



# Human-AI Co-creation: LLMs, Contextual Hints, Performance

Panagiotis Gratsanis<sup>1</sup>, Ioannis Karydis<sup>1</sup>, Spyros Sioutas<sup>2</sup>, and  
Gerasimos Vonitsanos<sup>2</sup>

<sup>1</sup> Dept. of Informatics, Ionian University, 49132 Kerkyra, Greece

<sup>2</sup> Dept. of Computer Engineering and Informatics, University of Patras, 26 504  
Patras, Greece

{pgratsanis,karydis}@ionio.gr, {sioutas,mvonitsanos}@ceid.upatras.gr

**Abstract.** Human-AI collaboration has increasingly impacted domains such as education, medicine, creative arts, and complex problem-solving, where differentiated interactions greatly influence outcomes. Effective collaboration hinges on the capability of Large Language Models (LLMs) to accurately interpret subtle contextual hints provided by humans, underscoring the need to understand how contextual information influences LLMs’ responses. To address this, herein, a structured benchmark is introduced for evaluating the effect of contextual hints on LLMs’ performance, providing sample data and code to enable reproducible experimentation. Experimental findings support the effectiveness of contextual hints, demonstrating these significantly shape the complexity and structural variation in model-generated outputs. Specifically, prompts enhanced by hints yield more detailed, paraphrased responses that diverge lexically and structurally from concise ground-truth answers. However, traditional similarity metrics often underestimate the value of these contextually enriched responses due to their lexical diversity, indicating a discrepancy between metric-based evaluations and human perceptions of quality. These insights highlight the importance of adopting more context-sensitive evaluation methods to better capture the quality and semantic richness of collaborative human-AI outputs.

**Keywords:** Large Language Models · Prompt engineering · Contextual hints · Performance evaluation · Co-creation.

## 1 Introduction

Large Language Models (LLMs), such as OpenAI’s (chat)GPT<sup>3</sup>, Anthropic’s Claude<sup>4</sup>, Google’s Gemini<sup>5</sup>, Meta’s LLaMA<sup>6</sup> to name but a few, have ushered a new era in successfully managing a vast breadth of tasks such as text generation, language understanding, arithmetic reasoning, contextual comprehension, as well as unprecedented capabilities in Natural Language Processing (NLP) [6].

---

<sup>3</sup> <https://chatgpt.com/>

<sup>4</sup> <https://claude.ai/>

<sup>5</sup> <https://deepmind.google/technologies/gemini/>

<sup>6</sup> <https://www.llama.com/>

LLMs’ initial focus was on NLP tasks, though latest research has also addressed their perception and ability to respond with multimodal information in order to complement their text-modalities [13]. Accordingly, LLMs’ input and output extended from text to other types such as video, image, and audio and thus their domain of usage grew significantly, beyond Computer Science, to fields such as Biology, Psychology, Physics, Political Science, Law, Art, History and many others [29]. It is thus only natural that their ability for complex creative tasks [12] has been used collaboratively with human creativity for content generation, education, programming, and creative arts, among others [43].

The majority of publicly available LLMs utilize a passive interaction mode wherein users provide, mostly textual, instructions, or prompts, in order to elicit the activation of the LLM and guide it in generating the desired output response [39]. Accordingly, the activity of “prompt engineering” evolved in order to assist in the creation of appropriate prompts aimed at improving LLMs’ accuracy, coherence, and factual consistency [9]. Evidently, prompting LLMs is a natively co-creating endeavor of, admittedly, varying degree given the Standard Definition of creativity as “the production of novel and useful ideas and products” [41].

For this co-creating process, a plethora of detailed human-AI interactions have been proposed [1] but the most readily available to users, and commonly used scenario, is the aforementioned prompting or guiding model’s output [11]. The traditional multi-step method of training LLMs has led research on prompt engineering to utilize context as an effective way to increase performance [7], especially for nuanced queries. Still, research on assessing how LLMs integrate subtle contextual cues, such as hints, and how these cues influence the structure, content & semantic alignment of the model-generated outputs is still scarce.

### 1.1 Motivation & Contribution

The motivation behind this work lies in the growing adoption of LLMs across domains wherein Human-AI Co-creation events take place that demand high interpretability and contextual sensitivity. For example, in educational settings, medical communication, problem solving, and almost all artistic expression, subtle phrasing and domain-aware cues can drastically alter the relevance and clarity of model outputs. Understanding how models respond to such contextual elements is thus crucial for designing effective prompts and evaluation pipelines.

Thus, in order to address the aforementioned challenges, the key contributions of this work are:

- Introduction of a structured benchmark that evaluates the influence of contextual hints on LLMs’ performance,
- Provision of sample data and programming code for the reproducibility of the proposed experimentation, and
- Experimentation results that strengthen the argument for the advantages of using contextual information in prompt engineering.

The remainder of the work is organized as follows: Section 2 details the related work and background information needed to contextualize the work. The proposed benchmark framework for experimentation is detailed in Section

3, while the results and related discussion on the evaluation of the proposed method is presented in Section 4. Finally, the work is concluded in Section 5.

## 2 Background & Literature Review

### 2.1 Large Language Models

LLMs have redefined the landscape of NLP, with architectures such as GPT (Generative Pretrained Transformers) [4], BERT (Bidirectional Encoder Representations from Transformers) [10], and T5 (Text-to-Text Transfer Transformer) [31] setting the foundation for modern generative AI.

These models are pretrained on massive corpora of text, acquiring general understanding of language enabling them to perform various tasks with minimal supervision. Scaling in parameters and data has proven critical in improving performance, as illustrated in the transition from GPT-2 to GPT-3 [4].

A key attribute of LLMs is their ability to integrate context into their input, not only at the sentence level, but across paragraphs and documents. Contextual understanding significantly influences output quality, particularly in generative tasks such as question answering, summarization, and text completion [21].

Recent studies [46] suggest that even subtle variations in prompt phrasing or the presence of contextual hints can lead to substantial changes in the quality, coherence, and accuracy of outputs. However, these effects remain largely underexplored in existing evaluation frameworks.

Despite their strengths, LLMs still exhibit limitations, including instability in output, incomplete semantic understanding, and a lack of transparency in decision-making processes [48]. Understanding how LLMs process contextual hints is essential for designing more robust prompting strategies and a central focus of this work.

### 2.2 Prompting Techniques

Prompting has become a core interface for interacting with LLMs, functioning as a method for task specification through natural language. Rather than relying exclusively on fine-tuning, prompting allows users to elicit desired outputs by crafting the input strategically [4].

The main prompting paradigms include: Zero-shot prompting, where no example is given, [18]; One-shot prompting, where a single example is provided, [14]; Few-shot prompting, where multiple examples illustrate the task [4].

These methods revealed the capacity of LLMs for in-context learning, where task behavior is inferred from the examples provided in the prompt. This has led to the rise of prompt engineering—the process of optimizing prompts to guide models more effectively [34]. Practices in prompt engineering include role-based instructions, rephrased questions, explicit definitions, and inclusion of relevant background knowledge. Among these, contextual prompts—that is, prompts enriched with indirect or task-relevant information—have shown promise in improving reasoning, coherence, and factual accuracy [17]. Yet, the influence of hints or context is difficult to quantify systematically, especially in the absence of benchmarks designed to isolate and evaluate such effects. This gap motivates the design of new evaluation strategies, such as the one proposed in this study.

### 2.3 Benchmarking in NLP

Benchmarking is crucial for assessing and comparing NLP models. Standard benchmarks such as GLUE [42] and SuperGLUE [32] test general language understanding across a suite of tasks, while SQuAD [32] focuses on reading comprehension. More recent benchmarks, such as BIG-bench Hard [15], its predecessor BIG-bench [37] and HELM [19], aim to assess the broader capabilities and responsible use of LLMs. However, most benchmarks rely on fixed prompt structures, and seldom account for contextual variation or hint sensitivity. This limits our ability to evaluate how prompt phrasing or contextual enrichment affects model performance.

### 2.4 Comparison Methods

Evaluating LLMs’ outputs, especially in generative tasks, requires more than simple string matching. Traditional metrics such as BLEU [26], ROUGE [20], and METEOR [2] focus on n-gram overlap and are useful for structured tasks like translation. However, they often fall short in open-ended generation where semantically valid outputs may diverge in wording.

To overcome these limitations, semantic-based metrics have been introduced. BERTScore [45] uses contextual embeddings to compute similarity between generated and reference texts, aligning more closely with human judgments. MoverScore [47] and BLEURT [36], further enhance semantic sensitivity using pre-trained language model embeddings and learned scoring functions.

Despite these advances, human evaluation remains essential, especially for assessing aspects like fluency, factual correctness, and relevance to context. Therefore, this study adopts a hybrid evaluation framework, combining automatic semantic similarity scores with targeted human assessment to analyze the influence of contextual hints. The challenge of comparing short text segments has also been addressed by Metzler et al. [23], who investigated lexical matching, stemming, and query expansion techniques to improve similarity estimation in sparse, context-poor environments. Their analysis emphasizes the limitations of conventional metrics when applied to brief inputs and highlights key trade-offs between effectiveness and computational efficiency. These findings support the need for hybrid and adaptive evaluation strategies in domains involving short, informative responses such as the ones explored in our study.

### 2.5 Metrics Analysis

To comprehensively evaluate the alignment between LLMs’ answers and the ground-truth, we employed four similarity metrics: Cosine Similarity, Jaccard Similarity, Edit Distance Similarity, and Word Overlap Similarity. Each metric captures a different dimension of similarity—semantic, lexical, structural, or token-level—and offers unique insights into the behavior of large language models under varying prompting conditions. Together, they enable a multifaceted view of textual correspondence that balances precision with interpretive flexibility.

**Cosine similarity** [35] is a metric that evaluates how similar two vectors are in terms of direction, regardless of their magnitude. It is computed as the cosine of the angle between two non-zero vectors A and B:

cosine similarity( $A, B$ ) =  $(A \cdot B) / (||A|| ||B||)$  where  $A \cdot B$  is the dot product of  $A$  &  $B$  and  $||A||, ||B||$  their Euclidean norms. The result ranges from -1 (completely opposite) to 1 (identical). In text analysis, cosine similarity is commonly used with TF-IDF vectorization to assess the semantic alignment between documents or responses without being affected by their length.

**Jaccard similarity** [40] measures the proportion of shared elements between two sets. For two token sets  $A$  and  $B$ , it is defined as:

$$\text{jaccard similarity}(A, B) = |A \cap B| / |A \cup B|$$

This metric produces a score between 0 and 1, where 1 indicates complete overlap and 0 no shared tokens. Jaccard is particularly sensitive to differences in lexical choice and penalizes paraphrasing or verbosity. It is often used in text comparison tasks to quantify literal word overlap, making it useful when structural fidelity is important.

**Edit distance** similarity quantifies how different two strings are by counting the minimum number of operations, insertions, deletions, or substitutions, required to transform one string into another. A widely used version of this is the Levenshtein distance. To express it as a similarity score, we use the normalized form: Chakraborty et al. [5] offer an efficient approach for computing edit distance in large-scale similarity search scenarios, grounding the metric in scalable NLP applications. Khalid et al. [16] further extend this by proposing parallel computation strategies to overcome performance bottlenecks in string similarity joins. McCauley [22] explores the use of edit distance in combination with locality-sensitive hashing for approximate nearest neighbor search, demonstrating its algorithmic versatility in large datasets.

$$\text{edit similarity}(A, B) = 1 - (\text{edit distance}(A, B) / \max(\text{len}(A), \text{len}(B)))$$

This produces a value between 0 (completely different) and 1 (identical). Edit distance is sensitive to word order and syntactic structure, making it valuable for detecting rephrasings or syntactic variation in model outputs. In text comparison, it serves as a structural alignment metric that complements lexical and semantic approaches.

**Word Overlap Similarity** is a set theoretic, asymmetric similarity measure defined as  $\text{Overlap}(A, B) = |A \cap B| / \min(|A|, |B|)$ , where  $A$  and  $B$  are sets of words. It quantifies shared elements relative to the smaller set, yielding values in  $[0, 1]$ . Unlike symmetric measures (e.g. Jaccard), it emphasizes inclusion, making it suitable for tasks where containment is key. It does not satisfy metric space properties (e.g. symmetry, triangle inequality), classifying it as a proximity—not distance—measure. Its computational simplicity suits applications like document clustering and semantic filtering. It has effectively been applied in hierarchical clustering [30] and gene function comparison [24].

### 3 Methodology

This study employed a systematic evaluation framework to examine how contextual hints affect the performance of large language models (LLMs) in answering domain-specific questions. The models under investigation were OpenAI’s

ChatGPT-3.5 and GPT-4o. To enable a structured comparison, we constructed two prompting conditions: (1) No Hint, where each model received only the question prompt; and (2) With Hint, where additional contextual cues were provided to guide the model’s reasoning.

The dataset comprises of 99 questions from the textbook “Biology: The Unity and Diversity of Life” [38]. These questions span multiple categories—multiple-choice, open-ended, data-driven, and critical-thinking—and were selected from the first nine chapters. Each question was normalized and aligned with its respective correct answer (or ground-truth), allowing consistent baseline for evaluation.

To support high-throughput querying with the aimed at OpenAI’s LLMs, a custom PHP script was developed to manage API communications, enforce model alternation, and log structured responses. The script accepted a structured CSV file as input, containing question IDs, ground-truth answers, and optional hints. It dynamically constructed prompts based on the selected condition (No Hint or With Hint) and used a parameter switch to alternate between GPT-3.5 and GPT-4o. For each query, it executed a POST request to OpenAI’s API, implemented retry logic for handling transient failures, and logged both request and response data in structured JSON format. Metadata such as model version, prompt condition, timestamps, and response tokens were recorded to ensure full reproducibility of the experimental process. This infrastructure was used to query both GPT-3.5 and GPT-4o for all 99 questions under both prompting conditions, producing four sets of model-generated responses.

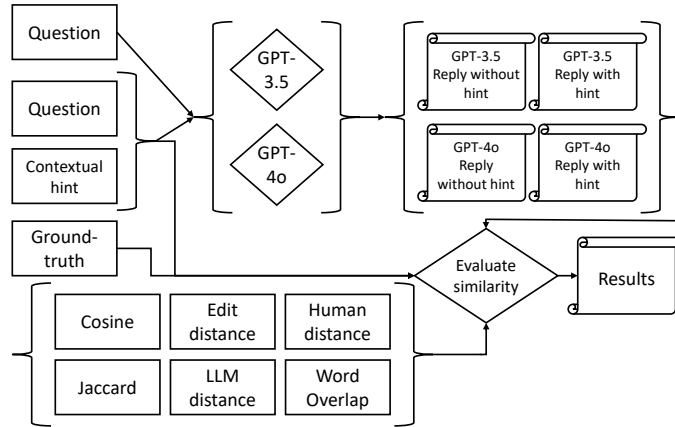
The dataset required several corrective preprocessing steps. These included the standardization of column names (e.g., Answer-ground-truth, gpt3.5withhint), filtering out empty or malformed model responses, and the conversion of inconsistent encoding. Boolean fields indicating correct answers were initially used for exact match scoring but were deemed too restrictive. As a result, we extended our evaluation with four similarity metrics: Cosine Similarity (semantic overlap), Jaccard Similarity (lexical set overlap), Edit Distance (string-based structural similarity), and Word Overlap (non-stopword lexical intersection).

These similarity scores were computed using custom Python scripts, leveraging libraries such as Gensim [33] (for vector representation and semantic similarity), scikit-learn [27] (for TF-IDF modeling and cosine similarity computation), and standard Python routines for string normalization and comparison. Prior to applying the lexical similarity metrics (Jaccard, Edit Distance, Word Overlap), all text responses were tokenized. Tokenization refers to the process of splitting a text string into smaller units called tokens—typically words. This step allows comparisons to be performed on a per-word basis, enabling fair computation of overlap and similarity. For instance, the sentence “The mitochondria is the powerhouse of the cell” is split into tokens such as ‘the’, ‘mitochondria’, ‘is’, ‘the’, ‘powerhouse’, ‘of’, ‘the’, and ‘cell’. Tokenization also serves to normalize word forms and remove punctuation, making the downstream metrics more reliable. In our implementation, tokenization was performed using the standard `word_tokenize` function from the NLTK library in Python [3], which splits text into linguistically meaningful units. This approach was chosen for its robustness

in handling punctuation, contractions, and multilingual content. Cosine Similarity was computed using TF-IDF vectors [28] to measure semantic alignment between model output and ground-truth, capturing shared terms’ relevance rather than just lexical overlap. Jaccard Similarity treated answers as sets of words and calculated the ratio of intersection over union, providing a stricter view of shared vocabulary. Edit Distance Similarity quantified the proportion of shared character sequences through the normalized Levenshtein ratio. Finally, Word Overlap Similarity counted the ratio of shared non-stopword tokens, offering a more lenient lexical alignment perspective. For semantic comparison, vector representations of the text were created using TF-IDF models. Lexical comparisons were implemented with token-based set operations and string matching utilities.

Moreover, and in order to further quantify the similarity of models’ outputs and ground-truth answers, two more similarity metrics were utilized: the “LLM” and “Human”. The “LLM similarity metric” refers to the evaluation of a model’s output and ground-truth answer given a prompt [49] (with or without the contextual hint) by an LLM. The LLM was prompted to include a numerical / percentage of similarity as a reply, in addition to a textual explanation for the evaluation. This process’s aim was to: initially, provide a higher-level of evaluation for models’ outputs and ground-truth answers, and also to test the ability of LLMs in doing so in comparison to human experts. The “Human similarity metric” refers to human intelligence experts that evaluated the the similarity of models’ outputs and ground-truth answers.

The architecture of the full proposed framework is presented in Figure 1.



**Fig. 1.** The architecture of the proposed framework.

To support metric comparability, all similarity values were normalized using min-max scaling prior to aggregation. This standardization allowed us to analyze relative performance differences across models and prompt types using unified scales. The finalized dataset included raw and normalized similarity

scores, ground-truth alignment, model outputs, and metadata for reproducibility and further statistical processing.

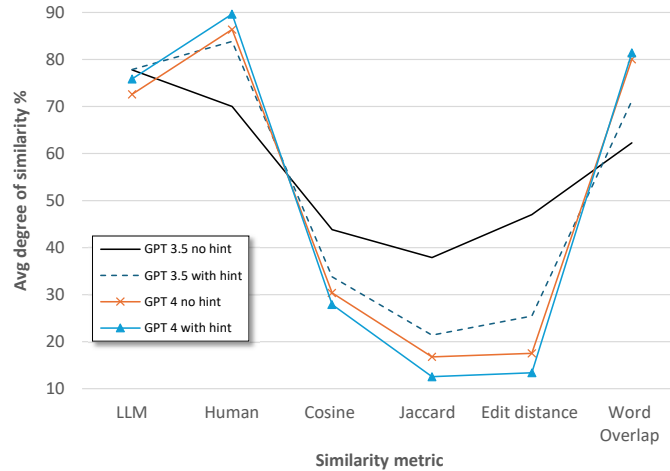
## 4 Experimental Evaluation

### 4.1 Experimental Setup

All algorithms described herein have been implemented using PHP and Python programming languages. The personal computer specifications used for conducting the experiments included an Intel Core i7-6700 CPU 3.40GHz with 16.0 GB RAM, 64-bit operating system and x64-based processor.

In order to support the reproducibility of the experimentation, data samples (due to copyright issues) and programming code for the execution of the experiments presented herein are available in <https://github.com/pgratsanis>.

### 4.2 Evaluation Results



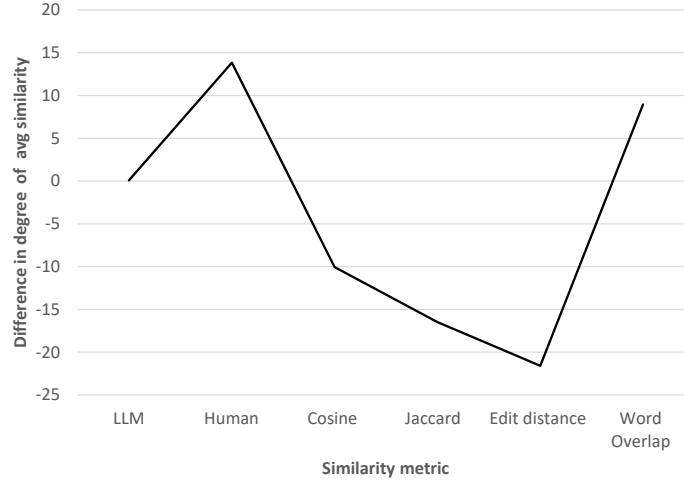
**Fig. 2.** Average similarity scores between model outputs and ground-truth answers for varying LLM models and prompting conditions.

In order to test the effect of contextual hints in LLMs’ prompts, the average similarity scores between model outputs and ground-truth answers under the two prompting conditions, with and without Hint, were calculated. Metrics were averaged across both GPT-3.5 and GPT-4o for a high-level overview. Detailed model- & condition- specific analyses are presented in Figures 2, 3 and 4.

The results received, as shown in Figure 2, indicate a comparable performance across all four combinations of LLM model and contextual condition alternatives. Clearly, all combinations present high degree of similarity for the



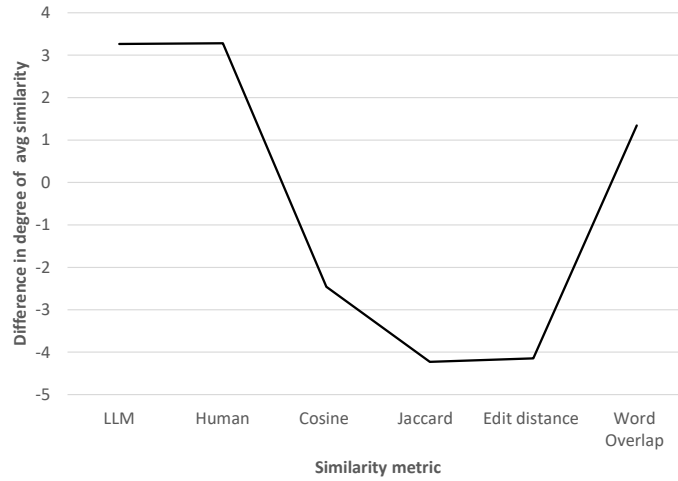
“LLM similarity metric” and “Human similarity metric”, a significant dip in performance for the Cosine, Jaccard and Edit distance measures, and again a sharp rise in performance for the Word Overlap metric.



**Fig. 3.** Comparison between performance without and with contextual hints for GPT-3.5.

This pattern indicates a number of interesting phenomena, namely the effect of LLM model, contextual hints, and similarity measure utilized. The LLM GPT-3.5 presented better performance than GPT-4o as per the Cosine, Jaccard, and Edit distance metrics, marginally better for “LLM metric”, while for metrics “Human”, and Word Overlap clearly worse. GPT-4o presented the best performance of the experimentation for the “Human similarity metric”, though GPT-3.5 with contextual hints was marginally less effective. The metrics Word Overlap and “LLM”, especially for GPT-4o presented comparable performances. GPT-3.5’s performance without contextual hints was significantly worse than with hints for the “Human similarity metric” and Word Overlap, while the opposite for the Cosine, Jaccard, and Edit distance metrics. The performance of GPT-4o with or without contextual hints was closely comparable for all metrics. Accordingly, GPT-3.5 tended to score higher in lexical metrics without hints, while GPT-4o maintained more consistent performance regardless of prompting.

Figure 3 presents a comparison between average performances of the GPT-3.5 LLM model without and with the use of contextual hints, for varying similarity measurements. The results obtained indicate that the existence of contextual hints did not affect the perceived similarity for the “LLM metric” while the opposite was true for the “Human similarity metric” presenting a differentiation of approx. 15 percentage points. The results also make evident that the discrepancy between usage or not of contextual hints is even more significant for the Jaccard and Edit distance measures, reaching a reduction of performance for the use of hints approx. 17 and 22 percentage points, respectively.



**Fig. 4.** Comparison between performance without and with contextual hints for GPT-4o.

Figure 4 presents a comparison of average performances of GPT-4o model without and with the use of contextual hints, again for varying similarity measurements. The results obtained differ significantly from GPT-3.5 LLM mostly in terms of absolute values, while the trend is mostly similar for both models. Herein, the existence of contextual hints marginally affected positively the perceived similarity for the “LLM”, “Human” and Word Overlap metrics while marginally affected negatively for the Cosine, Jaccard, and Edit distance metrics.

### 4.3 Discussion

In general, the results received indicate that hints encourage more detailed and informative answers that may deviate structurally or lexically from the ground-truth, but still retain relevant content. Notably, the drop in Edit Distance and Jaccard similarities suggest increased variation in expression, while the modest drop in Cosine Similarity highlights a more nuanced semantic drift. These outcomes support the hypothesis that traditional alignment-based metrics may not fully capture the benefits of guided prompting. To better understand the role of each similarity metric in this evaluation, we consider their individual behaviors:

**Cosine Similarity** measures semantic alignment based on TF-IDF vector representations. Our results indicate a decline in cosine based similarity when hints are provided. This may suggest that although hinted responses include richer language, these sometimes introduce semantic drifts away from the concise ground-truth phrasing.

**Jaccard Similarity** scores consistently decreased with the use of hints. This metric, sensitive to exact token overlap, penalizes lexical variation. Since hints encouraged verbose and paraphrased responses, exact token overlap diminished.

**Edit Distance Similarity**, based on character-level transformations, also declined with hints. Its sensitivity to syntactic divergence reflects how hinted answers, while potentially more informative, often adopt structurally distinct phrasings compared to the ground-truth.

In contrast to the above, **Word Overlap Similarity**’s performance increased with hints. This suggests that hinted responses, although more lexically diverse, still preserved a larger set of content-relevant non-stopword tokens. This may indicate stronger topical alignment, even if exact structure varies.

Together, these results reveal that traditional lexical metrics may underestimate the value of responses conditioned on hints [36]. Although classic measures penalize paraphrasing or elaboration, such responses may reflect deeper understanding, a distinction more faithfully captured by semantic metrics such as cosine similarity. While hints improve Word Overlap, they tend to reduce semantic and syntactic alignment according to traditional metrics. This suggests that hinted responses, though potentially richer, diverge in form from canonical answers. It also points to the limitations of classic metrics in capturing quality when responses are meaning-equivalent but lexically diverse.

Notably, these trends were observed in both GPT-3.5 and GPT-4o, although preliminary results suggest that GPT-4o maintains higher overall consistency with ground-truth [25], as also supported in OpenAI’s official evaluations, particularly in semantic similarity. Future versions of this study will expand on these inter-model comparisons.

Our findings suggest that hints can shift the model’s reasoning strategy, which allows for a more nuanced understanding of prompt-conditioned generation. This invites further research into context-sensitive evaluation methods that align better with human judgment, and collaboration.

The findings also underscore a broader theoretical implication: classic lexical similarity metrics, though widely used, may systematically undervalue high-quality, paraphrased responses emerging from hints. This reveals a misalignment in metric-based and human-perceived quality and reinforces the need for evaluation frameworks incorporating semantic understanding and context sensitivity.

Beyond general trends, domain-specific linguistic features in Biology, such as polysemous terms (“expression”, “culture”) and hierarchical concepts, were found to affect LLM performance. Models often misinterpreted such terms without hints, while contextual prompts improved alignment with the intended meaning, especially in specialized domains like molecular biology and taxonomy [44]. Hints also supported more structured reasoning in taxonomy and cellular processes. However, some factual errors persisted, suggesting that, while hinting enhances semantic relevance, domain expertise remains a limitation. These observations align with findings from educational evaluations of LLMs, where terminology-heavy biology questions exposed weaknesses in semantic precision and consistency, particularly in high-application tasks [8]. Altogether, they highlight the value of context-aware prompting in terminology-heavy fields.

## 5 Conclusion

Human-AI collaboration across domains such as education, medicine, creative arts, and complex problem-solving, where nuanced interactions significantly influence outcomes, has gained increasing attention. Effective such collaboration relies heavily on the ability of LLMs to accurately interpret and respond to subtle contextual hints provided by human partners. Thus, gaining insights into how these models integrate contextual information into their responses is crucial for enhancing the synergy between human creativity and AI assistance.

Key findings herein indicate that contextual hints influence the complexity and structure of LLMs’ responses. Hinted prompts may produce more detailed and paraphrased outputs, divergent lexically and structurally from ground-truth responses. Traditional similarity metrics often undervalue these richer, contextually-driven answers, highlighting a gap between metric-based assessments and human-perceived quality, particularly in nuanced semantic contexts.

Future work includes exploring the development of advanced evaluation metrics better aligned with human judgment and collaboration, comparative studies across diverse academic and creative domains in order to deepen understanding of how contextual hints can be tailored for effectiveness, and investigation of prompting strategies that may also improve human-AI collaborative processes.

## References

1. Amershi, S., Weld, D., Vorvoreanu, M., Fournery, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P.N., Inkpen, K., Teevan, J., Kikin-Gil, R., Horvitz, E.: Guidelines for human-ai interaction. In: CHI Conference on Human Factors in Computing Systems. p. 1–13. Association for Computing Machinery (2019)
2. Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: ACL Workshop (2005)
3. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O’Reilly Media (2009)
4. Brown, T.B., Mann, B., et al.: Language models are few-shot learners. ArXiv (2020)
5. Chakraborty, D., Das, D., Goldenberg, E., Koucký, M., Saks, M.E.: Approximating edit distance within constant factor in truly sub-quadratic time. IEEE 59th Annual Symposium on Foundations of Computer Science pp. 979–990 (2018)
6. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., Xie, X.: A survey on evaluation of large language models. ACM transactions on intelligent systems and technology **15**(3), 1–45 (2024)
7. Choi, Y., Asif, M.A., Han, Z., Willes, J., Krishnan, R.: Teaching LLMs how to learn with contextual fine-tuning. In: The Thirteenth International Conference on Learning Representations (2025)
8. Dao, X.Q., Le, N.B.: Llms performance on vietnamese high school biology examination. Int. J. Mod. Educ. Comp. Sci **15**, 14–30 (2023)
9. Debnath, T., Siddiky, M.N.A., Rahman, M.E., Das, P., Guha, A.K.: A comprehensive survey of prompt engineering techniques in large language models. TechRxiv (2025)

10. Devlin, J., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
11. Ding, Z.: Advancing gui for generative ai: Charting the design space of human-ai interactions through task creativity and complexity. In: International Conference on Intelligent User Interfaces. p. 140–143. Association for Computing Machinery (2024)
12. Ding, Z., Chan, J.: Mapping the design space of interactions in human-ai text co-creation tasks. ArXiv **abs/2303.06430** (2023)
13. He, Y.Y., Liu, Z., et al.: LLMs meet multimodal generation and editing: A survey. ArXiv **abs/2405.19334** (2024)
14. Hirohashi, Y., Hirakawa, T., Yamashita, T., Fujiyoshi, H.: Prompt learning with one-shot setting based feature space analysis in vision-and-language models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7761–7770 (2024)
15. Kazemi, M., Fatemi, B., Bansal, H., Palowitch, J., Anastasiou, C., Mehta, S.V., Jain, L.K., Aglietti, V., Jindal, D., Chen, P., et al.: Big-bench extra hard. arXiv preprint **arXiv:2502.19187** (2025)
16. Khalid, M., Yousaf, M.M., Sadiq, M.U.: Toward efficient similarity search under edit distance on hybrid architectures. Inf. **13**, 452 (2022)
17. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. ArXiv **abs/2205.11916** (2022)
18. Kolthoff, K., Kretzer, F., Fiebig, L., Bartelt, C., Maedche, A., Ponzetto, S.P.: Zero-shot prompting approaches for llm-based graphical user interface generation. arXiv preprint arXiv:2412.11328 (2024)
19. Liang, P., et al.: Holistic evaluation of language models. arXiv preprint arXiv:2211.09110 (2022)
20. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: ACL Workshop (2004)
21. Liu, N., et al.: Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics **12**, 157–173 (2023)
22. McCauley, S.: Approximate similarity search under edit distance using locality-sensitive hashing. ArXiv **abs/1907.01600** (2019)
23. Metzler, D., Dumais, S., Meek, C.: Similarity measures for short segments of text. In: Amati, G., Carpineto, C., Romano, G. (eds.) Advances in Information Retrieval. pp. 16–27. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
24. Mistry, M., Pavlidis, P.: Gene ontology term overlap as a measure of gene functional similarity. BMC bioinformatics **9**, 1–11 (2008)
25. OpenAI: Gpt-4 technical report. Tech. rep., OpenAI (2023), <https://openai.com/research/gpt-4>
26. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
27. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Louppe, G., Prettenhofer, P., Weiss, R., Weiss, R.J., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
28. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Louppe, G., Prettenhofer, P., Weiss, R., et al.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

29. Pramanick, A., Hou, Y., Mohammad, S., Gurevych, I.: Transforming scholarly landscapes: Influence of large language models on academic fields beyond computer science. arXiv **arXiv:2409.19508** (2024)
30. Qu, J., Jiang, Q., Weng, F., Hong, Z.: A hierarchical clustering based on overlap similarity measure. In: International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. vol. 3, pp. 905–910 (2007)
31. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* (2020)
32. Rajpurkar, P., et al.: Squad: 100,000+ questions for machine comprehension of text. In: *Proceedings of EMNLP* (2016)
33. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: *LREC Workshop on New Challenges for NLP Frameworks*. pp. 45–50 (2010)
34. Reynolds, L., McDonell, K.: Prompt programming for large language models: Beyond the few-shot paradigm. In: *CHI conference on human factors in computing systems*. pp. 1–7 (2021)
35. Schütze, H., Manning, C.D., Raghavan, P.: *Introduction to information retrieval*, vol. 39. Cambridge University Press Cambridge (2008)
36. Sellam, T., et al.: Bleurt: Learning robust metrics for text generation. In: *Proceedings of ACL* (2020)
37. Srivastava, A., et al.: Big-bench: Beyond the imitation game. arXiv preprint arXiv:2206.04615 **abs/2206.04615** (2022)
38. Starr, C., Taggart, R., Evers, C., Starr, L.: *Biology: The Unity and Diversity of Life*
39. Vatsal, S., Dubey, H.: A survey of prompt engineering methods in large language models for different nlp tasks. arXiv **arXiv:2407.12994** (2024)
40. Verma, D.V.K., Aggarwal, R.K.: A comparative analysis of similarity measures akin to the jaccard index in collaborative recommendations: empirical and theoretical perspective. *Social Network Analysis and Mining* **10** (2020)
41. Walia, C.: A dynamic definition of creativity. *Creativity Research Journal* **31**(3), 237–247 (2019)
42. Wang, A., et al.: Glue: A multi-task benchmark and analysis platform for natural language understanding. In: *Proceedings of ICLR* (2018)
43. Wu, Z., Ji, D., Yu, K., Zeng, X., Wu, D., Shidujaman, M.: Ai creativity and the human-ai co-creation model. In: *Human-computer interaction. theory, methods and tools: thematic area*. pp. 171–190 (2021)
44. Zhang, Q., Ding, K., Lyv, T., Wang, X., Yin, Q., Zhang, Y., Yu, J., Wang, Y., Li, X., Xiang, Z., Xiang, Z., Wang, Z., Qin, M., Zhang, M., Zhang, J., Cui, J., Xu, R., Chen, H., Fan, X., Xing, H., Chen, H.: Scientific large language models: A survey on biological & chemical domains. ArXiv **abs/2401.14656** (2024)
45. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. ArXiv **abs/1904.09675** (2019)
46. Zhang, Y., et al.: Contextual prompting for improving factual consistency in text generation. In: *Proceedings of EMNLP* (2022)
47. Zhao, W., et al.: Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In: *Proceedings of EMNLP* (2019)
48. Zhao, Z., Wallace, E., Feng, S., Klein, D., Singh, S.: Calibrate before use: Improving few-shot performance of language models. In: *International conference on machine learning*. pp. 12697–12706 (2021)
49. Zheng, X., Liu, Y., Li, R., Zhang, M.: Can large language models judge their own answers? arXiv preprint arXiv:2305.06358 (2023)