EyeLLM: Using Lookback Fixations to Enhance Human-LLM Alignment for Text Completion

Astha Singh[1], **Mark Torrance**[2], **Evgeny Chukharev**[1]
[1]Iowa State University, [2]Nottingham Trent University
asthas@iastate.edu, mark.torrance@ntu.ac.uk, evgeny@iastate.edu

Abstract

Recent advances in LLMs offer new opportunities for supporting student writing, particularly through real-time, composition-level feedback. However, for such support to be effective, LLMs need to generate text completions that align with the writer's internal representation of their developing message, a representation that is often implicit and difficult to observe. This paper investigates the use of eye-tracking data, specifically lookback fixations during pauses in text production, as a cue to this internal representation. Using eye movement data from students composing texts, we compare human-generated completions with LLM-generated completions based on prompts that either include or exclude words and sentences fixated during pauses. We find that incorporating lookback fixations enhances human-LLM alignment in generating text completions. These results provide empirical support for generating fixation-aware LLM feedback and lay the foundation for future educational tools that deliver real-time, composition-level feedback grounded in writers' attention and cognitive processes.[1]

1 Introduction

Natural language processing (NLP) solutions exist for scaffolding students who are learning to produce effective text. These support both surface-level accuracy (grammar and spelling), and also compositional-level effectiveness, i.e. helping students produce text that communicates a coherent message (e.g., Franzke et al., 2005; Roscoe and McNamara, 2013). Recent advances in large language models (LLMs) enable promising innovative applications for intelligent support of writing tasks. Specifically, there are potential advantages to providing composition-level feedback in real time, while the writer is still forming their message, rather than retrospectively, once their text is complete.

Achieving this is challenging, but is brought within reach by LLMs. These can generate plausible completions to text that a student is in the process of composing. However, providing feedback based on these plausible completions has limited learning benefit unless the LLM-generated completions are aligned with those that the student intended to produce. Human-LLM alignment will increase if the LLM captures important features of the writer's current internal representation of their developing message. However, these mental representations are not directly observable. They are also likely to be implicit, at least in part: The writer might not have explicitly articulated their developing message even in their own internal representation (Torrance, 2016).

One possible clue to this implicit internal representation is provided by writers' eye movement. During text production writers frequently hesitate, often very briefly, and look back within their own text. These lookback eye movements typically involve "hopping around" between isolated words and phrases rather than sustained reading (Chukharev-Hudilainen et al., 2019; Torrance et al., 2016). This eye movement is, however, targeted: Words are not fixated at random, but tend to be informationally rich. Previous work in cognitive psychology has suggested that lookbacks may be driven by the writer's internal representation of the emerging message (Torrance, 2016; Torrance et al., 2016), but this hypothesis has not been systematically evaluated.

In this paper, we propose the use of eye-tracking cues to enhance LLM performance in predicting

[1]Following the initial submission, we discovered an error in one of the analysis scripts that inadvertently introduced data contamination. To ensure the validity of the findings, all models were rerun using corrected code. This version reports the updated results. While specific numerical values have changed, the main conclusions of the study remain unaffected. Our code is available publicly at https://go.chukharev.com/bea-2025.

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 841–849

text completion. If eye fixations cue content for what the writer produces next, then lookback data can help provide completion suggestions that align more closely with the writer's current thinking. To test this hypothesis, we use keystroke and eye movement data from human writers composing argumentative texts. We extract hesitation events: pauses when writers stopped and looked back into their text and then, without editing, continued writing (e.g., finishing the sentence that they were writing before the pause). We compare writers' own completions with LLM completions generated on the basis of prompts that did and did not include the words and sentences that the writers fixated on during lookback. Increased overlap between LLM and writer completions when prompts incorporate information from lookbacks would be evidence for the potential value of eye-movement-informed message-level scaffolding of written composition.

The purpose of this paper is two-fold: First, we evaluate whether the information on the writer's lookback fixations can enhance the alignment between the human and the LLM in the text completion task. Second, we investigate whether LLM text completions with and without eye movement data can provide evidence for the (cognitive) hypothesis about the role of lookbacks in human text production. This lays the necessary groundwork for designing novel educational applications wherein useful composition-level feedback can be provided to students in real time, before the text is completed by the student.

2 Related Work

Functions of reading during writing in humans. Research in cognitive psychology suggests that writers often look back at their own text during pauses in production, particularly near sentence boundaries. These fixations frequently involve lexical processing rather than simple error-checking. Most are not part of sustained reading sequences but instead consist of gaze shifts among isolated words via forward and backward saccades. These lookbacks are thought to support the planning of upcoming text rather than merely identifying mistakes in previously written content (Chukharev-Hudilainen et al., 2019; Torrance et al., 2016).

Human–LLM Alignment. Recent efforts to enhance alignment between humans and large language models (LLMs) in writing support systems have focused on modeling writers' intentions and cognitive states (Zhang et al., 2024; Gero et al., 2022). However, these internal intentions are often difficult to directly observe. Looking back into existing text, in addition to supporting error monitoring, is likely to support ongoing text production, cuing both message and linguistic (lexical, syntactic) form for what the writer will say next (Chukharev-Hudilainen et al., 2019; Torrance et al., 2016). Knowledge of what words and sentences a writer fixates during these lookbacks, therefore, may provides insight into the writer's evolving mental representation of their developing composition.

While some recent approaches have explored aligning LLM-generated suggestions with user intentions (Reza et al., 2025), most have not incorporated real-time behavioral signals. Our work builds on this line of inquiry by explicitly integrating gaze-based cues into prompting strategies, aiming to improve alignment between LLM completions and the writer's unfolding mental model.

Eye-Tracking in NLP. Eye-tracking data has also been leveraged to improve NLP models across a variety of tasks. Prior studies show that incorporating gaze signals can enhance performance in named entity recognition (Hollenstein and Zhang, 2019), text comprehension (Reich et al., 2022), and question answering (Wang et al., 2024). More recently, (López-Cardona et al., 2025) introduced a reward model that uses eye-tracking data to optimize Human–AI alignment. Advances in LLMs have further spurred research into using neural and behavioral signals for better alignment. For instance, (Aw et al., 2023) demonstrate how instruction-tuning can align LLMs with human brain signals. In a similar vein, our study investigates whether reading fixations can serve as meaningful input for improving alignment between LLM text completion responses and writers' cognitive processes.

3 Methodology

3.1 Data

Thirty undergraduate college students (22 women, 8 men, age range 18-22, mean age 19.7 years) composed two texts each using the CyWrite text editor (Chukharev-Hudilainen et al., 2019)[2], while their eye movements were recorded with an SR Research EyeLink 1000 Plus system in a monocular remote setup, calibrated using a 9-point procedure. CyWrite maps on-screen eye fixation coordinates to

[2] https://github.com/chukharev/cywrite

corresponding in-text locations – i.e., the specific words being fixated – accounting for scrolling, line wrapping, and text edits. The writing task appeared as the top paragraph in the editor, with participants composing their responses below it. There was no time limit for the writing tasks, the order of tasks was counterbalanced across participants, and a short break was provided between the two tasks for each participant. Participants were not allowed to consult any external sources. All texts were composed in English, and all participants reported that English was their first language.

CyWrite generates a time-aligned log file that records the timestamp for every key press, key release, and eye fixation. Fixations are classified into sustained reading (operationalized as sequences of at least three consecutive eye fixations on words within the same line of text progressing from left to right) and fixating isolated words (defined as fixations on text that are not part of sustained reading sequences). For this study, we define *hesitations* as pauses between successive keypresses during which the participant engages in sustained reading.

3.2 Language Models

We generate responses for four LLMs, namely, GPT-3.5, GPT-4, LLaMa3-8B and Mistral7B. We use gpt-3.5-turbo (OpenAI, 2023) and gpt-4.1 (Achiam et al., 2023) via the OpenAI API. The exact number of parameters for the GPT models have not been officially disclosed but gpt-3.5-turbo is expected to have approximately 20 billion paramaters (Singh et al., 2023). The responses are generated at a temperature setting of 0.7. We use Llama3-8B and Mistral-7B through ollama (Ollama, 2023). LLaMa3-8B and Mistral-7B have 8 billion and 7 billion parameters, respectively. For all the models, the number of tokens to be generated is dynamically defined to be approximately equal to the number of tokens in the corresponding *completion*.

3.3 Prompt Design

We first create a baseline prompt that consists of the *pretext*, an instructional prompt, and the task description provided to the student. The two task descriptions are:

- Some people have said that finding and implementing green technologies, such as wind or solar power, should be the focus of our efforts to avert climate crisis. To what extent do you agree or disagree with this statement? Try to support your arguments with appropriate evidence from, for example, your knowledge of scientific evidence, your own experience, or your observations and reading.
- Some people have argued that animals should be given similar rights to humans. To what extent do you agree or disagree with this statement? Try to support your arguments with appropriate evidence from, for example, your knowledge of scientific evidence, your own experience, or your observations and reading.

To contrast the LLM responses with fixations against those without fixations, we generate responses for a control condition where along with the baseline prompt we provide the LLM with a matched number of non-fixated non-stop words from the *pretext* (if there are fewer non-fixated words in the *pretext* than fixated words, we include all non-fixated words). Thus, we generate LLM responses in four conditions:

1. **Baseline.** Baseline Prompt only
2. **Words.** Baseline + fixated non-stop words
3. **Sentences.** Baseline + filtered fixated sentences
4. **Control.** Baseline + a matched number of non-fixated words

The prompts for each of the conditions are presented in Table 1.

3.4 Evaluation Metrics

To evaluate the performance of LLMs with and without fixations, we establish similarity between human and LLM-generated completions in each of the four conditions on both semantic and token-based (surface linguistic form) measures.

3.4.1 Semantic Similarity

We quantify the semantic similarity between human and LLM responses by computing the cosine similarity between embedding vectors generated from the completions using the text-embedding-ada-002 model via OpenAI API (OpenAI, 2024). This approach captures global semantic alignment between the different completions.

3.4.2 Token-based Similarity

We calculate two token-based similarity metrics: F1 Score and Jaccard Index.

Condition	Prompt
Baseline	This was the task description provided to a student: <task_description >. Please write a continuation of: <pretext >.
Words	This was the task description provided to a student: <task_description >. We have identified the following key words as particularly important: <fixated words >. Please write a continuation of: <pretext >.
Sentences	This was the task description provided to a student: <task_description >. We have identified the following sentences as particularly important: <fixated sentences >. Please write a continuation of: <pretext >.
Control	This was the task description provided to a student: <task_description >. We have identified the following key words as particularly important: <non-fixated words >. Please write a continuation of: <pretext >.

Table 1: Prompt for each condition

F1 Score: F1 score accounts for both precision and recall, making it useful for evaluating word overlap between the two texts. Precision measures the proportion of shared words in the second text (W_2), while recall measures the proportion of shared words in the first text (W_1).

$$\text{Precision} = \frac{|W_1 \cap W_2|}{|W_2|}$$

$$\text{Recall} = \frac{|W_1 \cap W_2|}{|W_1|}$$

The **F1 Score**, which balances precision and recall, is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Jaccard Index: Jaccard Index is a set-based measure that quantifies the overlap between two texts by comparing the size of their intersection with their union. This metric focuses on shared words without considering their relative frequency. It is defined as:

$$\text{Jaccard Similarity} = \frac{|W_1 \cap W_2|}{|W_1 \cup W_2|}$$

4 Approach

In this section, we present our approach to data extraction and LLM response generation. The approach is outlined in Figure 1.

4.1 Extract Hesitation Events

The first step in our approach is to obtain valid hesitation events from human text production data. We defined *hesitations* as inter-keypress intervals where writing is interrupted by a pause, during which the writer engages in sustained reading. At the time of hesitation, we extract the span of text between the start of the current paragraph and the cursor location. We call this the *pretext*. In Figure 2, | represents the cursor location at the time of hesitation. We discard all hesitations where the pretext is empty.

Once we have a valid hesitation, we traverse the log file to extract the human *completion* of the pretext. To this end, we consider all keystrokes following the hesitation until the writer types a sentence-final punctuation symbol (./?/!). The completion is valid so long as the writer does not edit or delete any portion of the pretext at any point during the completion process. Valid human completions serve as the gold standard for comparison against the LLM-generated completions. However, we discard all invalid completions. We define a *hesitation event* as a valid hesitation followed by a valid completion. The process of extracting hesitation events is outlined in Algorithm 1.

Algorithm 1 Extract Hesitation Events

```
function GetHesitations(data)
    for i in data do
        if len(data[i].pretext) > 0 then
            if sustained reading in data[i] then
                for j from i+1 to len(data) do
                    if data[j] starts data[i] then
                        if data ends in {.,?,!} then
                            Extract e_n
```

4.2 Extract Fixations

Once we have a set of valid hesitation events e_n(pretext, completion), we extract, from the available eye-tracking data, all eye fixations on the text that occurred during each hesitation event. We include both sustained reading fixations, and fixations on isolated words. We apply the following

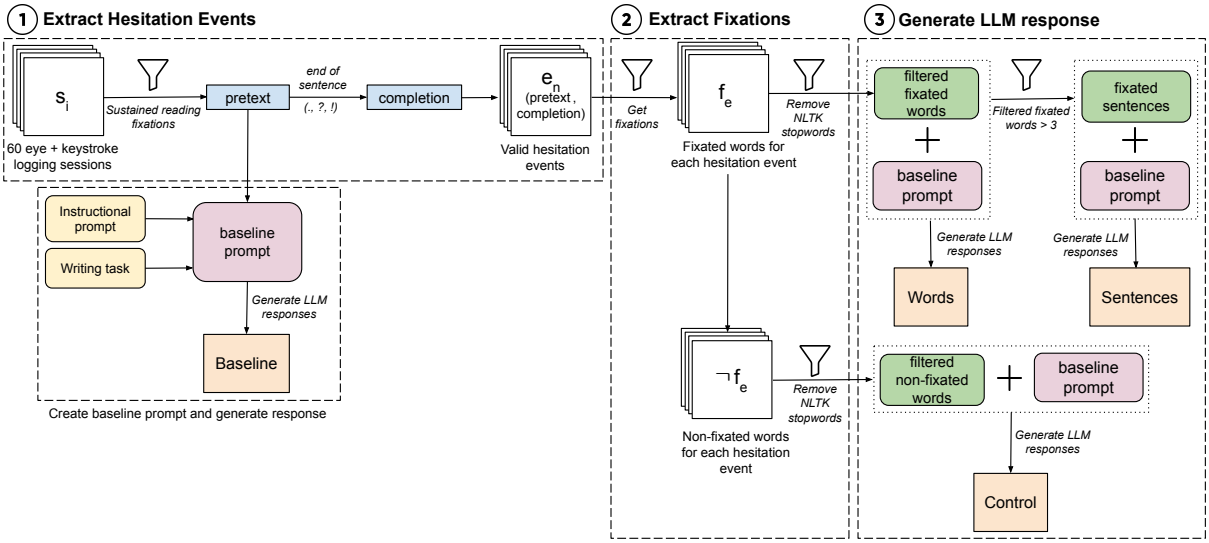
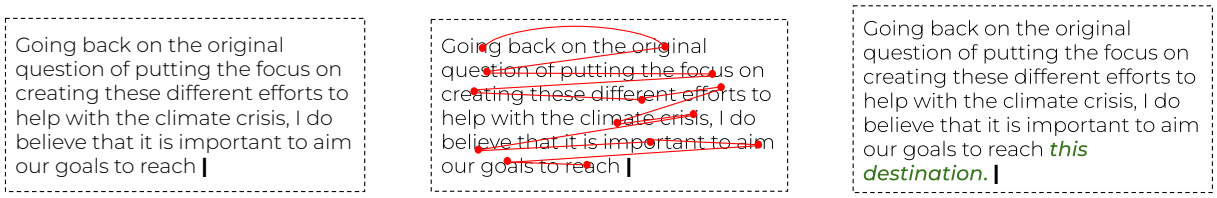


Figure 1: Overview of EyeLLM Approach



(a) (b) (c)

Figure 2: Example showing the extraction of pretext, fixations, and completion. (a) The writer pauses (hesitates) during text production. | indicates their cursor location at the point of hesitation. Everything between the start of the current paragraph and | is the *pretext*. (b) The writer fixates on the highlighted points as shown in the scanpath. Words containing eye fixations are *fixated words*. (c) The writer continues to produce text (highlighted in green). This is the *completion*.

Model	Similarity Scores											
	Semantic				F1				Jaccard			
	Control	Baseline	Words	Sentences	Control	Baseline	Words	Sentences	Control	Baseline	Words	Sentences
GPT-3.5	.8075	.8066	.8086*	**.8090***	.1483	.1484	**.1511***	.1480	.0823	.0824	**.0841***	.0821
GPT-4	.8078	.8056*	.8089	**.8101***	.1332	.1336	**.1347**	**.1347**	.0732	.0735	.0742	**.0744**
LLaMa3	.7945	.7958*	.7958	**.7959**	.1356	.1360	**.1364**	.1360	.0749	.0752	**.0754**	.0753
Mistral7B	.7935	.7919*	**.7950***	.7945	.1228	.1244	**.1262***	.1236	.0673	.0682	**.0693***	.0678

Table 2: Average similarity scores across all LLMs. The highest score for each LLM is highlighted in **bold**. * marks scores that are significantly different from Control, $p < 0.01$.

filtering criteria:

1. We only include lookback fixations, i.e. fixations on the text before the cursor at the time of the hesitation.

2. We exclude fixations on words from the NLTK list of stop words, to ensure that only fixations on semantically important words are included.

3. For the Sentences condition, we identify *fixated sentences* as sentences that contain valid fixations on at least three words.

The process of extracting fixations is outlined in Algorithm 2. We only consider *hesitation events* that have at least 1 fixated word and at least 1 fixated sentence. After this filtering process, we get 822 valid hesitation events. The mean number of fixated words across all valid hesitation events is 11.11 (median 8), and the mean number of fixated sentences is 1.90 (median 1).

Algorithm 2 Extracting Fixations

```
function GetFixations(data, hesitation_events, n)
    for hesitation in hesitation_events do
        Extract fixations for current hesitation
        Remove fixations on stop words
        Store sentences with ≥ n fixated words
        if valid fixations found then
            Append to results
        end if
    end for
    return results
```

4.3 Response Generation

After extracting fixation data for all hesitation events, we prompt several LLMs to generate completions. The full experimental setup is already described in Section 3.

5 Results

We run each model for 10 iterations. The results for all the models averaged over all iterations are presented in Table 2. We answer the following two research questions:

- **RQ1:** Does incorporating information about a writer's lookback fixations improve the alignment between human and LLM-generated text completions?

- **RQ2:** How do different LLMs compare across conditions with and without lookback fixations?

5.1 RQ1: Impact of Lookback Fixations on the Similarity Scores

We perform inferential hypothesis testing to assess whether prompting condition had a statistically significant effect on similarity scores. In our analysis, we treat each similarity measure as a dependent variable. We fit linear mixed effects models (LMERs) with prompting condition (Control, Baseline, Words, Sentences) as the fixed factor. As detailed above, we generate completions 10 times for each *hesitation event*. LMERs therefore include random by-event intercepts and slopes for prompting condition.

First, we perform the analysis separately for each LLM. We fit LMERs for each measure (semantic similarity, F1, Jaccard), resulting in a total of 12 series of nested LMERs. In each series, we first fit an intercept-only model (M_0), and then add the prompting condition fixed effect (M_1). We compare model fit using the likelihood ratio test. We adopt a conservative significance threshold $p < .01$ to guard against Type I errors. When M_1 significantly improves model fit over M_0, we evaluate the fixed-effect coefficients in M_1 to determine which prompting conditions show significant differences from the Control.

The results are shown in Table 2 and Figure 3. As expected, the Control condition does not outperform Baseline for F1 and Jaccard scores. For semantic similarity, however, Control provides significant performance gains over Baseline for GPT-4 and Mistral7B. This suggests that providing additional input to the LLM (even if it is unrelated to the eye-tracking signal) can improve human-LLM alignment in text completion.

Crucially, introducing eye-tracking signals (via Words and Sentences conditions) yields modest but statistically significant improvements over Control in all LLMs except LLaMa3. In terms of semantic similarity, Sentences generally outperform Words, except in Mistral7B. For token-based metrics (F1 and Jaccard), Words tend to perform better than Sentences, with GPT-4 being the exception.

To assess the overall effect of prompting condition across LLMs, we examine differences of average similarity scores between Control and other conditions (Table 3). To test for significance of these differences, we fit one LMER per similarity metric using data from all LLMs, treating LLM as a fixed effect, and including its interaction with prompting condition. Due to LMER convergence

issues, we simplify the random effects structure by removing the random by-event slopes. We then perform Tukey-adjusted pairwise comparisons of estimated marginal means across prompting conditions. We find that, for semantic similarity, all pairwise differences across conditions are statistically significant ($p < .0001$) except between Sentences and Words ($p = .426$). For F1 and Jaccard, Words significantly outperform all other conditions ($p < .01$), while differences among the remaining conditions are not significant ($p > .15$).

These findings support our hypothesis that including fixation-based information in prompts improves human–LLM alignment. Although LLM responses vary in sensitivity to the eye-tracking signal, overall we find that providing LLMs with fixated sentences enhances semantic alignment, while providing fixated words enhances both semantic and token-level alignment with human text completions.

Metric		**Change relative to Control**		
	Control	**Baseline**	**Words**	**Sentences**
Semantic	.8008	-.0008*	**+.0013***	**+.0016***
F1 Score	.1350	**+.0006**	**+.0021***	**+.0006**
Jaccard	.0744	**+.0004**	**+.0013***	**+.0005**

Table 3: Performance changes across prompting conditions, relative to the Control. Bold values indicate improvements. * indicates significant change ($p < .01$). Averages computed across all models.

5.2 RQ2: Differences among LLMs in Performance Across Prompting Conditions

To assess whether differences between LLMs are statistically significant, we extend the inferential tests from Section 5.1 by fitting a series of nested linear mixed-effects models (LMERs) for each similarity measure, using data from all four LLMs. We begin with a baseline intercept-only model (M_0), then sequentially add the fixed effect for prompting condition (M_1), the fixed effect for LLM (M_2), and finally the interaction between condition and LLM (M_3). Due to convergence issues, we remove by-event slopes from the random effects structure.

Model comparisons are conducted using likelihood ratio tests. For semantic similarity, successive models significantly improve the fit ($M_3 > M_2 > M_1 > M_0$, all $p < .0001$). The significant interaction term in M_3 for semantic similarity indicates that the effect of prompting condition varies by LLM–that is, not only do the LLMs differ overall, but the way they respond to different prompting conditions also differs significantly, but only with respect to the semantic metric. On the other hand, for Jaccard and F1, M_3 does not provide further improvement of model fit over M_2 ($p = .018; p = .044$, respectively). This suggests no evidence for the condition-LLM interaction for the token-based metrics.

Pairwise comparisons between LLMs reveal significant differences throughout (all $p < .0001$ with Tukey adjustment), with the exception of the difference between GPT-3.5 and GPT-4 for semantic similarity ($p = .157$).

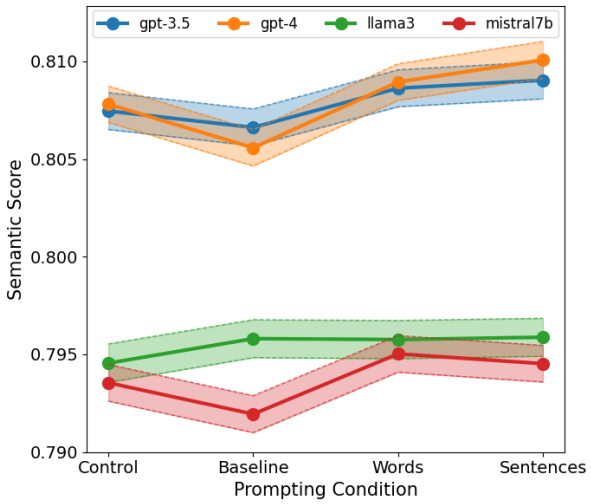
We present an overview of descriptive statistics for different similarity measures below.

Semantic Similarity: As shown in Figure 3a, GPT models consistently outperform LLaMa and Mistral in the semantic alignment across all conditions. With eye-tracking cues, all models show small relative improvements compared to the Control condition (between +.13% and +.29%), but only some of these improvements are statistically significant.

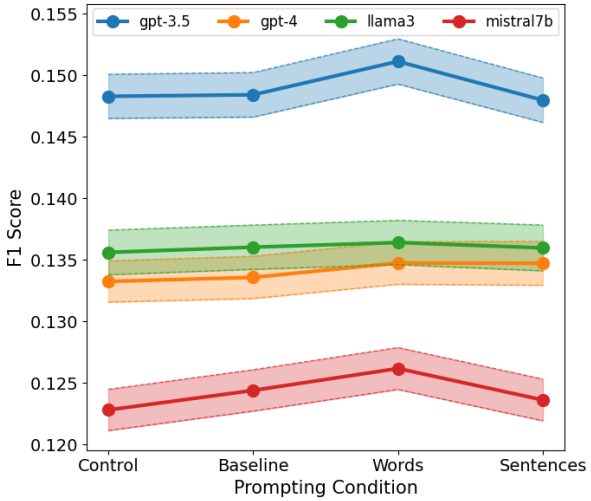
F1 Score: Figure 3b presents model comparisons based on the F1 score. Interestingly, GPT-3.5 outperforms all other models across all prompting conditions, showing the strongest token-level alignment with human completions. GPT-4 and LLaMa3 are closely comparable, while Mistral7B consistently underperforms relative to other models. Across prompting conditions, F1 score changes show greater variability. From Control to Words, only Mistral7B and GPT-3.5 show significant improvement (by +2.77% and +1.89%, respectively). GPT-4 and LLaMa3 show smaller improvements that do not reach significance threshold. All changes from Control to Sentences (ranging from +1.13% to -0.98%) are not statistically significant.

Jaccard Index: As shown in Figure 3c, Jaccard scores generally follow trends seen in F1 scores. From Control to Words, all models improve (between +0.67% and +2.97%), but only GPT-3.5 and Mistral7B show significant improvements. The shift to Sentences shows mixed changes (between +1.64% and -0.74%), none of which reach significance.

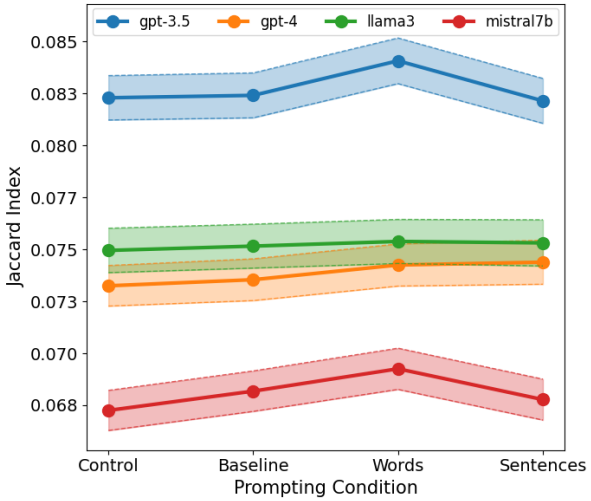
Overall, both GPT models show greatest semantic alignment with student text completions, while



(a) Semantic Score



(b) F1 Score



(c) Jaccard Index

Figure 3: Scores of different models across prompting conditions with 95% confidence intervals.

GPT-3.5 clearly leads on token-based similarity metrics. Eye-tracking cues are not sufficient to significantly change the relative performance of any two LLMs on any of the measures investigated.

6 Summary and Conclusion

To our knowledge, this paper is the first to investigate the impact of word- and sentence-level lookback fixation signal captured during writing pauses on LLM-generated text completions.

We first asked whether the eye-tracking cues improve the human-LLM alignment in the text completion task. By comparing different prompting conditions, we demonstrated that the addition of both the words and the sentences that a writer fixates resulted in small, but statistically significant improvement in the semantic alignment between LLM-generated text completions and what the writer themselves actually wrote. Adding fixated words (but not sentences) improves performance on token-based similarity metrics. This provides tentative (but, to date, best-available) evidence of the role of lookback in text planning and, again tentatively, suggests value in incorporating lookback data in intelligent, real-time tools for supporting and training written composition skills.

We then asked how different LLMs compare across prompting conditions. We found that GPT models outperform smaller open-source models on semantic metrics, while GPT-3.5 offers substantial advantages in token-based similarity. For semantic (but not token-based) metrics, significant statistical interaction between LLM and prompting condition suggests that different LLMs react differently to the eye-tracking signal.

Relative performance gains, while statistically significant, were small (in single-digit percent) across LLMs and similarity metrics. It remains to be seen whether improvements on this scale have practical value for developing educational technologies that support written composition. At the very least, they highlight the need for further research into how lookback information can be used to refine prompts.

Limitations

One limitation of our study lies in the scope of the data collected, which includes responses from 30 students. Nonetheless, we extracted 822 valid hesitation events across 60 composition sessions, reinforcing the robustness of our findings. Secondly,

while the variation in scores across conditions is statistically significant, it is relatively small and its practical significance will depend on the use case. Lastly, we do not present a complete end-to-end tool for providing LLM-generated writing assistance. However, this work establishes a strong foundation for future development in that direction.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. 2016868 and 2302644. We gratefully acknowledge the insightful comments and constructive feedback from the reviewers, which significantly contributed to improving the quality of this paper. We are grateful to Dr. Emily Dux Speltz, Zoë DeKruif, and Jamie Smith for their assistance with data collection from human participants.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Khai Loong Aw, Syrielle Montariol, Badr AlKhamissi, Martin Schrimpf, and Antoine Bosselut. 2023. Instruction-tuning aligns llms to the human brain. *arXiv preprint arXiv:2312.00575*.

Evgeny Chukharev-Hudilainen, Aysel Saricaoglu, Mark Torrance, and Hui-Hsien Feng. 2019. Combined Deployable Keystroke Logging and Eyetracking for Investigating L2 Writing Fluency. *Studies in Second Language Acquisition*, 41(3):583–604.

Marita Franzke, Eileen Kintsch, Donna Caccamise, Nina Johnson, and Scott Dooley. 2005. Summary street®: Computer support for comprehension and writing. *Journal of Educational Computing Research*, 33(1):53–80.

Katy Gero, Alex Calderwood, Charlotte Li, and Lydia Chilton. 2022. A design space for writing support tools using a cognitive process model of writing. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 11–24, Dublin, Ireland. Association for Computational Linguistics.

Nora Hollenstein and Ce Zhang. 2019. Entity recognition at first sight: Improving NER with eye movement information. 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 1–10, Minneapolis, Minnesota. Association for Computational Linguistics.

Ángela López-Cardona, Carlos Segura, Alexandros Karatzoglou, Sergi Abadal, and Ioannis Arapakis. 2025. Seeing eye to ai: Human alignment via gaze-based response rewards for large language models. In *International Conference on Learning Representations (ICLR)*.

Ollama. 2023. Ollama: Run llms locally. Accessed 2025.

OpenAI. 2023. gpt-3.5-turbo-0613 announcement. Function calling, 16k context window, and lower prices.

OpenAI. 2024. Openai api. Used to generate text embeddings via the OpenAI API.

David Robert Reich, Paul Prasse, Chiara Tschirner, Patrick Haller, Frank Goldhammer, and Lena A. Jäger. 2022. Inferring native and non-native human reading comprehension and subjective text difficulty from scanpaths in reading. In *2022 Symposium on Eye Tracking Research and Applications*, New York, NY, USA. Association for Computing Machinery.

Mohi Reza, Jeb Thomas-Mitchell, Peter Dushniku, Nathan Laundry, Joseph Jay Williams, and Anastasia Kuzminykh. 2025. Co-writing with ai, on human terms: Aligning research with user demands across the writing process. *Preprint*, arXiv:2504.12488.

Rod D. Roscoe and Danielle S. McNamara. 2013. Writing pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*, 105(4):1010–1025.

Mukul Singh, José Cambronero, Sumit Gulwani, Vu Le, Carina Negreanu, and Gust Verbruggen. 2023. Codefusion: A pre-trained diffusion model for code generation. *Preprint*, arXiv:2310.17680.

Mark Torrance. 2016. Understanding planning in text production. *Handbook of writing research*, 2:72–87.

Mark Torrance, Roger Johansson, Victoria Johansson, and Åsa Wengelin. 2016. Reading during the composition of multi-sentence texts: an eye-movement study. *Psychological Research*, 80(5):729–743.

Bingbing Wang, Bin Liang, Lanjun Zhou, and Ruifeng Xu. 2024. Gaze-infused bert: Do human gaze signals help pre-trained language models? *Neural Computing and Applications*, 36(20):12461–12482.

Shuo Zhang, Liangming Pan, Junzhou Zhao, and William Yang Wang. 2024. The knowledge alignment problem: Bridging human and external knowledge for large language models. *Preprint*, arXiv:2305.13669.