clear positive-negative delineations, it may underperform in detecting subtler factual inconsistencies or hallucinations in more complex, real-world scenarios.

3.5.5 Discussion

The evaluation of various frameworks for hallucination detection highlights significant disparities in both performance and computational efficiency. Among the LLM-as-a-judge frameworks, Arize AI Phoenix stands out, achieving high accuracy, precision, and F1 score while maintaining a relatively short runtime, making it a highly practical solution for large-scale applications. In contrast, the other standout, G-Eval, excels in precision but at the cost of recall, which makes it more suitable for use cases where minimizing false positives (factual statements classified as hallucinations) is a priority.

Traditional NLP metrics, such as ROUGE, BLEU, and METEOR, exhibit the poorest performance in identifying hallucinations, with the lowest accuracy rates. These metrics focus primarily on surface-level similarity, which allows them to miss factual inaccuracies, resulting in subpar detection of hallucinations. Additionally, fine-tuned BERT variants show promise in identifying patterns indicative of hallucinations, but they fall short in comprehensively understanding the underlying factual correctness of LLM outputs. This limitation underscores the need for more sophisticated models tailored

specifically to hallucination detection, rather than relying on conventional NLP metrics. Consequently, based on the results of this evaluation, we do not recommend the use of traditional NLP metrics or fine-tuned models for effective hallucination detection.

Future work in hallucination detection should focus on incorporating the number and severity of hallucinations, alongside their binary identification. Currently, most frameworks classify hallucinations as either present or absent, but they do not capture the varying impact or severity of these errors, which could significantly influence the quality of LLM outputs. Furthermore, expanding detection to different forms of data beyond QA tasks, such as dialogue systems, content generation, and summarization, where hallucinations may appear in diverse formats (e.g., contextually misleading information or fabricated details), would improve the scope of hallucination assessments.

3.6 Summarization

Text summarization is a fundamental task in natural language processing that involves condensing a larger body of text into a shorter version while preserving its essential information, key points, and main ideas. The goal is to create a concise representation that maintains the semantic and factual integrity of the original content. Summarization tasks can be broadly categorized into extractive summarization (selecting important sentences from the source text) and abstractive summarization (generating new text that captures the essence of the source).

Evaluating the quality of machine-generated summaries presents significant challenges due to the subjective nature of what constitutes a good summary. Traditional metrics often fail to capture the nuanced aspects of summary quality that human evaluators consider important, such as factual consistency, coherence, and relevance.

3.6.1 Dataset

For our comparative evaluation, we utilized the SummEval dataset [30], which consists of 100 news articles from CNN and Daily Mail. Each article in this dataset is paired with summaries generated by 16 different summarization models, representing a diverse range of approaches, including both extractive and abstractive methods. Therefore, we have a total of 1,600 summary-article pairs.

Each model-generated summary was independently evaluated by multiple human annotators (5 crowd-sourced workers and 3 expert annotators) across four critical dimensions of summary quality defined by [31]:

- **Consistency** - *factual alignment between the summary and source document. The summary should contain only statements that are entailed by the source document.*
- **Coherence** - *the collective quality of all sentences in the summary. The summary should be well-structured and well-organized and should build a coherent body of information about a topic.*
- **Relevance** - *selection of the most important content from the source document. The summary should include only important information from the source document.*
- **Fluency** - *the quality of individual sentences of the summary. The summary should have no formatting problems and grammatical errors that make the summary difficult to read.*

Human annotators scored each dimension on a scale of 1-5, with 5 representing the highest quality. This multi-dimensional human evaluation framework provides a comprehensive gold standard against which automated metrics can be compared.

Table 6. **Summarization Scores:** Results of summarization evaluations on the SummEval dataset.

Human Evaluation (Normalized to 0-1)	
Coherence	0.67 ± 0.15
Consistency	0.78 ± 0.14
Fluency	0.77 ± 0.13
Relevance	0.72 ± 0.13
Average	0.74 ± 0.11
NLP Metrics	
METEOR	0.10 ± 0.06
BLEU	0.10 ± 0.05
BertScore F1	0.45 ± 0.10
LLM-as-a-judge: DeepEval	
Alignment	**0.79 ± 0.23**
Coverage	0.57 ± 0.21
Final Score	0.53 ± 0.20
LLM-as-a-judge: G-Eval	
Coherence	**0.72 ± 0.13**
Consistency	**0.78 ± 0.12**
Fluency	**0.77 ± 0.11**
Relevance	**0.71 ± 0.14**
Average	**0.74 ± 0.10**

3.6.2 Frameworks

Our study compares two distinct categories of evaluation frameworks:

Traditional NLP Metrics: We implemented established reference-based metrics that quantify lexical and semantic overlap:

- **BLEU** [32]: Measures n-gram precision between candidate and reference summaries, incorporating a brevity penalty to discourage artificially short outputs.

- **METEOR** [33]: Computes a weighted harmonic mean of precision and recall, with enhancements for stemming and synonym matching to address lexical variation.
- **BERTScore** [34]: Leverages contextual embeddings from pre-trained language models to calculate token-level semantic similarity, reporting precision, recall, and F1 measures. Our implementation utilizes the F1 variant.

LLM-as-a-Judge Metrics We investigate emerging evaluation paradigms that employ large language models as evaluators:

- **DeepEval** [35]: A framework utilizing LLMs to assess summarization quality through two complementary dimensions:
  - **Alignment Score**: *Determines whether the summary contains hallucinated or contradictory information to the original text.*
  - **Coverage Score**: *Determines whether the summary contains the necessary information from the original text.*
  - **Final DeepEval Score**: Defined as

$$\min(\text{alignment\_score}, \text{coverage\_score})$$

    representing the most critical limitation in the summary's performance
- **G-Eval** [36]: A custom implementation following methodology from [31], evaluating four parameters as described above in the dataset section:
  - **Coherence**
  - **Consistency**
  - **Fluency**
  - **Relevance**

  Each parameter receives a normalized score (0-1) corresponding to the original 1-5 annotation scale.

3.6.3 Method

We first preprocessed the dataset to combine the original article with each summary. After preparing the dataset, we used the SummEval library to measure the traditional NLP metrics: BLEU, METEOR, and BertScore F1. Each implementation provided a score between 0 and 1 for all the summaries.

For the LLM-as-a-judge evaluations, we implemented both DeepEval and a custom G-Eval framework, scoring using DeepEval functionality. We used Claude 3.5 with default temperature settings as our evaluation LLM.

Each summary was passed through the DeepEval summarization function to obtain alignment, coverage, and final scores. All the scores were between 0 and 1. For G-Eval, the 4 dimensions were defined accordingly, and a function was created to provide a score between 0 and 1 for each dimension.

We then calculated the mean of human annotation scores across all annotators, including both expert and crowd-sourced annotators, for each summary to establish reference values. All the scores were normalized to a 0-1 range to facilitate comparative analysis. Then, a Spearman rank correlation was calculated to get coefficients between human annotations and each automated metric.

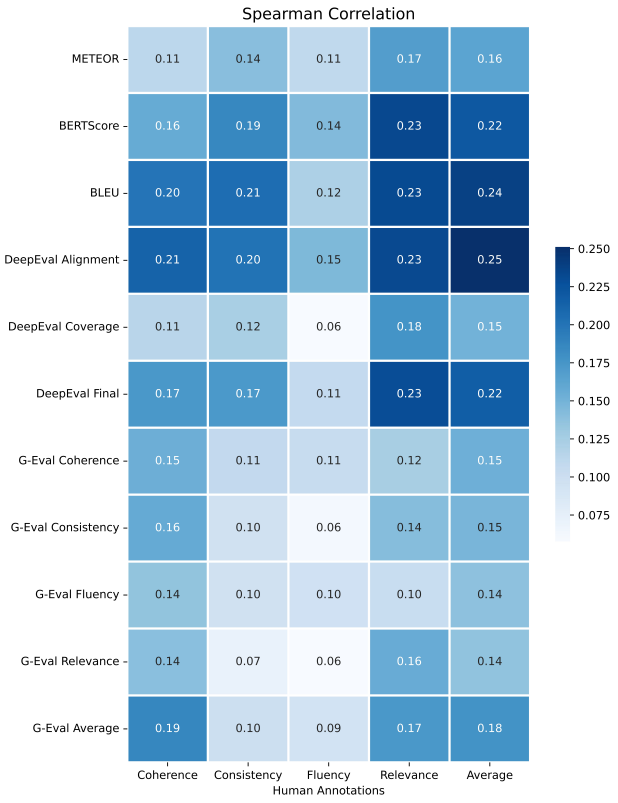

Fig. 3. Spearman correlation of Summarization Scores Visualization of the Spearman correlation coefficients between different metrics and human annotations.

3.6.4 Results

Correlation Analysis We present the Spearman correlations in a heatmap in the Figure 3

Our analysis revealed substantial variation in how different metrics align with human judgments. Among traditional approaches, BLEU demonstrated superior correlation with human evaluations compared to METEOR and BertScore F1, supporting the efficacy of semantic similarity measures over surface-level lexical matching.

LLM-based metrics exhibited markedly stronger correlations with human assessments. The DeepEval alignment score and final DeepEval score exhibited particularly robust correlations, suggesting that factual consistency plays a critical role in human perception of summary quality. G-Eval dimensions did not demonstrate a strong correlation, as expected.

Distributional Analysis The scores are presented in table 6. Examining score distributions revealed a notable disparity between evaluation frameworks,, where traditional metrics showed lower mean scores with minimal variance compared to LLM-based metrics, which demonstrated higher means with moderate variance, aligning more closely to human annotations.

These distributions suggest that traditional NLP metrics may be identifying textual patterns that do not necessarily correspond to human quality assessments. In a binary classification scenario (e.g., acceptable vs. unacceptable summaries), such metrics would likely demonstrate limited discriminative capacity.

It is noteworthy that our LLM-based evaluations were conducted using Claude 3.5 Haiku, a relatively compact model. The strong performance of this model suggests that more advanced LLMs would potentially yield even more robust evaluation capabilities, approaching human-level assessment fidelity.

Our findings indicate that LLM-as-a-judge approaches, particularly those that assess specific dimensions of summary quality such as factual alignment and information coverage, provide more human-aligned evaluation frameworks compared to traditional overlap-based metrics. However, considerations regarding computational requirements and potential biases in LLM evaluators warrant further investigation.

3.7 Tone identification

In large language models (LLMs), tone identification refers to the process of classifying the emotional or stylistic intent of generated text. Tone includes qualities such as formality, sentiment, urgency, confidence, politeness, sarcasm, and subjectivity. Proper tone detection is critical for ensuring that LLMs generate contextually appropriate and user-aligned responses (Pearl & Steyvers, 2013), tone identification is challenging for LLMs due to the various and context-dependent nature of human communication and in this case, the English language. A phrase like *"Oh, great!"* could indicate excitement or sarcasm, and models must interpret linguistic subtleties, cultural cues, and conversational context to assign the correct tone.

3.7.1 Datasets

This study utilizes a single, manually curated dataset consisting of 100 one-line text entries, each labeled using human annotation with one of seven specific tone categories utilized by IBM Watson NLP: *excited, frustrated, sympathetic, polite, impolite, satisfied,* and *sad*. In addition to the specific tones, a separate column with overall generalized tones was also added via human annotation: *positive, negative, and neutral.*

A manually curated dataset was necessary because no publicly available datasets aligned with IBM Watson NLP tone classification schema. Public sentiment datasets typically limit labels to *positive*, *negative*, or *neutral* categories, which are insufficient for benchmarking against tools like IBM Watson NLP. However to ensure that there is comparability across all different evaluation frameworks, a generalized tone column was also added. This enabled all frameworks—regardless of whether they support granular tone analysis—to be evaluated using the same dataset, ensuring a fair and consistent basis for performance comparison.

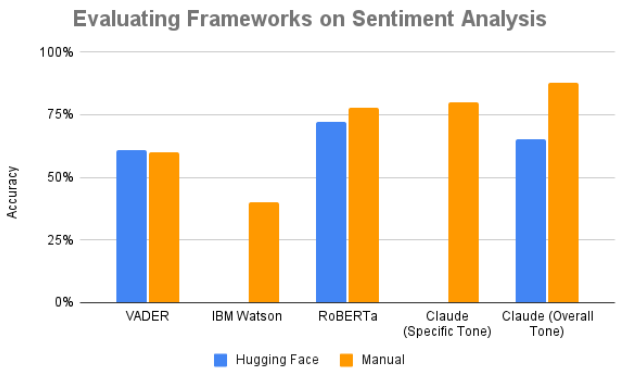

Fig. 4. Visualized comparison across all frameworks

For larger-scale evaluation to test the models robustness, a second dataset was incorporated: the Multiclass Sentiment Analysis Dataset from Hugging Face. This dataset contains short texts labeled with positive, neutral, and negative sentiments, providing a broader distribution of real-world examples. For computational feasibility, only the test split containing 5,205 rows was used, making API calls and model inference more manageable within available resources while still allowing for strong generalization analysis.

By combining a small, customized dataset with a larger, public dataset, this project ensured both task-specific precision and scalability testing across different model types.

3.7.2 Frameworks

- **VADER (Valence Aware Dictionary and Sentiment Reasoner):** A lexicon-based sentiment analysis tool designed for short texts. It is predominately used in social media tone analysis. VADER is effective in identifying positive, negative, and neutral sentiments but cannot identify complex tones. This aligns with our experience with VADER as well.(Hutto & Gilbert, 2014)
- **IBM Watson Natural Language Processing (NLP):** It is an NLP-driven framework designed for sentiment and tone analysis, which makes it effective for tone evaluation. Unlike traditional rule-based sentiment analysis tools, Watson leverages Natural Language Processing to understand emotion, sentiment, and tone across various contexts. A unique feature of this framework is its ability

Table 7. **Evaluation Metrics:** Comparison of evaluation metrics across frameworks and datasets.

Traditional NLP/Transformer Methods					
Framework	**Dataset**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
VADER	Hugging Face	0.61	0.63	0.61	0.59
	Manual Dataset	0.60	0.58	0.60	0.58
IBM Watson	Manual Dataset	0.40	0.41	0.40	0.36
RoBERTa	Hugging Face	0.72	0.72	0.73	0.71
	Manual Dataset	0.78	0.72	0.70	0.68
LLM-as-a-Judge Evaluation					
Claude (Overall Tone)	Hugging Face	0.65	0.65	0.65	0.64
	Manual Dataset	0.82	0.83	0.82	0.78
Claude (Specific Tone)	Manual Dataset	0.88	0.90	0.88	0.87

to detect complex tones such as excited, frustrated, impolite, polite, sad, satisfied, and sympathetic (IBM Cloud Docs, 2024).

- **RoBERTa Transformer Model:** RoBERTa is an updated and improved version of the BERT sentiment analysis model. This model, built on the basis of the Hugging Face Transformers library, utilizes deep learning to properly identify the sentiment of text classification between three label groups (positive, negative, and neutral).

3.7.3 Method

Anthropic Claude Sonnet 3.5 Model was prompted using two zero-shot prompt formats. For specific tone classification to compare against IBM Watson, the prompt was: *"What is the tone of this text? Please choose only one tone from the following options: excited, frustrated, sympathetic, polite, impolite, satisfied, or sad. Respond only with the tone (e.g., 'satisfied')."* For general sentiment classification, utilized by Hugging Face and manually curated dataset, to compare against VADER and RoBERTa, the prompt used was: *"What is the tone of this text? Please choose only one tone from the following options: negative, neutral, or positive. Respond only with the tone (e.g., 'neutral')."* Claude's outputs were retrieved through the Anthropic Python API and stored in designated columns.

IBM Watson's tone predictions were generated using its Natural Language Understanding (NLU) API. Exception handling was implemented to address errors such as API rate limits and unsupported text formats. The VADER model was applied locally to compute compound sentiment scores for each text entry, which were then mapped to categorical sentiment labels. RoBERTa was executed in Google Colab using the Hugging Face Transformers library, and a pre-trained model was applied to the same dataset to yield sentiment predictions. Due to the lack of intricate tone (only positive, negative, and neutral labels provided) in the Hugging Face dataset, it was only applied to VADER and RoBERTa models.

All model predictions were consolidated into a shared dataset and evaluated against the ground truth. Metrics including accuracy, precision, recall, and F1 score were computed using sklearn.metrics to assess each model's overall performance. This enabled a uniform comparison across all frameworks.

3.7.4 Results

The results are presented in Table 7 and Figure 4.

Anthropic's LLM, Claude, achieved the highest performance in specific tone classification with an accuracy of 88%, outperforming IBM Watson's NLP, which had an accuracy of 40%. In the general tone task, Claude also led with accuracy 82%, surpassing RoBERTa and VADER, which scored 78% and 60%, respectively. Across all key evaluation metrics, precision, recall and F1 score, Claude consistently outperformed the lexicon-based models (VADER and IBM Watson). Although RoBERTa trailed Claude, it showed strong overall performance, with a margin of only 9–16% in most metrics.

Claude achieved 82% accuracy on the basic tone classification on the manually curated dataset but dropped to 65% accuracy on the Hugging Face dataset, while RoBERTa outperformed Claude on the Hugging Face dataset with an accuracy of 72%. This difference could be due to the fact that RoBERTa is specifically fine-tuned for sentiment analysis tasks, whereas Claude is a general-purpose LLM trained on a broad corpus of text without targeted optimization for sentiment classification. Additionally, since API calls for over 5,000 rows of text were made, proper API key management and sufficient computational resources were necessary to ensure stable results. Using the most up-to-date Claude

models could potentially improve performance on larger or more standardized datasets. However, despite these challenges, Claude still achieved the highest accuracy overall and shows strong potential for use in real-world sentiment and tone analysis tasks.

These results suggest that transformer-based models, particularly LLMs like Claude, are better equipped to capture nuanced contextual information than NLP or lexicon-based systems. Although task-specific fine-tuning, as seen with RoBERTa, can provide performance advantages on standardized datasets, LLMs generally offer broader flexibility and strong baseline performance without requiring extensive retraining. Overall, the current findings reinforce the idea of leveraging LLMs like Claude for scalable, real-world sentiment and tone analysis applications.

3.8 Readability

LLM Readability refers to the reading quality of LLM output text considering multiple qualities, including syntax, lexical difficulty, grammar, and lexical diversity.

3.8.1 Datasets

The dataset chosen is the CommonLit Ease of Readability Corpus (CLEAR) corpus. This dataset was developed by CommonLit, a literacy education nonprofit, and the Georgia State University [37]. It contains 4724 reading passages of varying reading difficulty. For the purpose of readability assessment, these literature passages will be considered a suitable representation of textual LLM output. It is annotated with columns containing information about the passage (author, title, anthology, etc), and common readability scores. Each of these scores can be thought of as evaluation frameworks that utilize methods proposed in readability research (as a subset of educational, linguistic, and computational linguistic research), and were machine-scored by ARTE (Automatic Readability Tool for English). Namely, there are the Flesch Reading Ease score, New Dale-Chall, and CAREC scores. The final note to mention with the CLEAR dataset is that upon its launch, CommonLit ran a Kaggle competition to find the most effective readability assessment the public could design, and the best performing solutions had their predictions annotated as additional scores.

3.8.2 Frameworks

This study will create a novel readability scoring system as a first framework, and utilize LLM-as-a-judge methodology by directly prompting Claude Haiku 3.5 as a second framework.

3.8.3 Method

Though some of the readability scores annotated in the CLEAR dataset can serve as effective frameworks for evaluating readability (namely modern NLP-based scoring systems like CAREC and CML2RI), there is no known tailored framework that exists for evaluating the readability of LLM outputs. Many of the more recognized annotated scoring systems (e.g. Flesch, New Dale-Chall) are more applicable to educational contexts, and each system is defined under a different set of rules [38]. For example, Flesch considers a ratio of number of words to number of sentences in a text, as well as a ratio of total syllables to total words - in essence it equates readability as syntax. New Dale-Chall factors in word difficulty by referencing a list of 3000 "easy" words - if a word is not on that list, it counts as a penalty against the score. This is to say that from text to text, readability can vary greatly depending on what score is used. A piece of 1st-grade level text considered "simple" or "unchallenging" to read by some scores is not necessarily readable in the context of LLM text outputs; it may not be engaging to read or have a diverse vocabulary. Therefore, these reasons encourage the creation of a novel readability evaluation framework designed specifically for LLM text output.

Framework 1 is thus an original approach to calculating readability as a combination of syntax, lexical difficulty (AKA word/vocab difficulty), grammar, and lexical diversity. This syntactic component is computed by using the Flesch Reading Ease score, itself an established readability score focusing on syntactic structure. Lexical difficulty is assessed by counting the appearance of words in a testing corpus. This corpus of text is sourced from the `nltk` Python library, and it contains informal and formal English language text examples from movie reviews, news articles, prolific literature, web chats, and other sources. `nltk` pre-counts the frequency of each word in the corpus. With the excerpts in the dataset, the text is first tokenized to extract valid words (no stop-words or proper nouns), and each word is then lemmatized to its base form. These lower-cased and lemmatized words are then indexed into the corpus to count their frequency, so that a word with higher frequency is likely an "easier" word and a word seen less often is likely a "harder" word. The average word frequency is computed among each word in the excerpt to return the lexical difficulty score. Grammar will be assessed via LanguageTool's `language_tool_python` grammar-checking library, calculating number of errors per word. Lexical diversity is assessed using the `lexicalrichness` library, which calculates the Measure of Textual Lexical Diversity (MTLD) score. That number is the mean length of word strings in the excerpt that have reached a certain threshold of diversity. Together, these four component scores (Syntax Score, Lexical Difficulty Score, Grammar Score, and Lexical Diversity Score) will be scaled from 0 - 100, and then averaged to a number also between 0 - 100 to create a Novel Readability scoring framework.

Framework 2 utilizes the LLM-as-a-judge methodology. Claude 3.5 Haiku is prompted to score each excerpt on a scale of 0 - 100, based on its own definition of readability. It is also

Table 8. **Readability:** Results of Readability evaluation using different frameworks on the CLEAR dataset.

Evaluation	**Novel Readability**	**Claude Readability**	Syntax	Lexical Difficulty	Grammar	Lexical Diversity
Mean Abs Error (% Error)	9.892	10.582	11.221	9.506	28.266	26.408
Accuracy ($< 10\%$ Error)	58.59%	55.14%	49.79%	60.75%	19.71%	16.77%

given few-shot examples, by providing the lowest-scored and highest-scored excerpt from the Novel Readability scoring framework, along with their respective Novel Readability scores. Exactly, the prompt is as follows per each excerpt, dependent on the Novel Readability min/max:

On a readability scale from 0.0 to 100.0:
{bad_score} represents the least readable text like this: "{bad_score_example}"
{good_score} represents the most readable like this: "{good_score_example}"
Based on this scale, output only a single number between 0.0 and 100.0 that represents the readability score for this text: {excerpt}
Just provide the number without any explanation.

3.8.4 Results

The results are presented in Table 8. In bold are the Novel Readability and Claude Readability columns, and additionally, the Novel Readability composite scores (Syntax, Lexical Difficulty, Grammar, and Lexical Diversity). Though it appears that Lexical Difficulty has by a small margin the least error and highest accuracy, this can be explained by the human-written reading passages. Professional authors and LLM agents construct sentences in different ways. Lexical Difficulty may be a good predictor of readability for the human-written reading passages because lexical difficulty could be naturally correlated with syntax and grammar, in that a person who has an advanced vocabulary tends to write in a more long-winded way, potentially with a number of grammatical errors. Again, with the dataset and available scoring frameworks on market, there is a struggle to capture the true nature of LLM agent readability. This is why the Novel Readability score as a whole presents a more robust option than any of its composite scores, because it evenly evaluates multiple features of readability, while offering strong accuracy and low error. It even outperforms Claude by a small margin, while being interpretable.

This is to say that Novel Readability and Claude are both viable options. Claude querying takes significantly less time than computing the grammar score component of Novel Readability (though all other component scores compute nearly instantaneously). However, these performance metrics are calculated in comparison to a subjective ground truth, the CAREC score. There are numerous Python libraries, as well as an online tool and API endpoint offered at Georgia State University's ARTE program that are capable of computing the CAREC score for a body of text. It is a viable readability evaluation framework in itself, despite inherently lacking a tailored design for evaluating LLM output. The recommendation of this study for the strongest readability evaluation framework would be, in order, the Novel Readability score, LLM-as-a-Judge, and CAREC.

There remains much future work in assessing LLM readability. Unlike other dimensions of LLM quality, where the straightforward goal is to reduce hallucination or improve accuracy, readability is a dimension that requires a level of subjectivity. There needs to be research on what readability qualities users tend to value for textual LLM output. If a prompt sought more clarity on a topic, the user would value comprehensive detail and would be more forgiving towards a more difficult syntax and lexicon. If a prompt sought a quick answer, the user would value a very easy syntax and lexicon. Thus, there is no current one-size-fits-all solution to evaluating readability and no single rigid scoring system that would have long term utility. The historical readability research landscape for generic/educational contexts reflects this understanding, where published readability scores gradually modernized by considering more and more readability features, and employing flexible NLP techniques. Future readability scores developed in this way are sure to be able to consider more and more minute features of readability and will probably be even better candidates to use as LLM evaluation frameworks (perhaps even outpacing LLM-as-a-Judge), but they need to be designed explicitly for LLMs in order to orient their utility for LLMs. The difference between the educational and LLM context is that the LLM context serves a user base with preferences. Therefore, future research for the LLM readability niche need to consider user values for readability, and the score needs to be able to dynamically adapt to evaluate the needs of each prompt.

4 Conclusion

Our evaluation highlights that no single framework universally outperforms others across all tasks. Instead, effectiveness is highly task-dependent. For example, while RAG + MLflow performs best on retrieval accuracy in the SQuAD

dataset, DeepEval consistently provides strong evaluations across multiple tasks, including answer relevancy, toxicity, summarization, and bias detection. Additionally, transformer-based models like Claude demonstrate clear advantages in tasks requiring tone and sentiment analysis, and LLM-as-a-Judge metrics generally outperform traditional NLP metrics when evaluating complex behaviors such as hallucinations and bias.

These findings suggest an emerging pattern: LLM-as-a-Judge frameworks excel when evaluation requires nuanced, context-aware judgment, while traditional NLP metrics remain useful for more objective, surface-level assessments. This points to the importance of hybrid evaluation strategies that leverage the strengths of both approaches.

Our study, spanning over 10 datasets and eight evaluation dimensions, demonstrates the practical importance of evaluation design for real-world scenarios—from hospitals deploying diagnostic assistants to financial institutions monitoring risk. As LLMs grow in complexity, adaptive and interpretable evaluation methodologies will be critical.

Looking forward, as LLMs continue to evolve in complexity and capability, future research should focus on three key directions: (1) extending evaluations to additional frameworks and LLMs beyond Claude, (2) expanding to multimodal evaluations covering audio, video, and other non-text outputs, and (3) developing domain-specific evaluations to determine which frameworks excel in specialized areas like mathematics, coding, or reading comprehension. Ensuring responsible AI deployment will require not only choosing the right tools for today's tasks but also designing evaluation frameworks that remain robust, scalable, and fair as LLM technology advances.

5 Acknowledgments

The authors would like to acknowledge and thank the following individuals for their guidance and invaluable input:

- Ashwin Admala, Deloitte
- Neha Brahmabhatt, Deloitte
- Sarah Burinsky, Deloitte
- Charlie Evert, Deloitte
- Maria Kipreos, Deloitte
- Brendan McElron, Deloitte
- Chase Oden, Deloitte
- Miriam White, Deloitte

6 References

[1] A. Bavaresco, R. Bernardi, L. Bertolazzi, D. Elliott, R. Fernández, A. Gatt, E. Ghaleb, M. Giulianelli, M. Hanna, A. Koller *et al.*, "Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks," *arXiv preprint arXiv:2406.18403*, 2024.

[2] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 2383–2392. [Online]. Available: https://aclanthology.org/D16-1264/

[3] M. Maia, D. Jurgens, A. Hürriyetoğlu, M. Beloucif, A. M. Moreno, and J. Steinberger, "Overview of the fiqa 2018 shared task: Fine-grained opinion retrieval from financial microblogs and news," in *Proceedings of the First Workshop on Financial Technology and Natural Language Processing (FinNLP)*, 2018, pp. 1–6. [Online]. Available: https://ceur-ws.org/Vol-2156/FiQA_2018_paper_6.pdf

[4] Amazon Web Services, "What is rag? - retrieval-augmented generation ai explained," https://aws.amazon.com/what-is/retrieval-augmented-generation/, 2023, accessed: 2025-04-23.

[5] MLflow Contributors, "Mlflow: An open source platform for the machine learning lifecycle," https://mlflow.org/, 2023, accessed: 2025-04-23.

[6] J. Ip and K. Vongthongsri, "Deepeval: The open-source llm evaluation framework," 2025, accessed: 2025-04-22. [Online]. Available: https://www.deepeval.com/

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/about.html

[8] Z. Wang, Y. Dong, J. Zeng, V. Adams, M. N. Sreedhar, D. Egert, O. Delalleau, J. P. Scowcroft, N. Kant, A. Swope, and O. Kuchaiev, "Helpsteer: Multi-attribute helpfulness dataset for steerlm," 2023.

[9] C. AI, "Toxicity metric," 2025, accessed: 2025-04-22. [Online]. Available: https://www.deepeval.com/docs/metrics-toxicity

[10] Google, "Jigsaw toxic comment classification," https://huggingface.co/datasets/google/jigsaw_toxicity_pred, accessed: 2025-04-22.

[11] H. Face, "Distilbert," https://huggingface.co/docs/transformers/en/model_doc/distilbert, accessed: 2025-04-22.

[12] OpenAI, "Chatgpt," https://chat.openai.com/chat, 2025, april 22 version.

[13] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang, "Gender bias in coreference resolution: Evaluation and debiasing methods," *arXiv preprint arXiv:1804.06876*, 2018.

[14] N. Nangia, C. Vania, R. Siddhant, and S. R. Bowman, "Crows-pairs: A challenge dataset for measuring social biases in masked language models," in *Findings of the Association for Computational Linguistics: EACL 2024*, 2024, pp. 710–726. [Online]. Available: https://aclanthology.org/2024.findings-eacl.61/

[15] Y. Wang, X. Han, H. Li, P. Nakov, and T. Baldwin, "Do-not-answer: A dataset for evaluating safeguards in llms," in *Findings of the Association for Computational Linguistics: EACL 2024*, 2024, pp. 1714–1730. [Online]. Available: https://aclanthology.org/2024.findings-eacl.138/

[16] E. Fast, B. Chen, and M. S. Bernstein, "Empath: Understanding topic signals in large-scale text," *arXiv preprint arXiv:1602.06979*, 2016.

[17] I. O. Gallegos, R. A. Rossi, J. Barrow, M. M. Tanjim, S. Kim, F. Dernoncourt, T. Yu, R. Zhang, and N. K. Ahmed, "Bias and fairness in large language models: A survey," *Computational Linguistics*, vol. 50, no. 3, pp. 1097–1179, 2024.

[18] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *arXiv:2311.05232*, 2024.

[19] S. Banerjee, "Llms will always hallucinate, and we need to live with this," *arXiv:2409.05746*, 2024.

[20] RUCAIBox, "Halueval: Evaluation toolkit for hallucination in llms," https://github.com/RUCAIBox/HaluEval, 2023.

[21] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," *arXiv:1809.09600*, 2018.

[22] J. Li, X. Cheng, W. X. Zhao, J.-Y. Nie, and J.-R. Wen, "Halueval: A large-scale hallucination evaluation benchmark for large language models," *arXiv:2305.11747*, 2023.

[23] Arize AI, "Phoenix user guide," https://docs.arize.com/phoenix/user-guide.

[24] Confident AI, "Deepeval hallucination," https://www.deepeval.com/docs/metrics-hallucination.

[25] ——, "Deepeval: G-eval," https://www.deepeval.com/docs/metrics-llm-evals.

[26] Exploding Gradients, "Faithfulness metric," https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/faithfulness/.

[27] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-eval: Nlg evaluation using gpt-4 with better human alignment," *arXiv:2303.16634*, 2023.

[28] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 1906–1919.

[29] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv:1904.09675*, 2020.

[30] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "Summeval: Re-evaluating summarization evaluation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 391–409, 2021.

[31] W. Kryściński, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "Neural text summarization: A critical evaluation," *arXiv preprint arXiv:1908.08960*, 2019.

[32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[33] S. Banerjee and A. Lavie, "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*, 2005, pp. 65–72.

[34] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[35] J. Ip and K. Vongthongsri, "deepeval," https://confident-ai.com, 2025, version 2.7.9. The Open-Source LLM Evaluation Framework. Available at https://github.com/confident-ai/deepeval.

[36] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-eval: Nlg evaluation using gpt-4 with better human alignment," *arXiv preprint arXiv:2303.16634*, 2023.

[37] S. Crossley, A. Heintz, J. Choi, J. Bachelor, M. Karimi, and A. Malatinszky, "The commonlit ease of readability (clear) corpus," *Educational Data Mining 2021*, 2021.

[38] S. Crossley, S. Skalicky, and M. Dascalu, "Moving beyond classic readability formulas: New methods and new models," *Journal of Research in Reading*, 2019.