Investigating the Impact of Prompt Characteristics on Large Language Model Inference Time: A Correlational Study

Alisher Kabardiyadi, Askar Akhmetkhanov, Kira Strelnikova, and Nikolai Petukhov

Innopolis University

May 14, 2025

I Introduction

Large Language Models (LLMs) have developed rapidly over the last few years, transforming various fields, from natural language processing and content generation to customer support. As demand and the variety of applicable areas grew, so did workload, so it is crucial to comprehend the aspects that influence reaction times.

One key factor is the linguistic composition of the prompt: length, syntactic constructions, semantic load, semantic ambiguity, and grammatical structures. Yet, we did not find any work that systematically measured how different text-level features impact on inference latency. In addition, the influence of various linguistic factors is not yet well understood, specifically since the data may vary depending on the particular model and query language.

This study aims to measure specific textual qualities and apply the resulting prompts to the Qwen 2.5 model to examine correlations with inference time. The analysis focuses on metrics such as prompt length, syntactic complexity, and semantic ambiguity because these factors may influence processing efficiency. Controlled experiments will obtain empirical data and statistical analysis will determine the strength and nature of these relationships.

The study helps to determine the most resource-consuming points of existing LLM models. Moreover, the research may become a basis for LLM's inference-time forecasting. This will exhibit the behavior of the model, making the user experience more pleasant and enhancing the overall understanding of the process by the user.

In general, this research pursues the goal of bridging the gap between linguistic complexity and computational performance, offering guidance not only for developers but also for researchers who are working with large-scale language models.

II Literature Review

Large Language Models (LLMs) have demonstrated significant capabilities in domains such as medicine [1], education [2], and scientific analysis [3]. Despite these advances, practical deployment often encounters a key limitation: inference time, or the latency between prompt input and response generation. The quality and coherence of LLM outputs have been the subject of several research, but the computing cost—specifically, how quick properties affect inference duration—remains largely unexplored in scholarly debate.

This work synthesizes recent research in the domains of prompt engineering, linguistic complexity measurements, and natural language processing (NLP) methods to offer a framework for investigating prompt-based implications on model latency. A notable gap that this work attempts to fill is that, whereas prior research has focused on features of input design and response quality, few have directly investigated the influence of language structure on computing efficiency.

Despite extensive work on prompt design and linguistic complexity, we were unable to find appropriate works that would have empirically linked prompt characteristics with LLM inference time. Considering the increasing demand for low-latency or real-time LLM applications in academic and business settings, this oversight is particularly significant.

This study aims to close that gap by:

1. defining measurable prompt complexity criteria based on the body of current work;

2. examining their relationship to LLM inference latency; and

3. providing empirical insights to guide effective prompt design techniques,

This section is structured around the following pillars: prompt engineering, which demonstrates how input structuring influences model output; text complexity metrics, which provide quantifiable tools to characterize linguistic features of prompts. Thus, we will contextualize why specific prompt characteristics might systematically affect LLM processing time, thereby justifying the need for an empirical study of latency drivers.

A. Prompt Engineering: Techniques and Gaps

Prompt engineering has emerged as a pivotal tool for guiding LLM output quality. A number of studies have evaluated the influence of prompt structure on task performance, yet few have examined associated computational costs.

Using a range of role-based prompts, Nahass et al. [1] assessed ChatGPT-3.5 in a domain-specific context on medical board exams. Tailoring encourages improved answer accuracy, according to their research. However, the computational efficiency was not evaluated. Liu et al. [3] noted that prompts incorporating domain-specific chemical information improved accuracy and reduced hallucinations, but failed to address latency concerns.

Jung et al. [2] found that shorter prompts led to less comprehensive replies in medical contexts. The study suggests a trade-off between quick complexity and output quality, which may also apply to inference time. However, this aspect has yet to be investigated.

Garg et al. [4] found that educating students in structured prompting methods, such as few-shot learning, chain-of-thought, and the CLEAR framework, increased their performance in data analysis tasks. Although helpful, the study could not quantify the processing cost caused by more organized suggestions.

Tania et al. [5] discovered that multi-stage prompting produced more precise ecosystem representations than iterative refinement methods in business intelligence. However, the emphasis remained on response quality rather than model delay.

These studies show that timely structure has a significant impact on response effectiveness across professions. The lack of study on how these features affect inference time highlights the uniqueness and significance of the current findings.

B. Text Complexity Metrics

Established metrics from computational linguistics are used into this work to investigate the potential impact of quick qualities on processing delay. Lexical, syntactic, and semantic complexity measurements are among them; they provide measurable markers of text difficulty, which are expected to affect model calculation time.

Lexical complexity is associated with ease of processing for both humans and machines. Van Heuven et al. [6] developed the SUBTLEX-UK corpus, a subtitle-based word frequency dataset that better predicts lexical decision times than standard corpora. They also proposed the Zipf scale, which normalizes frequency measurements in logarithmic steps. Words with greater Zipf values (>4) are often processed faster, indicating that uncommon or domain-specific phrases (Zipf <3) may increase LLM computational load.

Syntax is a significant factor in determining processing difficulty. Szmrecsanyi [7] identified syntactic complexity measures, such as sentence count and syntactic embedding, which indicate that formal texts have more structural density than conversational texts.

Liu et al. [8] improved relationship extracting by means of dependent tree analysis. This work argues that more complex syntactic structures could need more processing resources and hence dependency tree depth is a crucial parameter.

Semantic Complexity: The depth and ambiguity of meaning that a text conveys are aspects of semantic complexity. Indirect indicators of semantic load are provided via readability formulae. The Flesch-Kincaid Reading Ease score was created by Kincaid et al. [9] and uses sentence and syllable length to measure text difficulty. By adding the New English Readability Formula (NERF), Lee and Lee [10] improved this model's sensitivity to contemporary lexical use.

Tseng et al. [11] examined semantic density in LLMs through an information-theoretic lens to enhance current methodologies. Their research demonstrated that semantically dense inputs can elevate processing demands, despite the absence of specific measurements for inference time.

C. Conclusion

The results of readability, computational linguistics, and prompt engineering have been combined in this paper to offer a theoretical framework for investigating the computing cost of LLM rapid design. The aspect of inference time is still mostly ignored, despite the literature's wealth of information on timely optimization for relevance and accuracy.

The current study offers a fresh view on the quick efficiency by combining measurements of lexical, syntactic, and semantic complexity. These results are intended to assist academics and practitioners in creating quick techniques that improve response quality and maximize performance for applications that are sensitive to latency.

III Methodology

We have considered several metrics to measure different aspects of text complexity. To test different metrics, we will use the BAAI/Infinity-Instruct dataset with 30 000 entities, where each entity is a dialog with LLM. We extracted user requests; the total number of prompts resulted in approximately 51 000. Then we applied a filter to leave messages only in English, since some metrics cannot recognize other languages. The final size of the dataset was 39564. We had to shrink it to 1000 prompts, since conducting this experiment with the full size would take approximately 13 days. The duration of such experiement would increase the chance of unpredictable results that are caused by network, software, or hardware issues.

We used Qwen 2.5 72b, the LLM deployed on local servers, to find a correlation between the actual response time and the metrics. We also conducted two different tests with various LLM response size restrictions (16 and limitless).

The complexity of a text can be measured with respect to different classes of metrics. These are lexical, syntactic, semantic, grammatical correctness, readability score. Each of these classes is represented by several metrics. Comparison between different classes and metrics is below.

A. Lexical metrics

This class of metrics is based on the complexity of independent words without considering context. This complexity can be measured by computing word frequency [12]. The metric proposes that a rarer word tends to be more difficult.

We normalize the finite value by dividing it by the test size to exclude the influence of large prompts that consist of a tremendous number of words, which in turn increases the frequency score even with common words dramatically. Moreover, the presented dataset contained several test with enormously frequent usage of specific rare words (for instance, "Input", "range", "Haskell", and others in an Informatics task that do not appear in common texts often). To overcome the problem of the imbalanced result, we applied in addition logarithmical normalization.

In addition, we experimented with the exclusion of stop words for the same prompts, but the results occurred to be approximately the same. All messages on average had relatively the same number of common stop words, which do not lead to any new relation or dependency.

In this work, we use the wordfreq library as a proxy for lexical difficulty. We hypothesize that lower-frequency phrases may take longer to infer due to reduced representation in training datasets.

B. Syntactic metrics

Syntactic complexity is sensible to the structure of sentences that are presented in a prompt. A metric based on this complexity type is one of the most essential to measure text difficulty [13]. In prompts, it reflects the grammatical relationships and hierarchical parsing demands, which affect attention mechanisms and working memory in LLMs.

Syntactic complexity can be measured with one of the listed metrics:

1. Number of clauses
2. Depth of the dependency tree [14]
3. Number of words and their length [7]

A clause is an inalienable component of each sentence, the syntactical complexity of it depends on the number of clauses inside it. The graph of the applying the metric is pictured below. All values are normalized with the number of sentences in a prompt.

The most complex cases are prompts with long sentences that have different clauses connected by "and", "that", "because" etc. More clauses (e.g., compound or nested sentences) require deeper contextual understanding. For example, "If X, then Y, but when Z..." demands tracking multiple dependencies.

The computation of an average depth of dependency trees gives an image presented further.

In this case, complex samples are texts with enumeration of different words. We use Dependency Trees because they show how words relate to each other in a sentence, helping us to measure how complex the structure is.

Capturing Hierarchy: A dependency tree shows how each word links to another, forming a clear hierarchy. Deeper trees indicate more layers of structure, so reflecting greater syntactic complexity.

Measuring Structure: Analyzing these links helps us to determine whether a sentence consists in many dependent clauses or complex phrases. This helps identify complex patterns in the syntax.

Practical and Quantifiable: Counting tree depth is a direct way to compare sentences. It's easier to quantify complexity by focusing on how words connect rather than relying on abstract grammatical rules.

As one of the most straightforward metrics, we calculated number of words in each sentence and results are presented below as well.

We employ three syntactic indicators, including total word count, clause count, and dependency tree depth. Each captures a distinct dimension of structural complexity, which may affect inference time.

C. Semantic metrics

The meaning and conceptual difficulty of the prompt (e.g., ambiguity, abstractness) are covered with the use of semantic complexity. We evaluated semantic complexity by counting polysemous words. Such words with multiple meanings (e.g., "bank," "crane") require disambiguation. This appends cognitive load, as the model must weigh contextual cues to select the correct sense. This metric could highlight how LLMs would be able to resolve contextual or conceptual challenges, which could slow inference due to iterative reasoning.

All of these look ambiguous, which confirmed the proper work of the metric. Polysemy may trigger conflicting attention patterns across layers, prolonging convergence to a stable output.

D. Grammatical correctness

For grammar check we used LanguageTool library. Grammar check graph is not fully correct since it counts as incorrect: foreign names, brand names, mathematical variables and consecutive spaces. Therefore, the result on the graph is a mix of real grammar errors and not real grammar errors.

Moreover, we proposed a complex metric that aimed to use a part of speech (PoS) distribution to distinguish texts with an either plentiful or meager usage of some parts of speech. We had computed in advance average PoS distribution, then calculated for a single entity how differ is it from the average. The results are presented below.

To verify the proper measurement, we printed five metrics with the highest score i.e.

with the largest deviation from the average.

- ? x 120 = 173 x 240
- hi
- ha
- Yes, please!
- yes?

All of the above contain none of typical and common PoS as nouns, verbs, or adjectives.

E. Readability score

Readability metrics are significant to our investigation because they quantify how easily text can be understood, revealing linguistic complexity that may affect both human comprehension and LLM processing efficiency. Each of the metrics has a formula to calculate its readability score.

Readability can be measured in the following ways:

Flesch–Kincaid Grade Level

Evaluates text based on:

- Average sentence length (in words)
- Average word length (in syllables)

This score correlates with the educational grade level needed to comprehend the text

.

Gunning Fog Index

Measures text complexity by considering:

- Average sentence length

- Percentage of complex (multi-syllabic) words

The resulting index denotes the years of formal education required to comprehend the book upon initial reading.

Dale-Chall Readability Score

Uses a predetermined list of “familiar words”.

Texts containing words not on this list receive higher complexity ratings.

The score reflects how easily a passage can be read by the average American student at specific grade levels.

We selected the Flesch–Kincaid Grade Level as our main readability metric because, unlike the Dale-Chall Readability Score, which relies on a fixed vocabulary list, the Flesch–Kincaid Grade Level directly measures the effect of sentence length and word complexity on LLM performance, making it easier to interpret and more flexible.

Because the Gunning-Fog can overestimate difficulty and create ambiguity by treating all multisyllabic words as complex, even if they are widely understood, we also chose not to use it. Furthermore, the Gunning-Fog index is less flexible than the Flesch-Kincaid in our study because it is primarily tailored to English, and short but uncommon words can still present comprehension difficulties.

TABLE 1
Overview of All Evaluated Metrics

Metrics	**Class**	**Select/decline reason**	**Chosen**
Word-frequency list	Lexical complexity	Rare tokens hypothesized to slow inference	✓
Identify stop words	Lexical complexity	Gave no extra dependencies compared to the previous one	✗
Number of clauses	Syntactic complexity	More clauses mean more comprehensive sentences, what leads to a higher parsing load on LLM	✓
Dependency tree depth	Syntactic complexity	Deeper trees reflect more complexity	✓
Words number	Syntactic complexity	Long prompts obviously (initial hypothesis) increase processing time	✓
Polysemous words	Semantic complexity	Multiple senses force ambiguity, could overload an LLM	✓
Checking errors	Grammatical correctness	Grammar mistakes include spaces, names, foreign words what makes it invalid	✗
Part of speech distribution	Grammatical correctness	Outliers in POS mix (e.g. too many symbols) indicate atypical input.	✓
Flesch–Kincaid score	Readability score	Mix of POS (e.g., too many symbols) indicates unusual input	✓
Gunning Fog Index	Readability score	Not chosen since it can overestimate difficulty by counting all multisyllabic words	✗
Dale–Chall Readability Score	Readability score	Not accepted since it relies on fixed vocabulary list that may not suit prompts	✗

The Flesch-Kincaid score and the lexical and syntactic complexities metrics are the most promising for observation [15]. In addition, a complex view of several metrics at the same time can make the results more accurate [15].

IV Results

We divided the results of the experiments into two subsections: with limitation to 16 tokens and without limitation of the response size.

TABLE 2
Correlation Results for Experiment 1 (max 16 tokens) and Experiment 2 (unlimited)

Metric	**Corr. (Exp 1)**	**p-value (Exp 1)**	**Corr. (Exp 2)**	**p-value (Exp 2)**
Word frequency	0.1660	1.29e-07	0.0623	0.0491
Number of clauses	0.0259	0.4135	0.0666	0.0352
Depth of dependency tree	–0.0088	0.7807	0.2309	1.42e-13
Words number	0.5672	3.36e-86	0.0572	0.0706
Polysemous words	0.0542	0.0867	–0.1694	7.09e-08
PoS distribution	–0.0891	0.0048	-0.1435	5.24e-06
Flesch–Kincaid grade	0.0602	0.0572	0.3429	5.79e-29

Table 2 shows the linear correlation (Pearson correlation) together with the associated p-value. All metrics values that have a p-value greater than 0.05 we indicated as insignificant.

A. Maximal number of tokens is ***16***

The observation took approximately half an hour overall. When limiting the LLM's responses to a maximum of 16 tokens, our correlation analyses reveal several interesting outcomes, which are depicted in the graphs below.

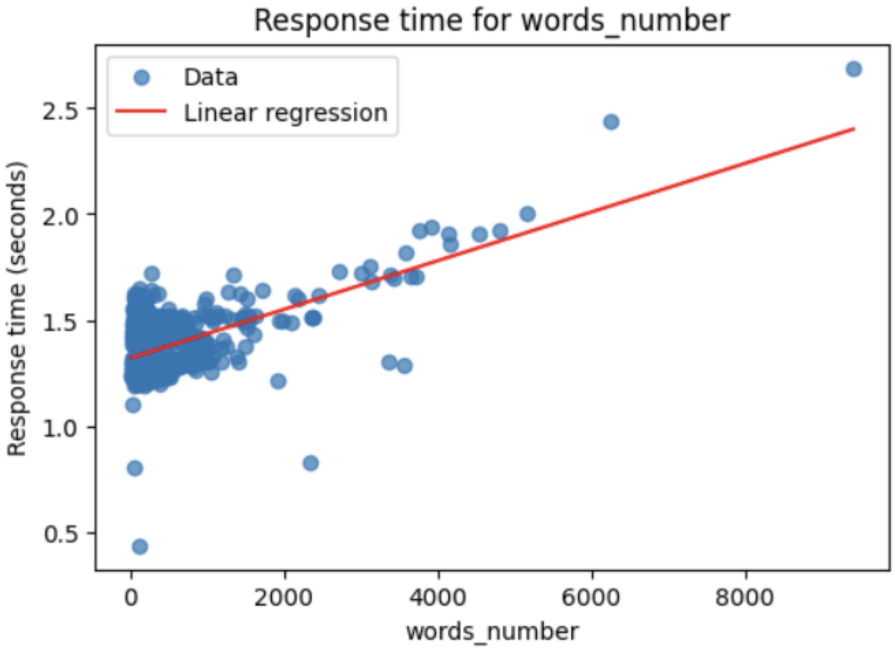

Fig. 1. Words number and response time

Fig. 1 displays the dependency of inference time on word number. The correlation between response time and this metric is the highest among the measured metric (0.5672) and the p-value is significantly lower than 0.05 (3.36e-86), making it a substantial metric.

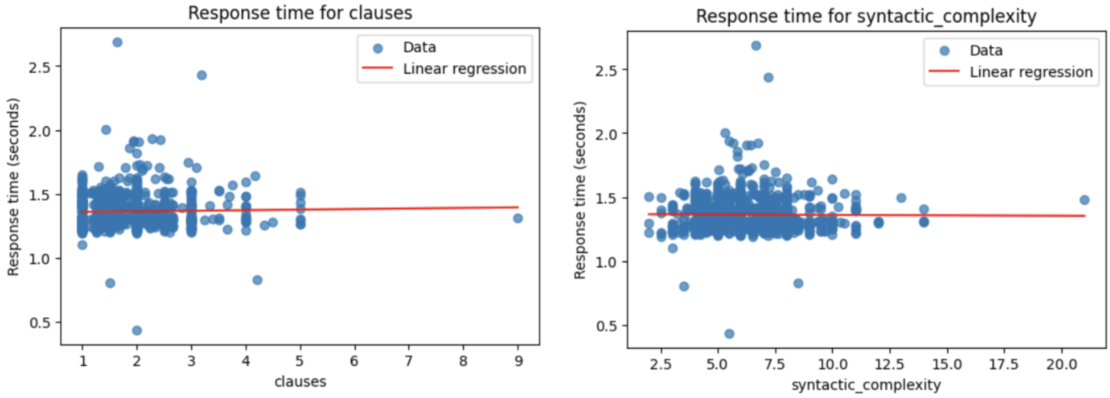

Fig. 2. Clauses & depth of dependency tree

This figure represents the metrics that did not show any correlation, while the p-value is extremely large. This makes the depth of dependency tree and number of clauses useless in the context of LLM's inference time.

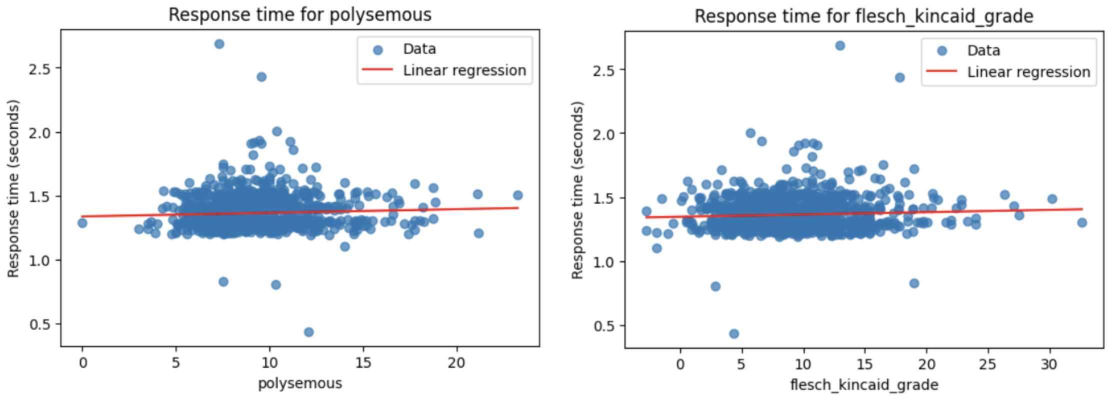

Fig. 3. Polysemous & Flesch-Kincaid grade

The share of polysemous words and the Flesch-Kincaid readability index show a relatively small correlation (0.05-0.06), and the p-value is near five percent. This indicates that the bond between these values and the inference time is weak and might not be statistically significant.

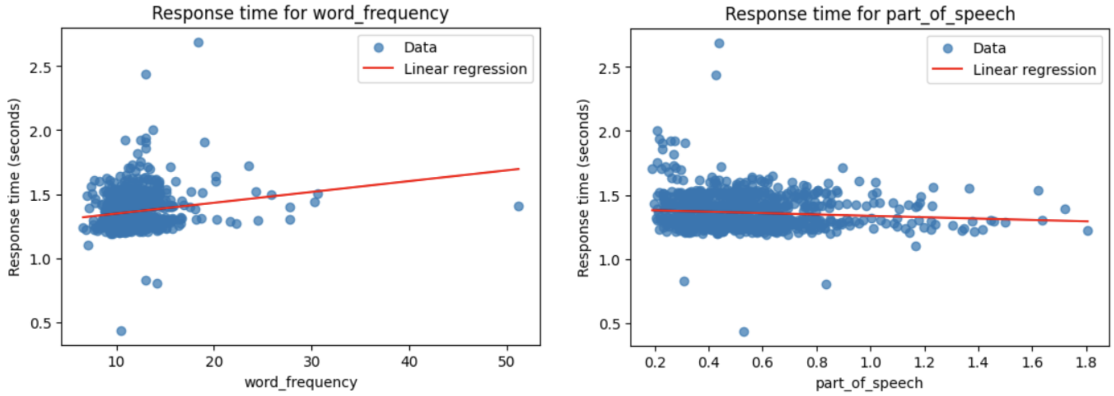

Fig. 4. Words frequency & PoS distribution

The data displayed by this graph are similarly poorly correlated with the inference time, but small p-values make it prominent.

*B. Maximal number of tokens is **unlimited***

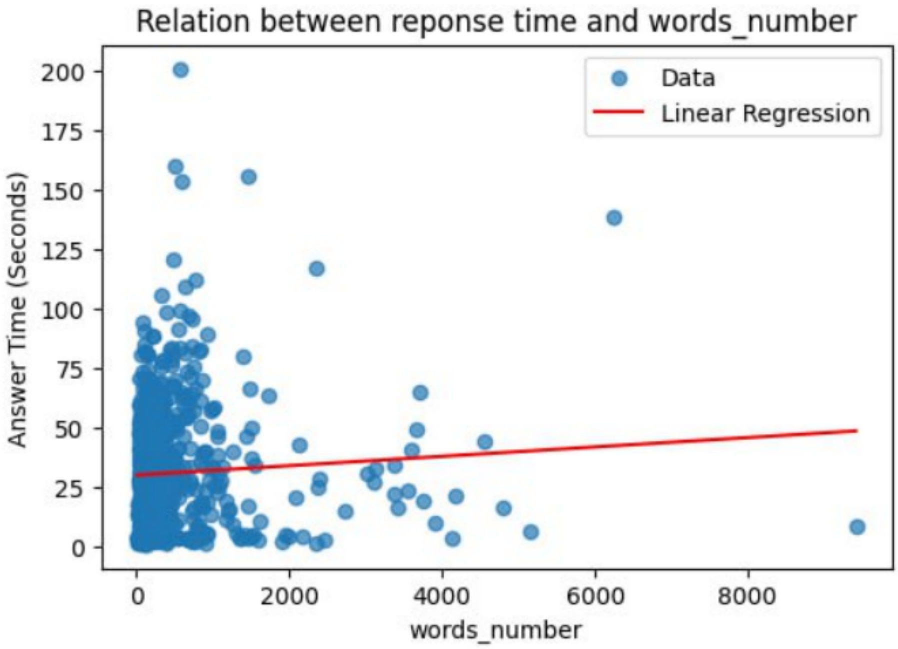

Fig. 5. Words number and response time

In contrast to the previous experiment, the correlation between the number of words and the response time is mostly insignificant. This means that the generation time does not depend strictly on the number of words.

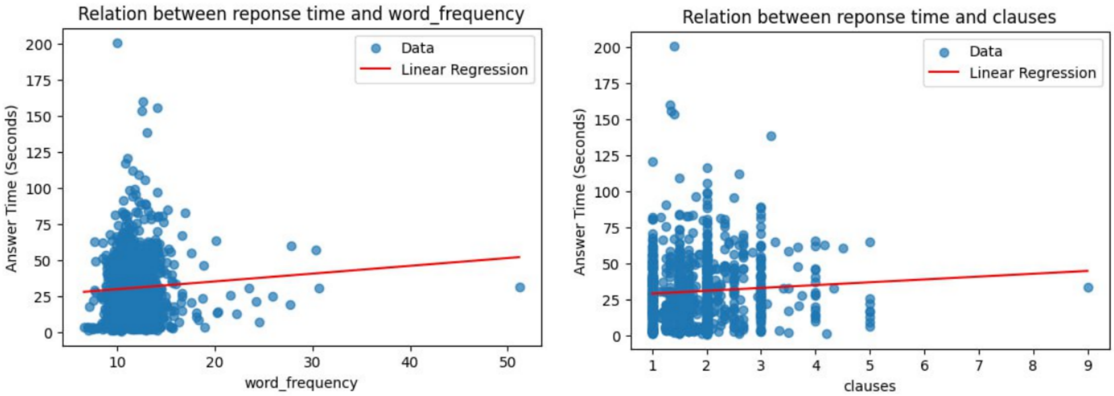

Fig. 6. Words frequency & clauses

Although these parameters are very weakly related to the response time, the p values (0.0491 and 0.0352) make them important.

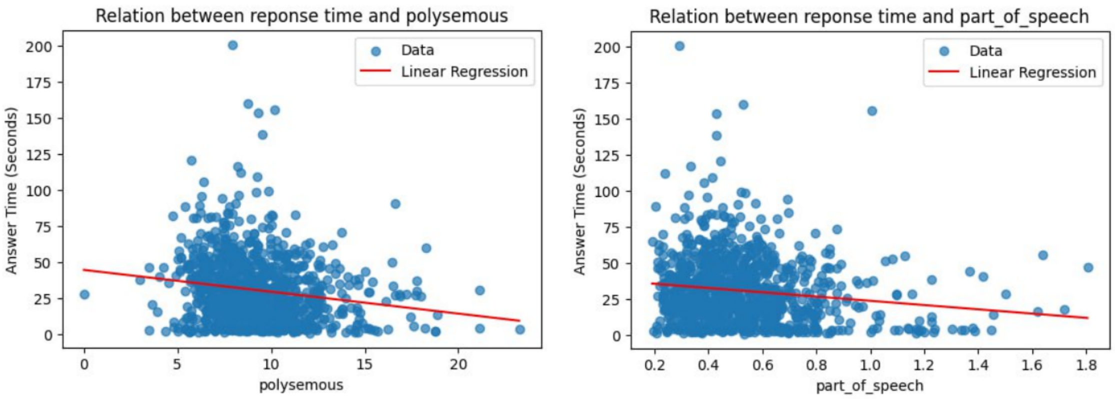

Fig. 7. Polysemous & PoS distribution

Polysemous words and part-of-speech distribution show a better, statistically significant, but negative correlation.

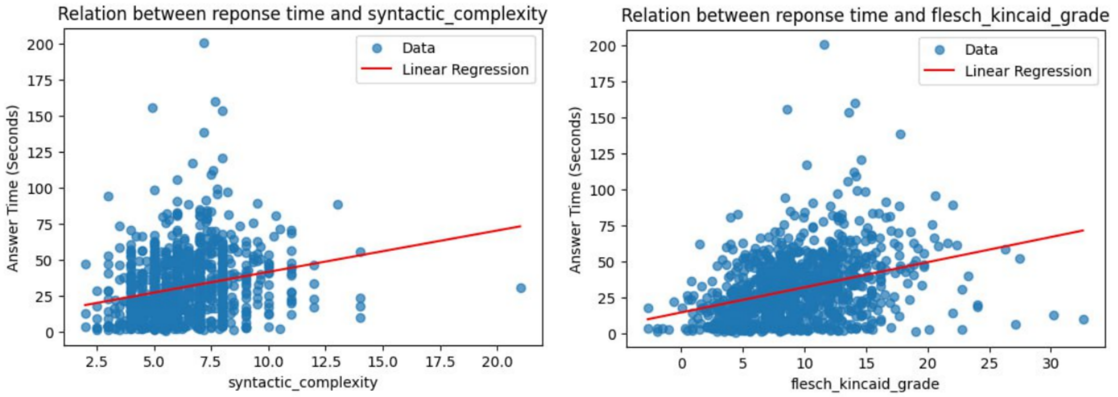

Fig. 8. Depth of dependency tree & Flesch-Kincaid grade

These metrics displayed the strongest correlation among all the others. Their p-values are $1.42 * 10^{-13}$ and $5.79 * 10^{-29}$, which makes them very relevant for the research.

V Discussion

At the beginning of the research, the hypothesis stated that the correlation between the number of words and the response time would be the strongest. Although we discovered that the correlation was noticeable with the number of tokens limited to 16 (i.e. the time to begin the generation), the correlation in the full case is insignificant. This finding rejects the initial hypothesis. Hereby, the inference time does not depend on the size of the prompt.

We investigated whether the number of polysemous words in a prompt does not have a perceptible influence on response time. In the first case, this metric showed a flat line and a negligible correlation. In the second case, the correlation was weak but with a tangible negative slope. Therefore, the content of the polysemous words has a small inverse influence on the overall generation time.

The experiment shows that the part-of-speech distribution has an inverse correlation with the total inference time and the time to the first 16 tokens of the response. In these tests, the metric displayed a p-value less than 0.05, making it prominent. The nature of this correlation means that the more a prompt deviates from an average share of different parts-of-speech, the faster the response. In the second test, the correlation is higher, which indicates that it has greater impact on the output generation than on prompt processing time. Such results may stipulate that LLM processes texts with a higher concentration of certain word classes is easier.

The Flesch-Kincaid readability score shows a positive correlation in both tests. In the first test, it has displayed the p-value of 5.72 percent, so we cannot confidently articulate the metric's statistical significance. Moreover, its correlation is pretty small, which means that it does not affect the inference time noteworthy. In the second test, the readability score presents the strongest positive correlation with response time and the smallest p-value.

The word frequency had a modest positive correlation in both cases. As one can see in the two graphs, the time for the same value of word-frequency metrics drastically varied. Therefore, the word frequency is one of the metrics that showed the correlation both in the overall generation time and the time before an LLM starts to respond.

Despite the absence of an appreciable correlation between the depth of the dependency tree and the response time before an LLM starts to generate a response, this metric showed a notable correlation over the total time. The p-value was 1.42e-13, the second smallest p-value among all metrics after the Flesch-Kincaid score. Hence, the dependencies and connections between words as well as the comprehensiveness of those connections demonstrate the substantial correlation with the response time.

The number of clauses belongs to the same group of complexities as the number of words and the depth of the dependency tree, which showed prominent correlations on the preliminary and overall response times, respectively. However, the number of clauses showed only insignificant and slightly discernible correlations in these cases, respectively. Therefore, this metric is not an illustration in comparison with two other metrics that represent syntactic complexity.

VI Conclusion

This study explored how different characteristics of a prompt affect the response time of a large language model. Among all the metrics we analyzed, the *depth of the dependency tree* and the *Flesch-Kincaid readability score* showed the strongest and most consistent correlations with total inference time. These results imply that the length of time it takes an LLM to produce a response is highly influenced by syntactic structure and general readability.

Other metrics, such as the number of *polysemous words* and *part-of-speech distribution*, also showed statistically significant but weaker and inverse correlations. Interestingly, the *number of words* in a prompt affects only the time before the model starts to respond, not the complete response duration. Thus, different stages of the generation process may be sensitive to other kinds of complexity.

These results have several practical implications. Developers of real-time systems using LLMs, such as chatbots, voice assistants, or educational tools—can use these metrics to predict or manage response times. For instance, prompts with deep syntactic structures or low readability scores might be flagged as potentially slow, allowing preemptive load balancing or latency warnings to users. This could lead to smoother interactions and a better overall experience.

On a broader scale, our findings contribute to a deeper understanding of the relationship between linguistic complexity and computational performance—an area that includes a large gap. This work paves the way for new prompt engineering techniques that strike a balance between speed and quality by offering empirical proof of which prompt features are most important for inference latency. Future research may broaden this analysis to include additional LLM architectures and languages or add more complex textual elements like discourse or logical depth. Integrating our metrics into automated prompt optimization systems that have the ability to instantly rephrase or simplify inputs is another exciting avenue. In the end, we anticipate that this study will close the gap between systems design and natural language processing, making LLMs not only more efficient in real-world applications.

References

[1] G. R. Nahass, S. W. Chin, I. M. Scharf, *et al.*, "Prompt engineering to increase GPT3.5's performance on the Plastic Surgery In-Service Exams," *Journal of Plastic, Reconstructive & Aesthetic Surgery*, 2024. DOI: `10.1016/j.bjps.2024.104548`.

[2] H. Jung, J. Oh, K. A. Stephenson, A. W. Joe, and Z. N. Mammo, "Prompt engineering with ChatGPT3.5 and GPT4 to improve patient education on retinal diseases," *Canadian Journal of Ophthalmology*, 2024. DOI: `10.1016/j.jcjo.2024.01.017`.

[3] H. Liu, H. Yin, Z. Luo, and X. Wang, "Integrating chemistry knowledge in large language models via prompt engineering," *Computers & Chemical Engineering*, vol. 182, 2025. DOI: `10.1016/j.compchemeng.2024.108798`.

[4] A. Garg, K. N. Soodhani, and R. Rajendran, "Enhancing data analysis and programming skills through structured prompt training: The impact of generative AI in engineering education," *Internet of Things and Cyber-Physical Systems*, vol. 3, 2025. DOI: `10.1016/j.iotcps.2024.01.009`.

[5] T. Tania, A. Yläkujala, L. Metso, T. Sinkkonen, and T. Kärri, "Prompt Engineering P2X business ecosystem with generative AI," *Procedia Computer Science*, vol. 249, 2025. DOI: `10.1016/j.procs.2025.01.082`.

[6] W. J. B. van Heuven, P. Mandera, E. Keuleers, and M. Brysbaert, "SUBTLEX-UK: A new and improved word frequency database for British English," *Quarterly Journal of Experimental Psychology*, vol. 67, no. 6, pp. 1176–1190, 2014. DOI: `10.1080/17470218.2013.850521`.

[7] B. M. Szmrecsanyi, "On operationalizing syntactic complexity," in *Le poids des mots: Proceedings of the 7th International Conference on Textual Data Statistical Analysis*,

Louvain-la-Neuve, Belgium, 2004, pp. 1032–1039. [Online]. Available: http://www.benszm.net/omnibuslit/Szmrecsanyi2004.pdf.

[8] S. Liu, X. Chen, J. Meng, and N. Lukač, "Improved relation extraction through key phrase identification using community detection on dependency trees," *Computer Speech & Language*, vol. 90, 2024. DOI: 10.1016/j.csl.2024.101636.

[9] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas for navy enlisted personnel," Naval Technical Training Command, Millington, TN, Tech. Rep. Research Branch Report 8-75, 1974. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA006655.pdf.

[10] B. W. Lee and J. H.-J. Lee, "Traditional readability formulas compared for English," *arXiv preprint arXiv:2301.02975*, 2024. [Online]. Available: https://arxiv.org/pdf/2301.02975.

[11] Y.-H. Tseng, P.-E. Chen, D.-C. Lian, and S.-K. Hsieh, "The semantic relations in LLMs: An information-theoretic compression approach," in *Proceedings of NeSy Bridge*, 2024. [Online]. Available: https://aclanthology.org/2024.neusymbridge-1.2.pdf.

[12] W. van Heuven, P. Mandera, E. Keuleers, and M. Brysbaert, "Subtlex-uk: A new and improved word frequency database for british english," *Quarterly journal of experimental psychology (2006)*, vol. 67, Jan. 2014. DOI: 10.1080/17470218.2013.850521.

[13] R. Esfandiari and N. Sadeghian, "The effect of choice of prompts on syntactic complexity, grammatical accuracy, and lexical diversity in l2 argumentative writing essays," *Iranian Journal of Applied Language Studies*, vol. 13, no. 2, pp. 197–218, 2021, ISSN: 2008-5494. DOI: 10.22111/ijals.2021.6882. eprint: https://ijals.usb.ac.ir/article_6882_acc04819e13735185003ed452b1fb0f1.pdf. [Online]. Available: https://ijals.usb.ac.ir/article_6882.html.

[14] S. Liu, X. Chen, J. Meng, and N. Lukač, "Improved relation extraction through key phrase identification using community detection on dependency trees," *Computer Speech and Language*, vol. 89, p. 101 706, 2025, ISSN: 0885-2308. DOI: https:

//doi.org/10.1016/j.csl.2024.101706. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230824000895.

[15] D. Rooein, P. Rottger, A. Shaitarova, and D. Hovy, “Beyond flesch-kincaid: Prompt-based metrics improve difficulty classification of educational texts,” *arXiv preprint arXiv:2405.09482*, 2024.