SMILE: A Composite Lexical-Semantic Metric for Question-Answering Evaluation

Anonymous ACL submission

Abstract

Traditional evaluation metrics for textual and visual question answering, like ROUGE, METEOR, and Exact Match (EM), focus heavily on n-gram based lexical similarity, often missing the deeper semantic understanding needed for accurate assessment. While measures like BERTScore and MoverScore leverage contextual embeddings to address this limitation, they lack flexibility in balancing sentence-level and keyword-level semantics and ignore lexical similarity, which remains important. Large Language Model (LLM) based evaluators, though powerful, come with drawbacks like high costs, bias, inconsistency, and hallucinations. To address these issues, we introduce **SMILE**: Semantic Metric Integrating Lexical Exactness, a novel approach that combines sentence-level semantic understanding with keyword-level semantic understanding and easy keyword matching. This composite method balances lexical precision and semantic relevance, offering a comprehensive evaluation. Extensive benchmarks across text, image, and video QA tasks show SMILE is highly correlated with human judgments and computationally lightweight, bridging the gap between lexical and semantic evaluation.

1 Introduction

Question answering (QA) is an essential task used to measure the progress of language-based models. Across text, image, and video domains, the primary measure of model performance on QA benchmarks is accuracy, which is typically computed via *exact (or easy) match (EM)*: A model response is deemed correct if the ground-truth answer, typically annotated by humans, exactly matches (or can be found within) the model response. As recent models have grown to be more capable language generators, model answers have grown more nuanced, making EM overly stringent (Wang et al., 2023a). A reasonable attempt to mitigate these issues is to employ N-gram based metrics typically used for text generation evaluation, such as ROUGE (Lin, 2004) or METEOR (Banerjee and Lavie, 2005), or embedding-based metrics like BERTScore (Zhang et al., 2019), to assess similarity between the predicted response and the ground-truth (Rajpurkar et al., 2016; Bajaj et al., 2016; Dunn et al., 2017; Kočiskỳ et al., 2018; Yang et al., 2018a). While such metrics capture high-level similarity between the model response and ground-truth, they may miss fine-grained details crucial to answer correctness (e.g., "The cat is *on* a chair" vs. "The cat is *under* a chair") that result in lower correlation with human judgments (Mañas et al., 2024).

Concurrently, due to their strong language comprehension abilities, large language models (LLMs) have been deployed as automatic evaluators for text generation. This approach, broadly known as *LLM-as-judge*, functions by either prompting more capable LLMs, like GPT-4o, or finetuning smaller LLMs specifically for evaluation. LLM-as-judge is appealing as LLMs can adapt to different evaluation criteria and generate explanations. Consequently, recent methods and benchmarks (Jacovi et al., 2025; Wang et al., 2024a) now employ judges as evaluators in QA settings.

However, using judge models for evaluation increases *costs*. For practitioners and developers with limited resources, repeatedly querying pay-per-use API models to evaluate large datasets (5K+ samples) or dedicating limited compute to hosting an evaluation server can be impractical for rapid development, which may drive them to use lesser but faster metrics. Beyond resource demands, generative evaluators also exhibit relatively high latency (see Figure 1) and are susceptible to hallucinations, as we qualitatively show in Section 3.

This work revisits embedding-based approaches for automatic QA evaluation and introduces Semantic Metric Integrating Lexical Exactness (SMILE), a lightweight yet high-performing framework for

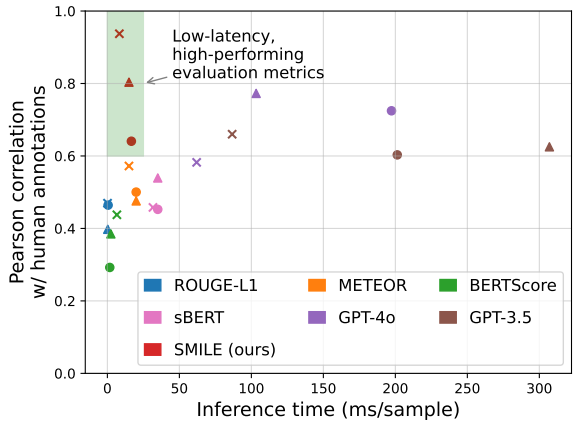

Figure 1: **SMILE offers high-performance, low-latency QA evaluation**, breaking the trade-off between cost and performance. Performance is averaged across human-annotated samples from benchmarks for natural language (×), image (△), and video (○) domains.

grading QA tasks. SMILE aims to retain the efficiency of embedding-based evaluators while addressing their limitations, such as lack of fine-grained response understanding. To do so, SMILE comprises two subscores: A semantic subscore to assess overall response content, and a keyword subscore to reward lexical exactness. Overall, SMILE offers a best-of-both-worlds evaluation solution: As Figure 1 shows, it correlates with human annotators as strongly as GPT-4o. Additionally, SMILE core components can be precomputed for fast lookup, resulting in a 9x speedup compared to API queries. SMILE's lightweight design allows it to run on CPU during evaluation, requiring minimal GPU VRAM to perform a *one time* evaluation dataset preprocessing step. Additionally, SMILE offers interpretability through its composite structure with two subscores. While SMILE can be applied in many settings, our study focuses on factoid QA tasks. Our contributions are:

(1) We revisit the promise of embedding-based automatic evaluation metrics for QA tasks and propose SMILE, which utilizes both sentence and keyword level similarity scores to evaluate based on holistic and fine-grained content.

(2) We construct a 225-sample human annotated test set from nine (text, image, video) QA datasets, labeled by 4 domain experts. This set is used to benchmark SMILE and other metrics based on their correlation with human judgments.

(3) We demonstrate that SMILE can serve as a lightweight drop-in replacement for more powerful LLM-as-judge models across modalities.

(4) Extensive ablation studies demonstrate the necessity of each of SMILE's components.

In all, our experiments demonstrate that SMILE is a *lightweight* yet *high-performing* automatic evaluation metric for QA settings.

2 Background and related work

This work addresses three QA modalities: Natural language QA (NLQA), visual QA (VQA), and video QA (VidQA). Across these, a model f receives an input (q, c) and generates a textual answer $y = f(q; c)$. The input consists of a question q and context c, where c is text for NLQA, an image for VQA, and a video for VidQA. The task of the model is to produce a natural language answer y to the question q based on the given context c.

For model evaluation, we adopt a *reference-based* protocol, assuming a human-annotated ground-truth answer $y^\star$ is available for each input (q, c) . Given a model response y, the goal is to determine its correctness. Specifically, we aim to design an evaluator j that produces an evaluation score $s = j(y, y^\star)$ based on y and $y^\star$. This *source-free* setup, where evaluation occurs without access to (q, c), aligns with the original exact match (EM) evaluation setup. Source-free evaluation may represent an *easier* evaluation setting, as prior work indicates LLM-based evaluators struggle when context c is included (Xu et al., 2025).

For practitioners prioritizing accuracy, the evaluation score s can be a binary correct/incorrect label. For detailed failure analysis, a finer-grained score (e.g. 0-5 scale) may be preferred. Regardless of the format, the score should be convertible to a binary label, typically via a straightforward threshold (e.g., scores ≤ 3 as incorrect, scores ≥ 4 as correct for a 0-5 scale) (Maaz et al., 2024; Wang et al., 2024a). We now review existing metrics and evaluators.

Text generation metrics. QA benchmarks have typically used EM or ROUGE (Lin, 2004) to assess model outputs, e.g., (Yang et al., 2018b; Rajpurkar et al., 2016; Dunn et al., 2017; Fan et al., 2019). As model responses grew more nuanced, n-gram metrics such as BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) were adopted in QA settings (Bajaj et al., 2016). Despite better correlation with human annotations than EM, n-gram metrics have been shown to be insufficient for modern QA tasks (Chen et al., 2019).

Embedding-based metrics. Embedding models, like BERT (Devlin et al., 2019) and finetuned

variants, e.g., (Reimers and Gurevych, 2019; Gao et al., 2021), are trained to measure semantic similarity. As such, embedding-based metrics are a natural step in overcoming the limitations of EM and n-gram-based metrics (Chen et al., 2019), with notable methods like BERTScore (Zhang et al., 2019), BARTScore (Yuan et al., 2021), BLEURT (Sellam et al., 2020) being used as evaluators in benchmarks (Ao et al., 2024). Recent work (Bulian et al., 2022; Risch et al., 2021; Lee et al., 2020) developed metrics specifically for the QA setting.

LLM-as-judge evaluators. The LLM-as-judge paradigm initially utilized frontier LLMs for automatic evaluation (Wang et al., 2023b; Liu et al., 2023c; Fu et al., 2024; Chiang and Lee, 2023). However, biases in prompted evaluators hold were soon identified (Panickssery et al., 2024; Wang et al., 2023c; Park et al., 2024), leading to the finetuning of smaller models for evaluation (Kim et al., 2024; Li et al., 2023a; Zheng et al., 2024; Wang et al., 2023d; Shiwen et al., 2024). Recent efforts focus on training for diverse evaluation protocols (Vu et al., 2024; Wang et al., 2024b), such as pairwise, single rating, and binary classification. Applying LLM-as-judges specifically to QA grading is a recent development (Mañas et al., 2024), with many benchmarks (Krishna et al., 2024; Jacovi et al., 2025) and studies (Maaz et al., 2024; Liu et al., 2023b) employing API models.

3 When do existing evaluation methods and metrics fail?

Before introducing SMILE, we present a qualitative case study highlighting the failure modes of LLM-as-judge and embedding-based metrics in QA evaluation. Our analysis of 225 human-evaluated model responses reveal a common failure point: verbose or generic model outputs, consistent with Mañas et al. (2024); Luo et al. (2021).

Surprisingly, we also find that using a powerful LLM like GPT-4o does not guarantee accurate evaluations. We highlight two representative failure modes in Figure 2 (columns 1-2). GPT-4o was prompted to generate a binary accuracy label, as well a 1-5 score (see Appendix A for prompt). For ease of comparison, we convert SMILE scores to a binary accuracy indicator. A primary concern with using LLMs as judges is hallucinations. Figure 2 (column 1) shows that even for relatively simple samples, GPT-4o may hallucinate an incorrect label. Figure 2 (column 2) is an example of *concreteness* bias, a known judge model bias (Park et al., 2024). Here, GPT-4o is tricked by a response that includes concrete artifacts, like the table the model generated, even if it incorrect response.

Seeking an efficient evaluator, we also analyzed failure modes of embedding-based approaches. Semantic similarity metrics like BERTScore exhibited well-known limitations (Zhang et al., 2019). Figure 2 (columns 3-4) highlights two key examples: column 3 shows how overly verbose model responses can easily misled semantic similarity metrics, as much of the output is irrelevant to the simple "yes" response. This can be viewed this as *distributional misalignment* (Agrawal et al., 2022): increasingly high-quality model outputs are often lengthier, contrasting with the typical short answers in factoid QA benchmarks. Conversely, when model responses are short but semantically relevant, these metrics are prone to false positives, as illustrated in Figure 2 (column 4). The limitations of embedding-based and LLM-as-judge methods motivated SMILE-a hallucination-free evaluation metric presented next.

4 The SMILE metric

Our analysis in Section 3 pinpointed two critical limitations of embedding-based approaches: (1) a *distributional gap* between verbose model responses and concise ground-truth answers, and (2) a *lack of fine-grained understanding* due to their semantic focus. SMILE directly addresses these issues with two key innovations: (1) Synthetic answer generation to bridge the stylistic distribution gap, and (2) targeted sub-scores capturing both semantic and lexical similarity between model responses and ground-truth.

Bridging the stylistic distribution gap. As shown in Section 3, assessing directly based on the ground truth $y^\star$, which is typically short for short-form (factoid) QA (e.g., a single word or short phrase), may be sub-optimal, as model responses tend to be more verbose. Motivated by past work that have used LLMs to perform other kinds of zero-shot distribution alignment (Gao et al., 2023; Xu et al., 2024), we utilize an LLM to generate a synthetic model response from the ground-truth. Our key insight is that for short-form QA tasks, a *lightweight* model (e.g., 3B parameter) can be deployed as a synthetic answer generator g. Specifically, the generator g takes as input the original question q and ground truth answer $y^\star$ and out-

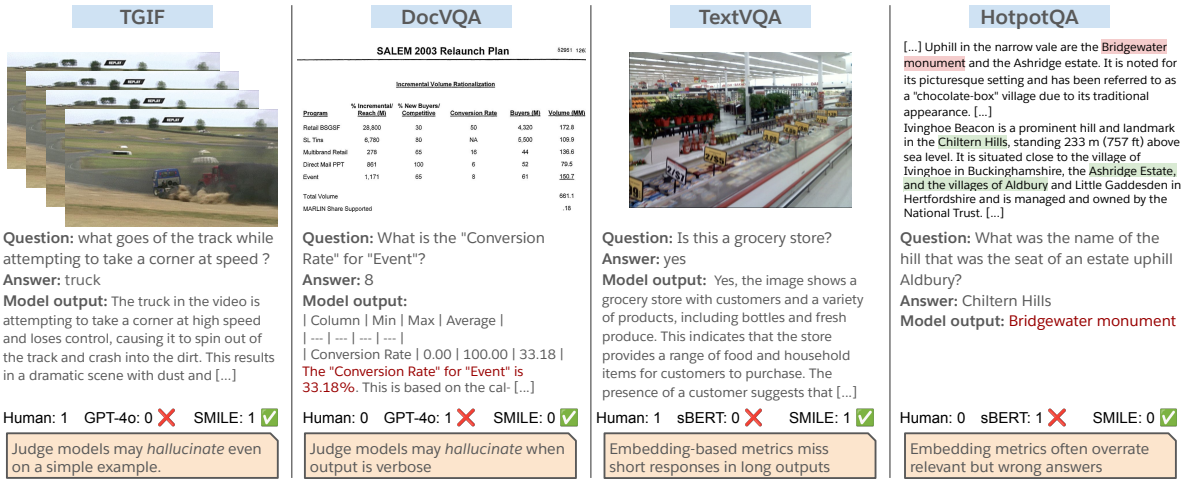

Figure 2: **Example failure cases for existing methods**. *Columns 1-2* illustrate LLM-as-judge failures: hallucination even on simple verification (column 1), and incorrect yet concrete responses (column 2). *Columns 3-4* illustrate embedding-based model failures with lengthy (column 3) and relevant but incorrect (column 4) responses. SMILE scores are converted to a 0/1 scale for comparison. [...] denotes omitted content to fit space.

puts a synthetic answer $\tilde{y} = g(y^\star, q)$, which aligns stylistically with model responses, but reflects the ground-truth answer content. As a concrete example from our evaluation setup, for input question "What is the Conversion Rate for Event?" and ground-truth "8", a generated synthetic answer is "The conversion rate of an event is 8". We emphasize that *synthetic answer generation is independent of the model being evaluated and is performed only once, prior to test-time, per evaluation set*. As a result, synthetic answers may be stored and used for any subsequent evaluations.

Integrating semantic and lexical similarity. The core idea of SMILE is to measure both *semantic* and *lexical* similarity between the model response and the ground-truth using an embedding model e. We calculate a semantic similarity score, which we denote s_s, as

$$s_s(y, \tilde{y}; e) = \texttt{sim}(e(y), e(\tilde{y})), \quad (1)$$

where, $\texttt{sim}(x, y) = (1 + \langle x, y\rangle / \|x\|_2 \|y\|_2)/2$, which is a linearly transformed cosine similarity that lies within an interpretable interval of $[0, 1]$. As we show in Section 5.2, generating synthetic answers bridges the stylistic distribution gap between ground-truth answers and model responses enough to make semantic similarity meaningful. However, Section 3 shows that this semantic similarity score alone is insufficient to capture the nuances of evaluation. As a result, we additionally compute a lexical similarity score, which we denote $s_\ell \in [0, 1]$, as

$$s_\ell(y, y^\star; e) = \frac{1}{2}(\text{EM}(y, y^\star) + \max_i \{\texttt{sim}\,(e(N_i[y]), e(y^\star))\}) \quad (2)$$

where $EM(y, y^\star) \in \{0, 1\}$ score between the prediction y and ground-truth $y^\star$ and $N_i[y]$ denotes the i-th n-gram of response y. In computing s_ℓ, we take advantage of the fact that $y^\star$ is typically a short phrase to compute two complementary scores. The easy match sub-score EM serves as a preliminary check for lexical answer correctness. However, as noted in prior work (Wang et al., 2023a; Luo et al., 2021), string matching may be too stringent for synonym-like answers (e.g., "cat" vs. "kitten"). As a result, we loosen the necessity for string matches via the maximum n-gram embedding similarity score, which serves as a continuous-valued measure of lexical exactness.

Evaluation with SMILE. With our semantic and lexical scores computed, we can now compute the SMILE score, denoted $s_{\texttt{SMILE}} \in [0, 1]$:

$$s_{\texttt{SMILE}}(y, y^\star; e, w) = \frac{1}{2}(w \cdot s_s(y, \tilde{y}; e) + (1 - w) \cdot s_\ell(y, y^\star; e)), \quad (3)$$

where $w \in (0, 1)$ is some user-specified weight to balance the two subscores. This weighting mechanism allows practitioners to express their preferences: Those who are more inclined towards exact match may place higher weight on s_ℓ, whereas those who value higher responses whose meaning is closest with the ground-truth may place a higher weight on s_s.

4.1 Optimizations for test-time speed-up

SMILE offers significant speed advantage over LLM-as-judge methods, as extracting represen-

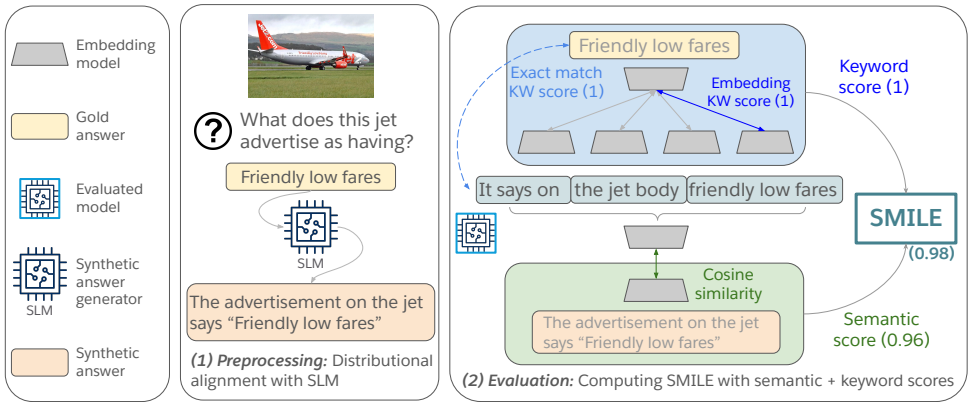

Figure 3: **Overview of SMILE**. SMILE evaluates QA outputs in two steps using a synthetic answer generator and an embedding model. (1) *One-time* preprocessing generates stylistically aligned synthetic answers using a small language model (SLM). (2) SMILE computes two sub-scores: a semantic score based on the synthetic answer, and a keyword subscore that combines exact match with embedding comparisons of model response N-grams to the gold answer. This subscoring balances semantic and lexical evaluation.

tations from lightweight embedding models like BERT (Devlin et al., 2019) is far faster than generating natural language outputs. This speed advantage can be further enhanced by *pre-computing* and storing representations for synthetic answers $e(\tilde{y})$ and keyword representations $e(y^\star)$ before evaluation. By storing $e(\tilde{y})$ and $e(y^\star)$, only the model response representations $e(y)$ and $e(N_i[y])$ need to be calculated during test time.

4.2 Interpretability of SMILE scores

SMILE's semantic and lexical subscores provide practitioners with more interpretable and actionable feedback than other metrics. These subscores enable monitoring of model performance along two complementary axes: semantic content, a holistic measure of response relevance, and lexical exactness, a finer-grained measure of response quality. Importantly, SMILE allows evaluation not only at the instance-level but also at the *population-level*. Aggregating s_s and s_ℓ across all test samples reveals a model's *general* strengths and weaknesses. This contrasts with LLM-as-judge methods, which offer more specific *instance-level* natural language feedback, making it hard to extract overall insights. See examples in Appendix E.

5 Experiments and results

Benchmarks and generator models. We assessed SMILE on established benchmarks across three domains: NLQA, VQA, and VidQA. To ensure diverse evaluation, we included three benchmarks per domain: MRQA (Fisch et al., 2019), HotpotQA (Yang et al., 2018b), and MuSiQue (Trivedi et al., 2022) for NLQA, TextVQA (Singh et al., 2019), DocVQA (Mathew et al., 2020), and POPE (Li et al., 2023b) for VQA, and TGIF (Jang et al., 2017), MSVD (Xu et al., 2017), and MSRVTT (Xu et al., 2016) for VidQA. For HotpotQA and MuSiQue, we used the standardized setup from ContextualBench (Nguyen et al., 2024). We generated responses using the following models for each domain: GPT-4o (Hurst et al., 2024) for NLQA, LLaVA-1.5 for VQA 7B (Liu et al., 2023a,b) , and Qwen2.5-VL 3B Instruct (Bai et al., 2025) for VidQA. These models were selected for their strong capabilities in producing high-quality textual responses, forming the basis of our analysis. We also evaluate on QA-Eval (Wang et al., 2023a), a large-scale NLQA dataset (∼10k samples) based on Natural Question (NQ) and TriviaQA (TQ), with responses from GPT-3.5 and GPT-4o. This setup enables robust comparison of SMILE against LLM judges at scale.

Data annotation efforts. To evaluate QA metrics, we assessed their alignment with human judgments using a *golden evaluation set*. We constructed this set by sampling model outputs from the nine benchmarks (three per domain), randomly selecting 25 input-output pairs per dataset for annotation. Four annotators (authors of the paper with native level English) evaluated the generated outputs based on a predefined rubric, considering correctness, relevance, and clarity. Given potential ambiguity, annotators used a 3 point scale: clearly incorrect, unclear, clearly correct. To check annotation quality, we calculated Krippendorff's alpha (Krippendorff, 2011), achieving a score of **0.71**,

indicating substantial inter-annotator agreement. This high agreement confirms the reliability of our annotations, so we proceed with it as the basis of our evaluation.

Baselines and metrics. We compared SMILE with established metrics, including traditional NLP measures: ROUGE-L, METEOR and Exact and Easy Match; alongside embedding-based similarity metrics: BERTScore (with Roberta-large) and sBERT cosine similarity[1]. Following (Maaz et al., 2024), we also employed GPT-4o and GPT-3.5-Turbo as judge models, prompting them for a 0-5 score and a binary yes/no prediction. For all baselines, we provide detailed implementation details in Appendix A, including judge model prompts.

SMILE implementation. We choose Llama-3.2-3B-Instruct as our synthetic answer generator g and ember-V1[2] as our embedding model e. This combination is computationally lightweight: the 335M parameter ember-v1 can run inference on a CPU, and generating responses with the 3B Llama model requires $<$ 10GB of VRAM. Furthermore, our ablation study (Section 5.2) shows that larger models offer only marginal performance improvements, highlighting SMILE's inherent lightweight nature. SMILE scores, similar to GPT-4o, are discretized into six bins (0–5), with scores ≥ 4 considered correct. The N-gram value is dynamically set based on ground truth answer length, and the parameter w fixed to 0.3.

5.1 Main experimental results

Using our golden evaluation set, we compare SMILE against existing baseline metrics. To holistically assess evaluators-human agreement, we computed Pearson correlation, Kendall's Tau-b, and Deviation from Human Accuracy. Pearson correlation and Kendall's Tau measure agreement with human annotations on the instance level, ranging from –1 (perfect disagreement) to +1 (perfect agreement). Kendall's Tau-b, focuses on ranking consistency and accounts for ties in the data. Deviation from Human Accuracy quantifies the difference between QA accuracy derived from evaluator scores and human annotations.

Pearson correlation results are presented in Table 1. SMILE consistently outperforms other evaluation metrics across tasks, achieving the highest overall correlation with human evaluations. Notably, SMILE significantly surpasses GPT-4o and GPT-3.5, despite their prominence as LLM-as-judge evaluators. Across all tasks, SMILE's positive correlation scores are significantly closer to 1 than most competitors, indicating strong agreement with human evaluations and validating the robustness of our approach.

Kendall's Tau-b results, presented in Table 2, establish SMILE's superior correlation with human rankings. Quantitatively, SMILE outperforms all competing metrics, further validating its effectiveness. SMILE surpasses GPT-4o and GPT-3.5, underscoring its exceptional ability to rank generated responses in a way that closely mirrors human annotated rankings.

Table 3 shows SMILE's evaluated accuracy aligns closely with human judgments across all tasks. Unlike metrics like ROUGE-L, METEOR and sBERT, which suffers from inflated or deflated scores(especially with short ground-truths and verbose model outputs), SMILE remains robust even after binarization. This reinforces it's reliability and makes it a practical and consistent alternative to costly human evaluation.

Finally, we evaluate SMILE and GPT-3.5/4o as evaluators on QA-Eval, using two prompting variants: (1) original prompt (based on (Maaz et al., 2024)), and (2) extract-style prompt (asks LLM to extract short answer first). As shown in Table 4, SMILE consistently outperforms GPT-3.5 and closely matches GPT-4o. Notably, LLMs degrade under the extract prompt, highlighting SMILE's robustness and prompt independence.

5.2 Ablations

This section presents an ablation study of SMILE centered on three key perspectives: (1) *Component analysis*, systematically removing steps (synthetic answer generation, semantic similarity score, keyword score) to demonstrate their individual importance, (2) *Model scaling*, examining the impact of using larger models for both synthetic answer generation and embedding, (3) *Hyperparameter tuning*, analyzing the effect of the weight w in SMILE. Results are detailed in Table 5 and Figure 5.

Component analysis. SMILE comprises three key components: (1) semantic similarity, (2) lexical exactness, and (3) distribution alignment via a lightweight language model. Table 5 (top) summarizes the contribution of these components to SMILE's robust performance. Experiments demonstrate that *both keyword and sentence similarity*

[1]https://huggingface.co/sentence-transformers/all-roberta-large-v1

[2]https://huggingface.co/llmrails/ember-v1

Metrics	Video QA: Qwen2.5			Visual QA: llava 1.5 7B			Language QA: GPT-4o			Overall
	TGIF	MSVD	MSRVTT	TextVQA	DocVQA	POPE	MRQA	HotpotQA	MUSIQUE	
Exact Match	nan	nan	nan	nan	nan	0.099	nan	0.109	0.143	0.117
Easy Match	0.793	0.481	0.237	0.795	0.375	0.451	0.676	0.657	0.890	0.595
ROUGE-L	0.603	0.477	0.313	0.531	0.661	0.001	0.368	0.603	0.438	0.444
METEOR	0.663	0.527	0.311	0.636	0.706	0.086	0.454	0.664	0.599	0.516
BERTScore	0.421	0.357	0.099	0.358	0.633	0.164	0.272	0.620	0.421	0.372
sBERT	0.472	0.506	0.380	0.602	0.852	0.164	0.352	0.664	0.358	0.483
GPT-3.5	**0.825**	0.666	0.318	0.626	0.828	0.422	0.746	0.668	0.566	0.629
GPT-4o	0.778	**0.790**	**0.606**	**0.829**	0.790	0.699	0.760	0.678	0.814	0.693
SMILE	0.824	0.663	0.435	0.787	**0.908**	**0.716**	**0.872**	**0.943**	**0.977**	**0.792**

Table 1: **Pearson correlation with human judgments** (↑) across Video, Visual, and Language QA. SMILE consistently outperforms traditional metrics and LLM-based judges across all modalities, achieving the highest overall correlation.

Metrics	Video QA: Qwen2.5			Visual QA: llava 1.5 7B			Language QA: GPT-4o			Overall
	TGIF	MSVD	MSRVTT	TextVQA	DocVQA	POPE	MRQA	HotpotQA	MuSiQue	
Exact Match	nan	nan	nan	nan	nan	0.100	nan	0.109	0.145	0.118
Easy Match	0.765	0.500	0.253	**0.773**	0.361	0.420	0.676	0.657	0.890	0.588
ROUGE-L	0.598	0.542	0.357	0.543	0.673	0.162	0.350	0.496	0.555	0.475
METEOR	0.592	0.544	0.346	0.607	0.672	0.161	0.276	0.458	0.589	0.472
BERTScore	0.347	0.272	0.088	0.244	0.496	0.125	0.210	0.455	0.454	0.299
sBERT	0.393	0.411	0.297	0.435	0.662	0.102	0.289	0.455	0.386	0.381
GPT-3.5	0.738	0.555	0.294	0.589	0.709	0.441	0.490	0.439	0.588	0.538
GPT-4o	0.686	**0.717**	**0.561**	0.746	0.686	**0.676**	0.193	0.488	0.784	0.615
SMILE	**0.753**	0.637	0.432	0.714	**0.805**	0.607	**0.805**	**0.782**	**1.000**	**0.726**

Table 2: **Kendall's Tau-b with human judgments** (↑) across Video, Visual, and Language QA. SMILE consistently outperforms traditional metrics and LLM-based judges across all modalities, achieving the highest overall ranking agreement. nan indicates Exact Match (EM) found no exact matches.

scores are essential. Removing keyword scores significantly reduces Pearson correlation, underscoring the critical role of lexical exactness in QA evaluation. Conversely, relying solely on keyword scores neglects global structure, degrading performance notably in VidQA and VQA. Synthetic answers are also crucial, particularly for verbose model predictions in VidQA and VQA. Figure 4 illustrates the effect of synthetic answer generation, which effectively maps extremely short gold answers to longer model outputs. Combining sentence scores, keyword scores, and synthetic answers yields robust and accurate evaluation across domains.

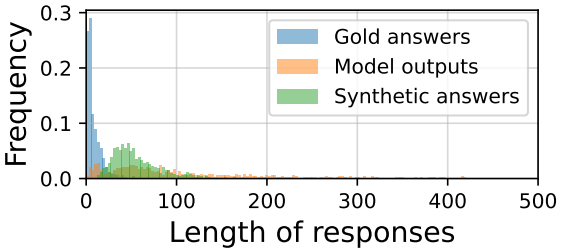

Figure 4: Length of gold answers, model outputs, and synthetic answers, across all domains and benchmarks in characters. Synthetic answers more align better with model outputs in terms of output length, enabling better semantic evaluation.

Model scaling. A key advantage of SMILE is that it offers the ability to *efficiently* run evaluation. Our model choices in Section 5 demonstrate this: SMILE at inference time requires only a 355M parameter embedding model, and pre-generating synthetic answers requires only a 3B generative model. Table 5 (bottom) further establishes SMILE's lightweight nature: increasing model capacity yields minimal performance gains, if at all. Our model ablation focused on the synthetic answer generation model and the embedding model. Using GPT-3.5-Turbo instead of Llama-3.2-3B-Instruct for synthetic answers yielded comparable correlation with human judgments, indicating that *effective synthetic answer generation is achievable with smaller lightweight models*. Replacing ember-v1 with the substantially larger GTE-7B (Li et al., 2023c) embedding model resulted in only a marginal performance gain of less than 2%, despite a 20× increase in model size. This indicates that *SMILE remains effective even with lightweight*

	Video QA: Qwen2.5			Visual QA: llava 1.5 7B			Language QA: GPT-4o			Overall
	TGIF	MSVD	MSRVTT	TextVQA	DocVQA	POPE	MRQA	HotpotQA	MuSiQue	
Human	0.64	0.48	0.56	0.40	0.24	0.80	0.92	0.88	0.80	**0.64**
Exact Match	-0.64	-0.48	-0.56	-0.40	-0.24	-0.76	-0.92	-0.80	-0.72	0.53
Easy Match	**0.04**	-0.12	-0.20	-0.04	**0.00**	0.16	-0.08	-0.12	-0.04	0.09
ROUGE-L	-0.64	-0.48	-0.56	-0.40	-0.24	-0.72	-0.76	-0.40	-0.64	0.54
METEOR	-0.64	-0.48	-0.56	-0.40	-0.20	-0.80	-0.68	-0.28	-0.52	0.51
BERTScore	0.36	0.52	0.44	0.60	0.76	0.20	0.08	0.12	0.20	0.36
sBERT	-0.64	-0.48	-0.56	-0.4	-0.24	-0.76	-0.92	-0.08	-0.72	0.52
GPT-3.5	-0.12	**-0.08**	**0.00**	**0.00**	-0.08	-0.12	0.04	**0.00**	0.04	**0.05**
GPT-4o	**-0.04**	-0.12	-0.04	**0.00**	-0.12	**-0.08**	**0.00**	-0.04	0.04	**0.05**
SMILE	**-0.04**	0.16	0.20	-0.04	0.04	-0.12	-0.04	**0.00**	**0.00**	0.07

Table 3: **Deviation from human-evaluated accuracy** across Video, Visual, and Language QA tasks. SMILE closely aligns with human judgment.

	GPT 3.5		GPT-4o		Overall
	NQ	TQ	NQ	TQ	
GPT-3.5, original prompt	0.756	0.849	0.713	0.706	0.756
GPT-4o, original prompt	0.865	0.913	0.815	0.806	0.850
GPT-3.5, extract prompt	0.478	0.572	0.413	0.440	0.476
GPT-4o, extract prompt	0.831	0.898	0.783	0.774	0.821
SMILE	0.829	0.889	0.786	0.760	0.816

Table 4: **Pearson Correlation with human judgment on QAEval.** SMILE shows strong agreement with human annotations, outperforming GPT-3.5 and roughly matching GPT-4o.

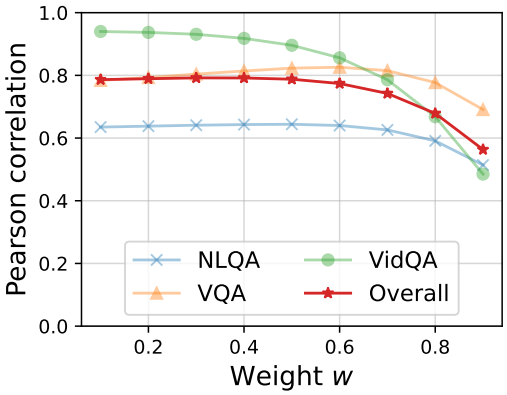

Figure 5: Sweep of w, which trades off lexical exactness for semantic similarity as w increases. SMILE exhibits relatively stable aggregate performance for $w \leq 0.5$.

		Video QA	Visual QA	Language QA	Overall
Component Ablation	**SMILE**	0.641	0.804	0.931	**0.792**
	w/o keyword scores	0.383	0.533	0.249	0.388
	w/o sentence scores	0.682	0.764	0.883	0.776
	w/o synthetic answers	0.638	0.764	0.883	0.776
Model Ablation	**SMILE**	0.641	0.804	0.931	**0.792**
	Embedding: GTE_{7B}	0.647	0.824	0.947	0.806
	Syn. answer: GPT-3.5-Turbo	0.636	0.802	0.930	0.790

Table 5: **Component and model ablations.** Performance is assessed by Pearson correlation. Keyword scores are the primary contributor, highlighting the importance of lexical exactness. Embedding model scaling yields marginal ($< 2\%$) gains.

embedding models.

Hyperparameter tuning. SMILE's lone hyperparameter w allows practitioners to precisely decide the impact of the semantic and keyword subscores. Specifically, as w increases, more importance is given to the semantic subscore. As we show in Figure 5, overall performance is relatively stable for $w \leq 0.5$ before smoothly decreasing. This aligns with results from our component ablation study in Table 5: The keyword subscores alone exhibited relatively strong performance, while the semantic subscore fared worse. However, SMILE hyperparameter choice is relatively forgiving, with any choice of w that slightly upweights the keyword subscore likely to perform well.

6 Conclusion

We introduce SMILE, a novel, lightweight QA evaluation metric that overcomes limitations of existing methods by integrating semantic and lexical analysis. Its efficiency addresses the high cost, biases, and inconsistencies of LLM-based evaluators. Benchmarking across text, image, and video QA demonstrates SMILE's strong correlation with human judgment, surpassing traditional metrics and LLM judges like GPT-4o. Its design also offers interpretability, and ablation studies validate the importance of its components. In summary, SMILE provides a robust, efficient, and interpretable solution for QA evaluation across modalities, effectively balancing lexical precision and semantic relevance as a promising alternative to costly LLM evaluations.

Limitations

Although SMILE offers a lightweight, interpretable, and scalable alternative to LLM-based evaluators, it comes with certain limitations. (1) SMILE is designed for source-free evaluation and does not access the context. Although efficient, this may cause it to miss context-dependent errors. (2) The metric relies on synthetic answers to align ground-truths with model outputs. The quality of these synthetic answers can affect the the scoring, especially in long-form or open-ended responses. (3) Our evaluation is limited to factoid QA tasks. SMILE's effectiveness on complex reasoning, multi-hop, or conversational QA remains unexplored. (4) SMILE includes a weighting parameter to balance lexical and semantic components, which may require tuning for specific tasks or domains.

References

Aishwarya Agrawal, Ivana Kajić, Emanuele Bugliarello, Elnaz Davoodi, Anita Gergely, Phil Blunsom, and Aida Nematzadeh. 2022. Reassessing evaluation practices in visual question answering: A case study on out-of-distribution generalization. *arXiv preprint arXiv:2205.12191*.

Junyi Ao, Yuancheng Wang, Xiaohai Tian, Dekun Chen, Jun Zhang, Lu Lu, Yuxuan Wang, Haizhou Li, and Zhizheng Wu. 2024. Sd-eval: A benchmark dataset for spoken dialogue understanding beyond words. *arXiv preprint arXiv:2406.13340*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, and 1 others. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

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.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Boerschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. *arXiv preprint arXiv:2202.07654*.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd workshop on machine reading for question answering*, pages 119–124.

Cheng-Han Chiang and Hung-Yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. *arXiv preprint arXiv:1910.09753*.

Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Alon Jacovi, Andrew Wang, Chris Alberti, Connie Tao, Jon Lipovetz, Kate Olszewska, Lukas Haas, Michelle Liu, Nate Keating, Adam Bloniarz, and 1 others. 2025. The facts grounding leaderboard: Benchmarking llms' ability to ground responses to long-form input. *arXiv preprint arXiv:2501.03200*.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2758–2766.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2024. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. *Preprint*, arXiv:2409.12941.

Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Joongbo Shin, and Kyomin Jung. 2020. Kpqa: A metric for generative question answering using keyphrase weights. *arXiv preprint arXiv:2005.00192*.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023a. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023c. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *NeurIPS*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023c. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Man Luo, Shailaja Keyur Sampat, Riley Tallman, Yankai Zeng, Manuha Vancha, Akarshan Sajja, and Chitta Baral. 2021. 'just because you are right, doesn't mean i am wrong': Overcoming a bottleneck in development and evaluation of open-ended vqa tasks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.

Oscar Mañas, Benno Krojer, and Aishwarya Agrawal. 2024. Improving automatic vqa evaluation using large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Minesh Mathew, Dimosthenis Karatzas, R Manmatha, and CV Jawahar. 2020. Docvqa: A dataset for vqa on document images. corr abs/2007.00398 (2020). *arXiv preprint arXiv:2007.00398*.

Xuan-Phi Nguyen, Shrey Pandit, Senthil Purushwalkam, Austin Xu, Hailin Chen, Yifei Ming, Zixuan Ke, Silvio Savarese, Caiming Xong, and Shafiq Joty. 2024. Sfr-rag: Towards contextually faithful llms. *arXiv preprint arXiv:2409.09916*.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*.

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

Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. 2024. Offsetbias: Leveraging debiased data for tuning evaluators. *arXiv preprint arXiv:2407.06551*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic answer similarity for evaluating question answering models. *arXiv preprint arXiv:2108.06130*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Tu Shiwen, Zhao Liang, Chris Yuhao Liu, Liang Zeng, and Yang Liu. 2024. Skywork critic model series. https://huggingface.co/Skywork.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Tu Vu, Kalpesh Krishna, Salaheddin Alzubi, Chris Tar, Manaal Faruqui, and Yun-Hsuan Sung. 2024. Foundational autoraters: Taming large language models for better automatic evaluation. *arXiv preprint arXiv:2407.10817*.

Cunxiang Wang, Sirui Cheng, Qipeng Guo, Yuanhao Yue, Bowen Ding, Zhikun Xu, Yidong Wang, Xiangkun Hu, Zheng Zhang, and Yue Zhang. 2023a. Evaluating open-qa evaluation. *Advances in Neural Information Processing Systems*, 36:77013–77042.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023b. Is chatgpt a good nlg evaluator? a preliminary study. In *Proceedings of EMNLP Workshop*, page 1.

Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. 2024a. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*.

Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2024b. Direct judgement preference optimization. *Preprint*, arXiv:2409.14664.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023c. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Yidong Wang, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, and 1 others. 2023d. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. In *The Twelfth International Conference on Learning Representations*.

Austin Xu, Srijan Bansal, Yifei Ming, Semih Yavuz, and Shafiq Joty. 2025. Does context matter? contextual-judgebench for evaluating llm-based judges in contextual settings. *arXiv preprint arXiv:2503.15620*.

Austin Xu, Will Monroe, and Klinton Bicknell. 2024. Large language model augmented exercise retrieval for personalized language learning. In *Proceedings of the 14th Learning Analytics and Knowledge Conference*, pages 284–294.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

An Yang, Kai Liu, Jing Liu, Yajuan Lyu, and Sujian Li. 2018a. Adaptations of rouge and bleu to better evaluate machine reading comprehension task. *arXiv preprint arXiv:1806.03578*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in neural information processing systems*, 34:27263–27277.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2024. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36.

A Additional SMILE and baseline implementation details

A.1 Prompt templates

As described in Section 4, we prompt the synthetic answer generator with the original question and ground-truth answer and task it with generating a synthetic answer. We provide the full prompt below.

```
### Synthetic answer generation prompts
## System prompt:
You are an intelligent chatbot designed for generating
answer as a sentence from question-answer pairs.
Your task is to generate a single sentence answer
using the question and the answer already provided.
Here's how you can accomplish the task:
------
##INSTRUCTIONS:
- Look at the provided answer.
- Generate a short single sentence response using the
question and the answer.
- Response SHOULD ALWAYS USE THE WORDS FROM ANSWER
provided.
- DO NOT USE THE QUESTION AS IT IS IN THE RESPONSE.
- Return only the response and nothing else.

## User prompt
Please phrase a short single sentence answer using
question-answer pair only:
Question: {<question>}
Answer: {<answer>}
DO NOT PROVIDE ANY OTHER OUTPUT APART FROM A SINGLE
SHORT SENTENCE.
```

To prompt GPT-4o and GPT-3.5 as judge models, we utilize prompts adopted from (Maaz et al., 2024), as described in Section 5. We provide full prompts below.

```
### Original prompt: GPT-4o/GPT-3.5-Turbo judge prompts
## System prompts
You are an intelligent chatbot designed for evaluating
the correctness of generative outputs for
question-answer pairs.
Your task is to compare the predicted answer with the
correct answer and determine if they match
meaningfully. Here's how you can accomplish the task:
------
##INSTRUCTIONS:
- Focus on the meaningful match between the predicted
answer and the correct answer.
- Consider synonyms or paraphrases as valid matches.
- Evaluate the correctness of the prediction compared
to the answer.

## User prompt
Please evaluate the following video-based
question-answer pair:
Question: {<question>}
Correct Answer: {<answer>}
Predicted Answer: {<model_output>}
Provide your evaluation only as a yes/no and score
where the score is an integer value between 0 and 5,
with 5 indicating the highest meaningful match.
Please generate the response in the form of a Python
dictionary string with keys 'pred' and 'score', where
value of 'pred' is  a string of 'yes' or 'no' and
value of 'score' is in INTEGER, not STRING.
DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION.
Only provide the Python dictionary string.
For example, your response should look like this:
{{'pred': 'yes', 'score': 4}}.
```

```
### Extract prompt: GPT-4o/GPT-3.5-Turbo judge prompts
## System prompts
You are an expert evaluator for video-based question
answering systems. Your task is to judge the factual
accuracy of a predicted answer by comparing it to a
correct answer. You will follow a structured
evaluation approach to ensure consistency:
------
## INSTRUCTIONS:
Step 1: Extract the key facts from the Correct Answer.
Step 2: In case the Correct Answer is a list, choose
the best answer that matches the Predicted Answer.
Step 3: Extract the key facts from the Predicted
Answer.
Step 4: Compare the two sets of facts and determine
how consistent they are.
 - Consider paraphrasing, synonyms, and partial
overlaps.
 - Ignore grammatical errors.
 - Penalize hallucinated or contradicted information.
Step 4: Based on the comparison, assign a factual
accuracy score between 0 and 5 (INTEGER only), where:
 5 = Fully accurate and aligned
 4 = Mostly accurate, minor omissions or paraphrasing
 3 = Partially correct but with notable missing or
incorrect info
 2 = Limited accuracy, mostly incorrect or unrelated
 1 = Completely inaccurate
 0 = No relation or total hallucination
Respond strictly in the following format:
{'score': X, 'pred':Y} where X is an integer between 0
and 5 and Y is a either 'yes'(X>3) or 'no'(X<=3). Do
not include any explanation or extra text.}

## User prompt
Evaluate the following video-based QA pair:
Question: {<question>}
Correct Answer: {<answer>}
Predicted Answer: {<model_output>}
Return your evaluation following the instructions
above.
DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION.
Only provide the Python dictionary string.
```

A.2 SMILE text processing

As a part of text pre-processing, we perform standard text normalization on words present in ground truth answers and predictions. We first convert each string to lower case and remove all punctuation. Then, each word is lemmatized using POS-aware lemmatization to capture accurate base forms. If the resulting processed word is empty after these steps, the original lower-case word is retained.

A.3 Metrics conversion to accuracy

For all evaluated baselines and metrics, we must convert from scores to binary correct or incorrect accuracy labels. ROUGE, METEOR, BERTScore, sBERT, and SMILE all output continuous-valued scores between 0 and 1. We apply a threshold of 0.67, considering anything above the threshold to be correct and anything below to be incorrect. The choice of 0.67 is the same as considering anything with a score of 4 or above to be correct after converting the continuous [0,1] score to a 0-5 scale with uniform binning. For GPT-3.5-Turbo and GPT-4o,

the model is prompted to output a yes/no label for correctness, which we use directly.

B Data annotation details

B.1 Annotation instructions

Annotators were given with detailed instructions on how to annotate responses. We adopted a 3 point scale: `clearly incorrect`, `unclear`, and `clearly correct`. We defined each of these categories as follows:

Clearly incorrect: The model definitively produces a response that is incorrect.

Unclear: The model response cannot be confirmed correct from the ground-truth answer.

Clearly correct: The model response can be explicitly verified as correct using the ground-truth answer.

We also defined edge case behavior:

Extraneous information: If the model response correctly answers the question, but includes other information that may or may not be factual, we consider the response `clearly correct`. As a concrete example, for question "What brand of soda is in this picture" with ground-truth "Coca-Cola", we consider the model response "Coca-Cola is in this picture. It is the most popular soda in the world by unit sales and has over 60 different flavors" to be correct, even though it contains extraneous factually verifiable information.

Synonyms or ambiguous subjects: We consider a model response that answers the question using an ambiguous subject to be `unclear`. As a concrete example, for question "who describes a video game??" with ground-truth "man", we consider the model response "person" to be unclear, as it does not describe in sufficient detail the person.

B.2 Annotation aggregation and conversion to accuracy labels

We collected responses from four annotators. To aggregate individual annotations into a single label, we utilized majority vote, employing random tie-breaking as needed. To form final accuracy labels, we consider `clearly correct` responses to be accurate and consider `unclear` and `clearly incorrect` responses to be inaccurate.

C Cost effective Model selection with SMILE

Selecting optimal checkpoints during ML model training is crucial for maximizing performance on downstream tasks. Traditionally, this selection process relies on not-so reliable metrics like METEOR and ROUGE, or expensive metrics such as LLM-based judge evaluations. In this experiment, we use SMILE to identify the best checkpoint. Specifically, we select two intermediate checkpoints(with similar performance) from the Video model and evaluate their performance on the TGIF benchmark. The evaluation is conducted using five metrics: GPT-4o, GPT-3.5-turbo, METEOR, sBERT and SMILE.

Metrics	Checkpoint 1		Checkpoint 2	
	Rank	Cost($)	Rank	Cost($)
GPT-4o	1	12	2	11.99
GPT-3.5-turbo	1	12	2	12.00
METEOR	1	-	2	-
sBERT	2	-	1	-
SMILE	**1**	-	**2**	-

Table 6: Checkpoint selection on TGIF video QA (Video model): Rank (1=best) and approximate evaluation cost (USD) per metric. SMILE ranks checkpoints similarly to GPT metrics, but without inference costs. "-" denotes methods without API inference cost.

As per Table 6, our findings demonstrate that *SMILE selects the same optimal checkpoint (i.e. checkpoint 1) as GPT-4o and GPT-3.5-turbo*. This alignment highlights SMILE's effectiveness, emphasizing its capability to provide reliable checkpoint selection without incurring additional evaluation cost.

An advantage of SMILE is its substantial reduction in evaluation costs compared to GPT-based models. GPT-4o and GPT-3.5 cost's around $12 for each checkpoint evaluation on TGIF, and the cost increases as more checkpoints and evaluation benchmarks are added. In contrast, SMILE has almost no extra cost. Therefore, adopting SMILE not only maintains performance accuracy but also significantly lowers monetary overhead, making it a highly efficient and scalable solution for checkpoint selection.

D SMILE as a drop-in replacement for GPT-4o

Building on SMILE's alignment with human judgment, we now demonstrate its capability to supplant GPT-4o as an evaluation metric. To do so, we compare model accuracy derived from SMILE scores against that from GPT-4o-based evaluation each benchmark's *complete test-set.* We find that SMILE exhibits the lowest overall deviation among

all tested methods, as summarized in Table 7. This compelling result strongly suggests SMILE is a reliable and direct alternative to resource-intensive LLM-as-judge approaches like GPT-4o.

E SMILE Interpretability Examples

As discussed in Section 4.2, we provide detailed SMILE subscores in Figure 6 for the examples from Figure 2. In the TGIF example from Figure 6, the model output shows a high semantic score s_s (Equation (1)), reflecting strong relevance to the synthetic answer. The lexical relevance score s_l (Equation (2)) is also high, indicating a perfect overlap with the ground truth. To clarify which word contributes most to the keyword score, we also return the word(s) with the maximum similarity ("max sim words"). These components together offer actionable insights into model strengths and weaknesses, helping guide targeted improvements.

F Supplement ablation results

In this section, we present additional plots to supplement our Model Ablation described in Section 5.2. Specifically, we include scatter plots and distribution plots to further illustrate the performance difference when varying model choices for synthetic answer generation and embedding.

F.1 Synthetic answer generation ablation

Referring to Figure 7, we see a very strong linear correlation between the two sets of generated synthetic answers and thus backs our claim that *generating synthetic answers is a fairly simple task* as mentioned in Section 5.2. Figure 8, further bolster our claim, and highlights that the 'avg score' distribution remains very similar, hence we see a marginal difference in the performance as reported in table 5.

F.2 Embedding model ablation

Figure 9 and Figure 10 provides insight into the performance variation observed in Table 5, highlighting that keyword scores exhibit greater sensitivity to the choice of embedding model compared to sentence scores.

	Video QA: Qwen2.5			Visual QA: llava 1.5 7B			Language QA: GPT-4o			
	TGIF	MSVD	MSRVTT	TextVQA	DocVQA	POPE	MRQA	HotpotQA	MuSiQue	Overall
GPT-4o	0.705	0.657	0.503	0.436	0.191	0.783	0.920	0.909	0.700	**0.645**
Exact Match	-0.705	-0.657	-0.503	-0.424	-0.187	-0.782	-0.877	-0.884	-0.681	0.633
Easy Match	**-0.025**	**-0.217**	**-0.106**	-0.056	**-0.044**	**-0.021**	-0.083	-0.152	-0.135	0.093
ROUGE-L	-0.705	-0.657	-0.503	-0.413	-0.179	-0.773	-0.648	-0.493	-0.553	0.547
METEOR	-0.705	-0.657	-0.503	-0.405	-0.166	-0.783	-0.592	-0.509	-0.454	0.530
BERTScore	0.294	0.340	0.497	0.562	0.805	0.216	**0.008**	0.091	0.300	0.354
sBERT	-0.705	-0.657	-0.503	-0.379	-0.145	-0.780	-0.566	-0.399	-0.504	0.515
SMILE	-0.032	-0.241	0.132	**0.021**	0.104	-0.041	**-0.008**	**-0.016**	**0.005**	**0.067**

Table 7: **Deviation from GPT-4o accuracy** across Video, Visual, and Language QA tasks, using *complete test sets; Accuracy may deviate human annotated test-set results in Table 3*. SMILE exhibits the smallest deviation among evaluators, closely aligning with GPT-4o.

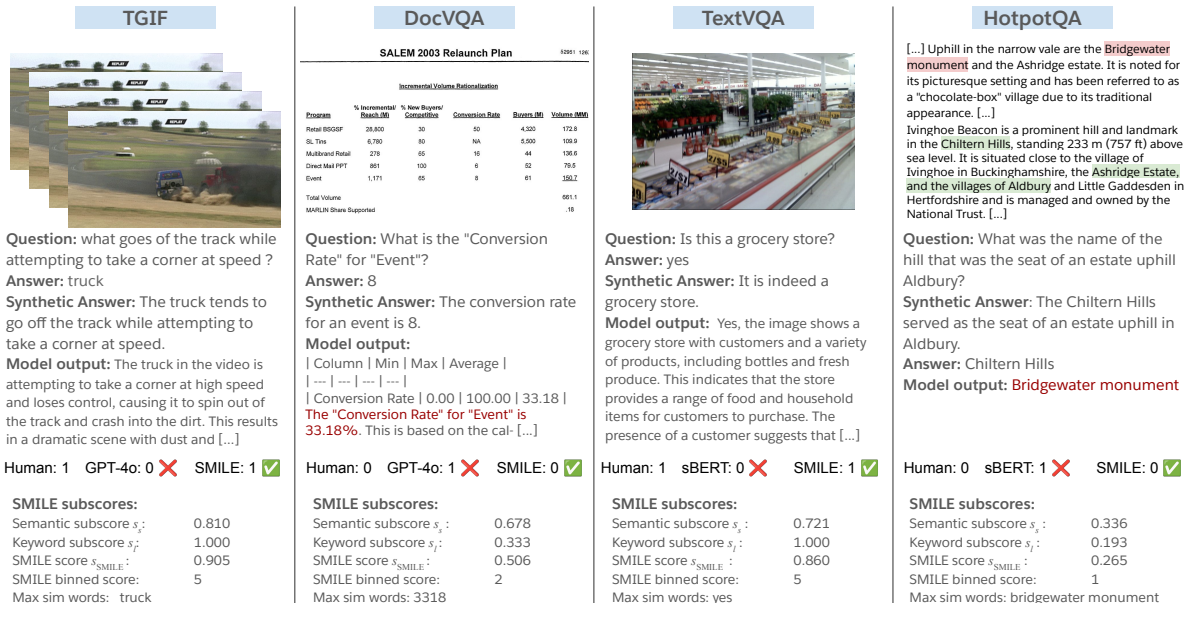

Figure 6: **Example failure cases of existing methods with detailed SMILE scores**.

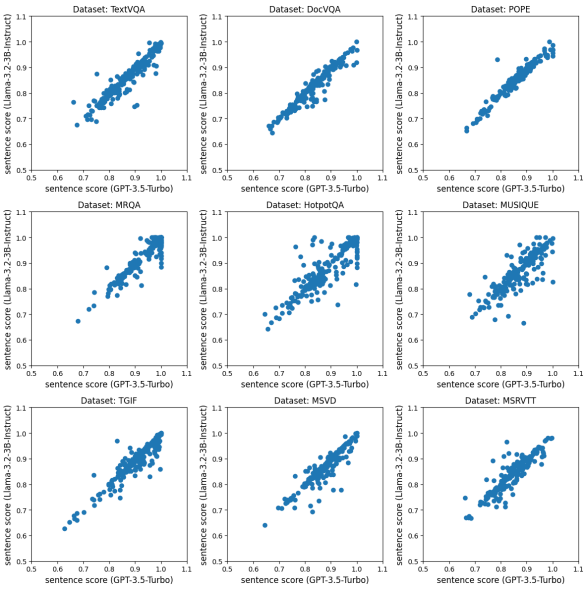

Figure 7: Distribution analysis of SMILE sentence embedding scores across different synthetic answer sets. A strong linear relationship is observed between the two synthetic answer sets, indicating that synthetic answers can reliably be generated using any state-of-the-art generation model.

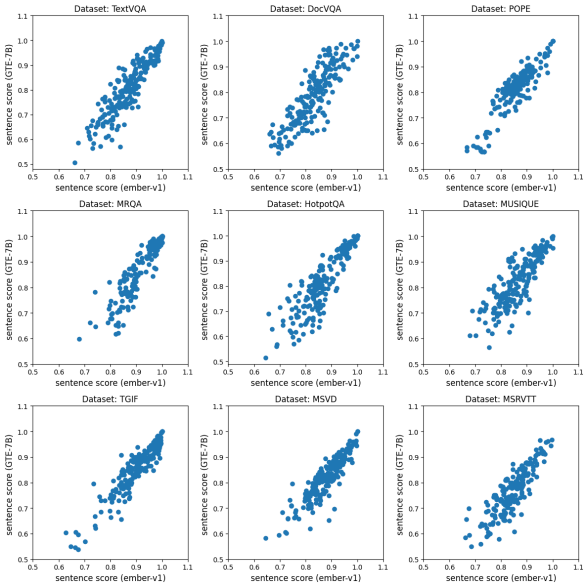

Figure 9: Analyzing Sentence score distributions using different embedding models. Sentence scores show stronger linear correlation, indicating that it is robust to change in embedding model.

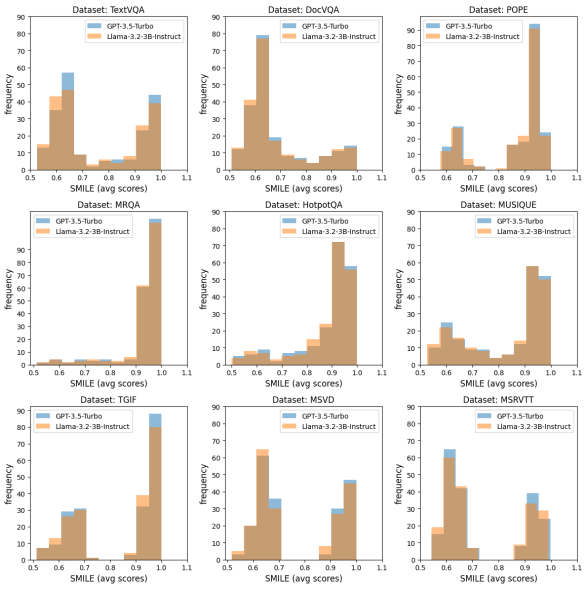

Figure 8: Distribution analysis of 'SMILE avg scores' across different synthetic answer sets. We see a very similar score distribution, highlighting the fact the performance remains very similar.

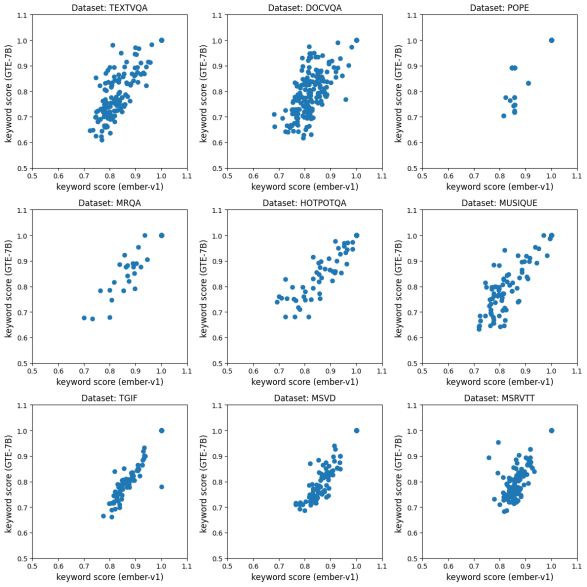

Figure 10: Analyzing Keyword score distributions using different embedding models. Keyword scores show a linear correlation, but has some added noise, indicating that it is more sensitive to changes in embedding models.