# Evaluating Large Language Model-based Natural Language Generation for Modular Dialog systems

**Vincent Emmerling** *, **Christoph Kowalski** *, **Amelie Robrecht-Hilbig** *, **Stefan Kopp**
University of Bielefeld
vemmerling, ckowalski1, arobrecht, skopp@techfak.uni-bielefeld.de
All authors marked with * contributed equally. Names are ordered alphabetically.

**Abstract**

While many dialogue systems currently use end-to-end solutions, modular systems offer greater control, sustainability, and more human-like dialogue. This makes them relevant, especially when aiming to study human behavior patterns in interactions or applying them to sensitive domains. In this paper, we develop an automated metric to measure the quality of an LLM-based NLG-component in a modular system based on the hallucination tendency and linguistic quality. We apply the metric to various language models and usage techniques and, based on the results, discuss the conditions a model must meet in order to be a good candidate for an NLG-component in a real-time capable dialogue system. Although such automated metrics cannot replace a real interaction study, they help to compare potential approaches of the individual modules. Therefore, they are indispensable when developing and testing modules in isolation. One advancement of the introduced metrics is that it is developed and tested on a German dataset, showing challenges when working with languages other than English and discrepancies to the abilities of Generative AI assumed in current state-of-the-art literature.

**Keywords:** Natural Language Generation, Hallucination Metric, Modular Dialogsystem, Annotator Agreement

## 1. Introduction

With the appearance of Large Language Models (LLMs), more and more dialogue system use end-to-end approaches. While there is no current survey comparing the amount of modular architectures to the amount of end-to-end systems, the growing amount of surveys comparing LLM-based end-to-end approaches stresses their popularity (Qin et al., 2023; Wang et al., 2025; Yi et al., 2025). At the same time, modular systems have the advantage of breaking down the dialogue into subtasks. This allows them to combine different expert systems and makes the dialogue structure more human. Recent modular systems also use LLMs in their components, either in combination with other LLMs or other approaches (Hakimov et al., 2024; Zheng et al., 2025). Given their high competencies in language productions, LLMs are especially promising to form the Natual Language Generation (NLG)-component (Ni and Li, 2023). How such an NLG-component for a modular architecture can be developed and tested will be explored in this paper.

Human interactions are characterized by both conversation partners dynamically adapting to their interlocutor, their preferences, and their current level of knowledge (Brennan and Hanna, 2009). Therefore, we develop a modular dialog system that generates an adaptive explanation for the board game Quarto (Robrecht and Kopp, 2023; Robrecht-Hilbig et al., 2026). The full interaction is structured into a sequence of related adaptation cycles consisting of an explainee's (the partner perceiving the explanation) and an explainer's (the partner generating the explanation) turn. Each cycle consists of the cognitive and the interactive adaptation phase. The process of cognitive adaptation describes how the partner model is updated based on the perceived user feedback. In the interactive adaptation, the agent chooses the next utterance based on the current partner model (Robrecht-Hilbig et al., 2026). SNAPE-PM (Fig. 2) is such an adaptive explainer, designed to co-constructively explain the board game *Quarto!* to a user in German. This architecture has already been tested in user studies and showed a significant positive increase in user understanding while only achieving mixed results for user satisfaction (Robrecht-Hilbig et al., 2026).

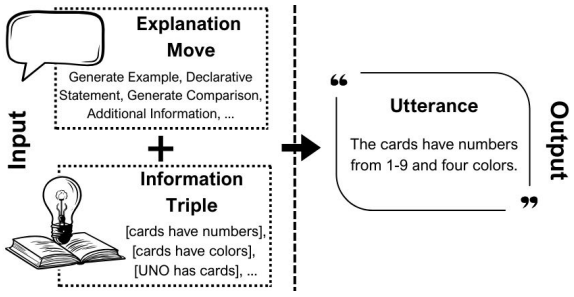


Figure 1: Two-parted input and one-parted output of the NLG-component in a modular system for adaptive explanation generation.

The agent uses an NLG component to transform the structured data used in the dialog state, e.g. *triple: [Players, are, Opponents]* combined with an instruction on how to present this information, into



*Proceedings of The Fourth Workshop on the Role of Resources in the Age of Large Language Models (RESOURCEFUL 2026), pages 142–160 May 11, 2026.* 

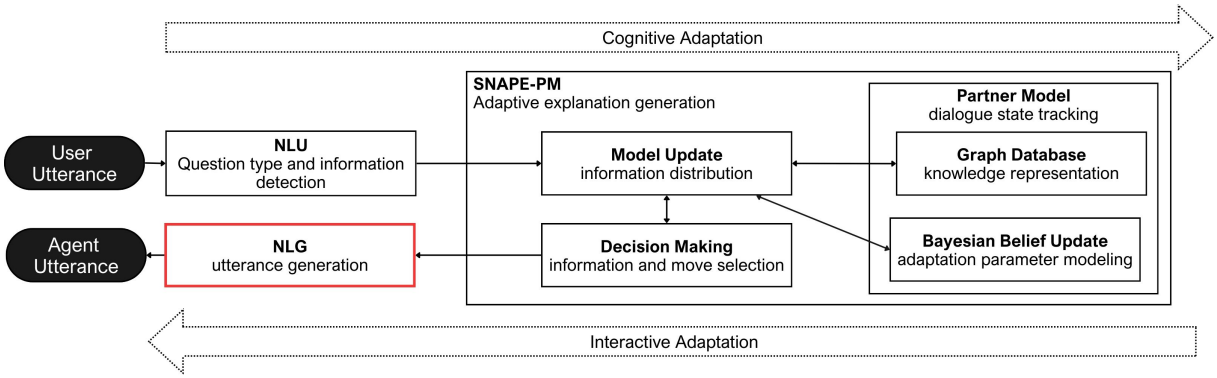


Figure 2: Visualization of the dataflow in SNAPE-PM. During the cognitive adaptation, the feedback is analyzed, and the partner model is updated. The interactive adaptation consists of the partner-based selection of the next information and move, and its verbalization. The NLG component is part of the interactive adaptation.

natural language for the user to understand. An illustration of the NLG task can be found in Figure 1. Large language models (LLMs) are increasingly used for NLG due to their outstanding capabilities in language generation. One major challenge when using LLMs for NLG is hallucinations, as the LLM might add information to increase sentence quality or naturalness. However, modular systems have a decision-making component that decides what information should be addressed, and therefore, the LLM should not add any extra information. In the case of the proposed explanation system, the decision-making component decides the speed and style of explanation, and therefore, the NLG should not interfere with that decision by hallucinating extra information that might increase explanation complexity. This paper addresses the issue of developing an NLG that is free of hallucination, while processing time allows real-time interaction. The two main questions addressed in this paper are: Which models and methods (prompting vs. chain-of-thought prompting vs. fine-tuning) achieve the best results in NLG? How can the quality of NLG be evaluated?

To answer these questions, we discuss the current state-of-the-art (Sec. 2) in NLG, emphasizing the opportunities and challenges of using LLMs in that domain. In the rest of the paper, we discuss the models used, their application, and their evaluation (Sec. 3). Based on the results (Sec. 4), we provide some recommendations for optimizing NLG components in modular agents (Sec. 5).

## 2. Related Work

### 2.1. NLG with LLMs

The NLG component is central for modular dialog systems as natural language is the interface with which the agent communicates with the user. Before the invention of transformers, grammars and templates were used to generate natural language for dialog systems (Santhanam and Shaikh, 2019). With the advances in LLMs, using those has become state-of-the-art for NLG tasks as they are more flexible than pre-defined grammars (Santhanam and Shaikh, 2019). However, using LLMs for NLG causes new challenges like hallucination. Ji et al. (2023) differentiate between intrinsic and extrinsic hallucination, where intrinsic hallucination is the contradiction of the source that it references. Extrinsic hallucination is not necessarily wrong information, but information that is not available through the source that it references. This means that the text might refer to information from other sources, attributing it to the wrong source or including more sources than it was asked to do. Even when the information provided is not factually wrong, it is seen as a hallucination as it is unverifiable (Ji et al., 2023). While many end-to-end dialog systems are based on LLMs, their task is different to that of the NLG component in a modular system, as the end-to-end system combines the understanding of the user's utterance, the choice of the action, and the creation of natural language. The task of the NLG in modular systems is more restricted and should therefore not interfere with the decision-making process supporting the need for a hallucination free NLG.

### 2.2. Task-specific adaptation of LLMs

While LLMs have great zero-shot capabilities, they often fail to accomplish specific tasks if they are not instructed properly or have not seen examples of the task they need to perform. The two most prominent techniques for tailoring LLMs to one's needs are prompting and fine-tuning. While it is theoretically possible to train an LLM from scratch, this approach is often too costly and data-intensive

to be practical for most users and developers.

When prompting a language model, the task is provided as natural language input. Most systems in use today have been pre-fine-tuned for handling instructions (instruction models). The field of prompt engineering primarily focuses on how to effectively structure and formulate prompts to elicit the best outputs from the system (Chen et al., 2025). One effective strategy for obtaining more structured responses is called chain-of-thought (CoT) prompting. This method incorporates intermediate reasoning steps, potentially enhancing output quality for reasoning tasks (Wang et al., 2024). Another popular approach to improve output quality is few-shot prompting, which involves giving the model a few examples before requesting it to complete the task. However, the effectiveness of few-shot prompting compared to one-shot prompting varies based on factors such as the specific task and the size of the model (Chen et al., 2025).

While prompting utilizes the capabilities of the given language model directly, fine-tuning involves adapting the model for a specific task. This process requires providing the base model, which has already learned general language patterns, with a dataset containing task-specific data to optimize its parameters for that task (Wu et al., 2025). In contrast to the data necessary for training a model from scratch, fine-tuning is more data-efficient and is therefore frequently used when optimizing for domains with sparse or rare data availability.

### 2.3. Evaluating the performance of LLMs

When evaluating the performance of an LLM, this is usually done in one of two ways: either by using automated metrics or by using human evaluation. For automated metrics, statistical inaccuracy and the use of incorrect evaluation methods are often criticized (Miller, 2024; Sun et al., 2024). The performance highly depends on the size and version of the model, the type of the task, and the formulation of the given prompt. Rapid developments in the field, therefore, necessitate suitable, quickly applicable, and generalizable testing instruments. Currently, several benchmarks testing various abilities in LLMs are developed (Yu et al., 2024; Herron et al., 2024). Both these newly developed and already established testing methods, such as BLEU or ROUGE scores, can reflect the actual performance of LLMs to a very limited extent, as they only reflect parts of linguistic interaction, are influenced by biases, and their results are often reported incorrectly or selectively (Reiter, 2018; Banerjee et al., 2024). Therefore, automated metrics are underinformative in most cases and should not replace human evaluation, but as shown by Suh et al. (2025) they often do. Nevertheless, they are time- and cost-efficient and represent a good option for initial evaluation, as long as they are carefully and appropriately selected (Van Der Lee et al., 2021). Furthermore, Wei and Jia (2021) show that automatic metrics can have statistical advantages with regard to the evaluation of NLG components. While there exist automatic hallucination metrics, these are mostly used for English text and do not produce good results for other languages, such as German (Ul Islam et al., 2025; Kang et al., 2024). Since user evaluation of individual components is costly and time-consuming, and in some cases very unnatural, we propose a combination: each component of a modular system is tested and pre-evaluated during development using automated metrics, so that the finished system can then be evaluated with human users. Therefore, we introduce a metric for testing hallucinations in German NLG-components, which can be used as a first indicator of the component's performance.

## 3. Architecture and Evaluation of an LLM-based NLG

As prompting strategies differ per model and architecture, prompt engineering is a time-consuming process, which may result in a prompt that might work well for one particular model but not for others. Instead of optimizing prompts for a specific model, we opt to introduce a metric to assess the quality of an NLG component, which can be used to compare different techniques and models. This approach enables using the gathered insights for the adaptation of the NLG component when more capable or efficient models are available. As no single model is known to best solve the problem of language generation from structured data, a broad list of open-source models is compared. These include commonly used models from Meta, Google, and co, as visible in Table 1. For this paper we only consider results from the models that are tested for all three techniques: single-shot prompting (baseline), CoT-prompting (DSPy), and fine-tuning. As we aim for an online adaptation, the model has to react in real-time to be a viable option. The selected models need to comply with the hardware limitation of the workstation (Hardware specification: RTX 4090 24 GB VRAM) running the SNAPE-PM architecture in order to enable hosting large-scale human user studies. Due to this limitation, closed-source cloud-hosted models like ChatGPT or Google's Gemini family of models are not considered. All models are run with the help of Ollama, as it provides a clear API for easy integration.

In order to fine-tune LLMs on a single workstation, finetuning of quantized LLMs is conducted with the help of the QLORA (Dettmers et al., 2023)

Table 1: Model usage across different techniques.

| Model | Baseline | CoT | fine-tuned |
|---|---|---|---|
| Gemma2 | ✓ | ✓ | ✓ |
| Llama3.2 | ✓ | ✓ | ✓ |
| Qwen2.5-l | ✓ | ✓ | ✓ |
| Phi4 | ✓ | ✓ | |
| Llama3.3 | ✓ | | |
| Llama3.1 | ✓ | | |
| Mistral | ✓ | | |
| Qwen2.5-s | ✓ | | |

implementation in unsloth[1], which simplifies fine-tuning by extending the widespread Huggingface Transformers library. Hyperparameter tuning is performed using unsloth in a standard grid search approach, where $\text{lora}_r$, $\text{lora}_\alpha$, dropout probability, and $\text{learning-rate}_\eta$ are optimized for. As Ollama supports loading and inference with quantized models, the technical integration of the fine-tuned model into the overall system is straightforward.

For an optimized Chain-of-Thought (CoT)-reasoning prompt with the DSPy[2] framework, an evaluation metric is supplied to the optimizer. In theory, this enables optimization without an optimization dataset. However, such a metric needs to reliably evaluate language quality and hallucination in German while being computationally lightweight. Because such a metric is not available, the text similarity, in the form of the BERTscore (Zhang et al., 2020), of the generated utterances and the utterances from the finetuning dataset is used as an evaluation metric. Additionally, the score is zero if specific move requirements, e.g., a comparison move does not include the comparison domain, are not satisfied. While our proposed hallucination metric could be used for CoT prompt optimization, its computational complexity makes the computational intensive DSPy optimization infeasible. Optimization is performed by iteratively generating bootstrapped examples and identifying optimal prompt combinations. While DSPy determines prompts internally, it relies on a user-defined pipeline describing steps called *predictions* to define the task. We define a two-stage pipeline where the first prediction is a potentially suboptimal utterance and the second prediction cleans up the sentence. We use the MIPROv2 optimizer with the described similarity metric, pipeline, 100 candidates, 500 trials total per run, and up to 10 bootstraped examples.

To evaluate the effectiveness of the fine-tuning and DSPY, a baseline is created which utilizes a straightforward prompt describing the task, visible in the appendix in 10.1, supplemented by a separate prompt addition based on the NLG moves described in the next section (Tab.2).

### 3.1. Task-specific reference dataset

As both CoT and the fine-tuning process require reference utterances for optimization, a dataset is created that represents the NLG task of the adaptive explainer. The input to the NLG is a list of triples, which are to be verbalized, and the explanation move, which describes the style of explanation, for example by providing information using a comparison or deepening information by giving additional information. While some moves are used to fulfill different goals at different dialogue states, they share the same linguistic surface. For example: *Quarto is a board game.* can be a declarative statement to provide new information, if not mentioned before, but a repetition otherwise. Due to this linguistic similarity, those moves, however, are mapped to the same prompt, decreasing the number of linguistic meta moves for the NLG (Tab. 2).

Table 2: The explanation moves used in SNAPE-PM and their mapping to the three NLG-moves used in the NLG-component.

| Explanation move | NLG move |
|---|---|
| Deepen Comparison<br>Provide Comparison | Generate Comparison |
| Provide Declarative<br>Deepen Additional<br>Answer Declarative<br>Paraphrase Information | State Information |
| Answer Paraphrase | Confirm Clarify |

To create the required dataset, a knowledge graph with 54 triples related to the game of *UNO* is built manually. Based on the ontology and the different explanatory moves, a dataset with reference utterances is created. The dataset contains all relevant move permutations, as SNAPE-PM utilizes different explanatory moves to verbalize a piece of information and the corresponding reference utterances, which are free of any form of hallucination. The dataset consists of 142 data points. This switch in domains (from the original domain of *Quarto!* to *UNO*) is necessary to not poison the training dataset and to not eliminate the ability to measure result quality.

[1] https://unsloth.ai/
[2] https://dspy.ai/

[3] Maximum perplexity is set to 1000 and the maximum possible number of errors to 15, as these were the highest observed values in the dataset.

$$\text{nlg component score} = 1 - \Big(w_1 \times \text{hallucination} + w_2 \times (1 - \text{naturalness}) + w_3 \times \frac{\text{perplexity}}{\max(\text{perplexity})} + w_4 \times \frac{\text{language errors}}{\max(\text{possible errors})}\Big)^3 \quad (1)$$

### 3.2. Methods for evaluation

The main goal of the evaluation process is to determine if generated utterances are free of hallucinations. This is crucial to ensure the NLG component reliably verbalizes triples for the explanation process.

Although desirable, manually annotating hallucinations is time-intensive and not feasible for the large number of models evaluated. To remedy this issue an automatic hallucination detection metric is developed. While automatic metrics can benefit from statistical power, without proven agreement with human annotators, they can not be used reliably in practice (see Sec. 2). The developed automatic hallucination detection metric is LLM-based, using a prompt that is designed to achieve reliable agreement with human raters. As pointed out beforehand, existing hallucination metrics fail to measure hallucination for languages other than English. Therefore, a custom hallucination metric applicable to German utterances needs to be developed (Ul Islam et al., 2025; Kang et al., 2024). The *deepseek-r1:27b* model showed the most promising results from a small set of different LLMs and is used to evaluate hallucinations in combination with the iteratively developed prompt (see Figure 3). The prompt follows a modular design consisting of (1) the **context** defining triple and reference data, (2) the **sentence to evaluate**, which is the actual utterance, (3) the **evaluation criteria** which is dependent on the given move, and (4) a binary **scoring**. We compared the hallucination metric to human judgment. As the objective is to develop an NLG component that does not hallucinate, most scores should be zero, indicating the absence of hallucination. Agreement for all prompt versions is measured using Gwet's AC1 in addition to the interclass correlation (ICC) between human raters (n=1) and the prompt-based metric (Fig.3 for the agreement over all prompt versions). A common measure for annotator agreement is Cohen's $\kappa$, however, it is not suitable to capture agreement for unbalanced classes. As the goal is to generate utterances that are free of hallucination, the class distribution is severely imbalanced. For Gwet AC1, a score of $0.876$ is reached, and for ICC $0.865$ indicating a strong correlation for both. The scores are calculated based on 384 utterances, which were labeled by a human annotator and by the developed hallucination metric.

The custom hallucination metric produces a binary classification indicating whether hallucination is present. While the hallucination metric detects

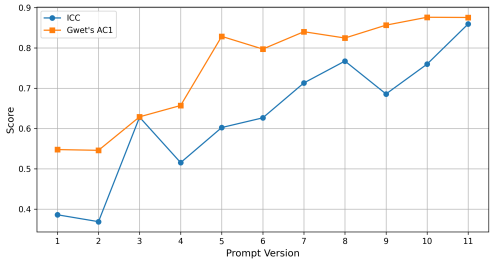


Figure 3: Plot showing the growing agreement between human annotator and agreement measures over the revision of prompts.

the presence or absence of hallucinations, it does not provide insight into overall utterance quality. As both factors are relevant for a high-quality NLG-component, the NLG component score combines those measures. First, UniEval (Zhong et al., 2022) provides a normalized score ranging from 0 to 1, which indicates the naturalness of the utterance. Second, perplexity (Xu et al., 2025) is measured as a positive floating-point value, with lower scores indicating better performance. Third, LanguageTool [4] reports error counts within the sentence. Based on a weighted combination of the hallucination metric, the naturalness score, a relative perplexity, and a relative language quality score, we introduce this NLG component score (Eq. 1). The development of such a metric is important for the developer to have an indication of the performance of the respective NLG before conducting user studies. Additionally, focusing solely on a single metric, such as perplexity or hallucination, may lead to an NLG component that optimizes only that goal, for example, producing hallucination-free utterances that do not sound natural.

## 4. Results

The evaluation compares the results of the previously introduced models using prompting (baseline), CoT prompting (dspy), and fine-tuning. Here, we focus in particular on the results with regard to hallucination tendencies, the previously introduced NLG component score, and the real-time capability of the various approaches. The results discussed in this section only compare the three language models that have been used for all three techniques. Some are excluded as they produced formatting

[4] https://github.com/languagetool-org/languagetool

failures in DSPy-prompting (Phi4), showed bad performance in the baseline (Llama 3.1, Mistral, Qwen2.5-small), or had to high computational demands for fine-tuning (Llama 3.3).

### 4.1. Hallucinations

Figure 4 shows that different language models show different tendencies to hallucinate for different moves. While the prompted Gemma2 hallucinates for $69\%$ of the comparisons, but never when stating information, Qwen2.5-large hallucinates in less than $50\%$ for all moves (confirm clarify: $38\%$; generate comparison: $44\%$; state information: $12\%$). In general, the tendency to hallucinate is reduced by both the CoT and the fine-tuned utilization approach (baseline: $30\%$; DSPy: $12\%$; fine-tuned: $13\%$). While Gemma2 and Llama3.2 improve their performance when using CoT prompting, Qwen 2.5 only shows minor improvement compared to the baseline. Most models do decrease their hallucination frequency when fine-tuned. Llama3.2, for example, improves its performance by $25\%$ from $38\%$ to only $10\%$ of its output including hallucinations. Nevertheless, the performance of Gemma2 is rather negatively influenced. While comparisons are generated with less hallucinations ($69\% \rightarrow 25\%$), the hallucination frequency for the moves confirm clarify ($5\% \rightarrow 44\%$) and state information ($0\% \rightarrow 25\%$) increase in comparison to the baseline. In all baseline models, intrinsic and extrinsic hallucination can be observed (see Figure 8).

### 4.2. NLG component score

In addition to the main measure of hallucination frequency, the NLG component score also includes measures of the naturalness and linguistic quality of the utterances, thus providing a more comprehensive picture. LanguageTool results showed strong linguistic quality for all three fine-tuned models (see Figure 18) with fewer than 0.2 errors per utterance, whereas Gemma2 was an outlier at $0.73$ errors, and perplexity scores ranged from $216$ to $350$ (see Figure 19). UniEval naturalness scores clustered around $0.8$, improving for Qwen2.5-Large and Llama3.2 but declining for Gemma2 and Phi4 (see Figure 17). When using CoT prompting, language errors were minimal across models, with Llama3.2 showing the highest rate at 0.23 errors per utterance (see Appendix). Perplexity improved overall, with Qwen2.5 achieving the lowest average (129.54), followed by Llama3.2 (see Appendix). UniEval results mirrored the fine-tuned models, with all scores clustering around 0.8 (see Appendix). The results comparing the nlg component scores (Fig. 5) show that models that are prompted using CoT or fine-tuned have show significantly higher scores on average. Next to the decrease of hallucinations that has been discussed before (Sec. 4.1), this is caused by model-dependent changes in the different component such as the perplexity scores: While fine-tuning decreases the perplexity score of Qwen2.5-large from $435.75$ to $262.19$ and CoT to $129.54$, the perplexity score of Gemma2 increases when fine-tuned ($198.4 \rightarrow 216.58$) and even worse when using CoT ($353.08$). For Qwen2.5-large, both fine-tuning and CoT significantly help to increase the score for all moves, Llama 3.2 rather benefits from fine-tuning, while Gemma2 benefits from fine-tuning and CoT when generating comparisons only (Tab. 3).

| model | move | baseline | CoT | fine-tuned |
|---|---|---|---|---|
| Gemma2 | CC | **0.05** | 0.12 | 0.27 |
| | GC | 0.31 | **0.12** | 0.19 |
| | SI | **0.03** | 0.20 | 0.15 |
| Llama3.2 | CC | 0.23 | **0.15** | 0.18 |
| | GC | 0.46 | 0.12 | **0.07** |
| | SI | **0.16** | 0.18 | 0.25 |
| Qwen2.5-l | CC | 0.31 | 0.13 | **0.11** |
| | GC | 0.28 | 0.21 | **0.07** |
| | SI | 0.25 | **0.10** | 0.16 |

Table 3: Table showing NLG component scores distributed by moves (confirm clarify (CC), generate comparison (GC), state information (SI)), model and technique. The lowest value for each move is marked in bold, the lowest for the move for all models is marked in red.

### 4.3. Runtime

While CoT prompting is convincing in terms of its low hallucination frequency and high NLG component score, the technique leads to significantly increased runtime depending on the moves (Fig. 6). Since SNAPE-PM is a modular system; the runtime of the individual models add up (in some cases), which makes short runtime even more important. While Llama is the fastest with an average of $116.6ms$ (fine-tuned) and $303.1$ (DSPy), Gemma2 is the slowest fine-tuned ($626.8$) and Qwen2.5-large the slowest DSPy prompted model ($1314.5$). Especially when generating a comparison with the DPSy-prompted Qwen2.5-large model the runtime of over 21 seconds clearly exceeds the maximum runtime of a component for real-time dialog systems.

### 4.4. Domain transfer

While the models in the previous sections are tested on the same domain they were trained on (*UNO*), in this section, we take a look at the performance

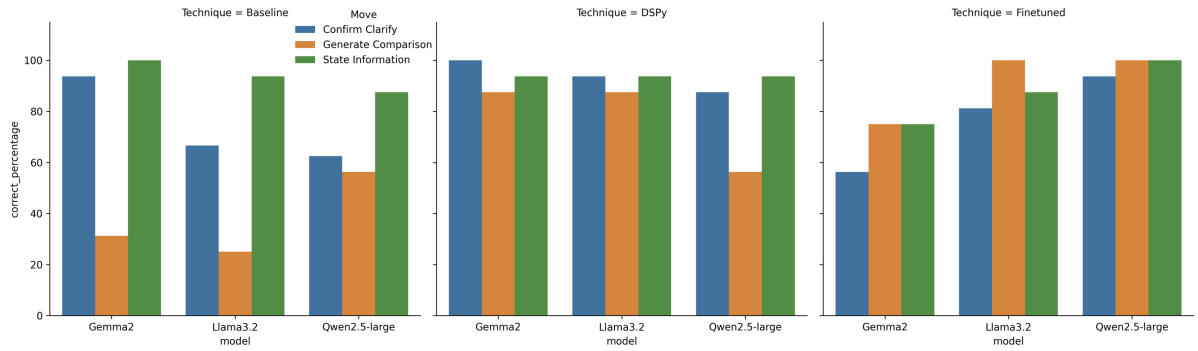


Figure 4: Model-wise hallucination metric results per move for baseline, dspy, and fine-tuned models.

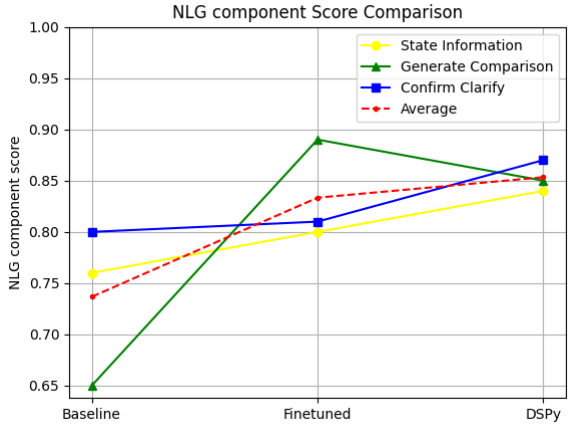


Figure 5: NLG component scores for all three techniques per move and on average.

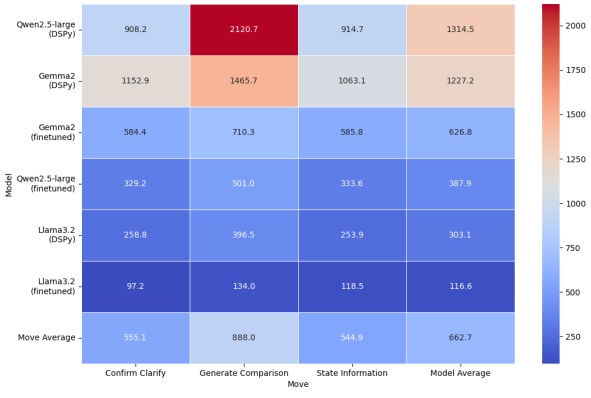


Figure 6: Heatmap of runtime performance in $ms$ comparing fine-tunes and DSPy prompted models.

when transferring to the domain of *Quarto!*. We only look at the fine-tuned LLMs regarding the domain transfer to *Quarto!* as for *UNO*, they perform better than the baseline on the hallucination and nlg component score and were significantly faster than the

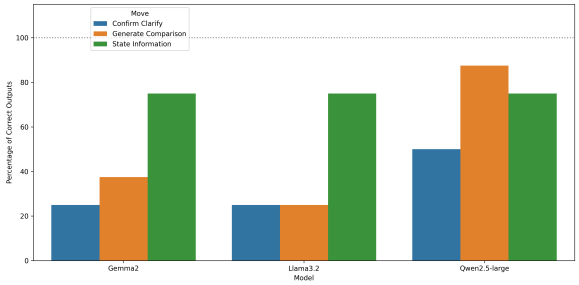


Figure 7: Percentage of hallucination-free utterances for fine-tuned models distributed by move and model.

CoT-prompted models. In general, one can say that the performance on the *Quarto!*-domain is worse than on the *UNO*-domain (Fig. 7). Interestingly, we see that Qwen2.5 event though it is only half the size of Gemma2, performs well, especially when generating comparisons for *Quarto!*. While all three models reach high scores when stating information (Gemma2: 0.76; Llama3.2: 0.76; Qwen2.5-large: 0.74), the scores differ significantly when generating comparisons (Gemma2: 0.38; Llama3.2: 0.24; Qwen2.5-large: 0.83) and using the confirm clarify move (Gemma2: 0.23; Llama3.2: 0.24; Qwen2.5-large: 0.48).

## 5. Discussion

We introduced an LLM-based **hallucination metric** that has been tested and iteratively improved to show a high agreement with the human annotator. This newly invented hallucination metric shows that the tendency to hallucinate is not only dependent on task (the explanation move fulfillment), model, or technique, but also on combinations of these factors. While fine-tuning increases the ability to generate hallucination-free comparisons for Gemma2, the performance for the other moves is decreased. This is not the case for the other two models, which show fewer hallucinations for all moves when fine-tuned. Gemma2 might use general knowledge for

the *confirm clarify* and *state information* moves, which are weakened after fine-tuning. Looking at the results for all models and techniques, patterns for *generate comparison* differ from the two other moves. While it is causing the most hallucinations for the two prompting approaches, it causes the least after fine-tuning. This is probably related to the higher complexity of the task, as it has been shown that hallucination frequency correlates with task complexity (Chakraborty et al., 2025). While our observation for the baseline models aligns with Chakraborty et al. (2025), who find that larger models hallucinate less, this does not hold for the CoT-prompted or fine-tuned models.

Next to the primary goal of hallucination-free utterances, linguistic correctness and quality are also relevant factors for a good NLG-component. All these factors are evaluated in combination using the **NLG component score**. While the single-shot prompted Gemma2 model performed well for the simpler moves, all models had a low score when generating comparisons in the baseline condition. These scores increased with both other techniques. Especially Qwen2.5-large showed a small variance and high overall scores when fine-tuned.

While the comparison of hallucination frequency and NLG component score show that the quality of single-shot prompted models is not sufficient to be used for a dialog system, the **runtime** comparison excluded CoT-prompting as an option. Even though speech quality is high, a reasoning time above 20 seconds is way too high to make this technique considerable for a real-time interaction.

Before looking into the actual results, one needs to reflect on the applicability of the **two domains**, as both are closely related. When selecting training and testing domains, one needs to balance between similarities and differences. While a higher level of similarity usually has a positive influence on the task performance, it also supports over-fitting. At the same time, a fine-tuned model is not designed to perform well on a completely unrelated task or a differently structured domain. With the domains of *UNO* and *Quarto!* we decided to choose two domains sharing high similarities while being aware that they might be too close. The results for both techniques clearly show that, despite the high degree of similarity, performance in the new domain clearly declines. However, general structures remain intact. For both domains all models show high performance when stating information. While Gemma2 and Llama3.2 show a large decrease of performance, Qwen2.5-large displays better domain-transfer capabilities.

# 6. Conclusion

Developing an NLG component that is free of hallucination is crucial for any modular dialog system, as the information reported to the user should be decided upon by the decision-making component and should not be altered by the NLG. As new LLMs with better performance are proposed regularly, the goal of this paper is not to propose one model that performs best but to introduce a new metric and to compare prompting, finetuning, and DSPy for several different models. This paper proposes a novel hallucination metric, using an LLM, which is optimized for agreement with human raters. Building upon this hallucination metrics, a combined NLG component score is developed as a weighted combination of the hallucination metric with scores rating the overall language quality, consisting of naturalness, perplexity, and language errors. These metrics are used to assess the performance of prompted LLMs, fine-tuned LLMs, and LLMs using DSPy for the task of language generation from structured data. It shows that finetuning and the application of CoT both increase performance. However, using DSPy greatly increases runtime, which makes it unsuitable in real-time applications. Additionally, great differences can be observed between different models and their performance on different linguistic moves. While the proposed metric enables testing the NLG component of dialog systems, the generated utterances also show that, for the final evaluation, it is necessary to consider the user and the context, and therefore, an evaluation of the complete system needs to be conducted.

## 6.1. Take-Home Messages

General considerations on using LLMs without hallucinations in structured data–to–text generation can help improve the NLG of a dialogue system. First, a bigger model does not necessarily result in better performance. Second, prompting, finetuning, and DSPy have different effects on different models. Third, the task-specific properties influence the choice of technique, as CoT methods in combination with large models are not suitable for time-sensitive applications.

## 6.2. Future Work

Not only is the generation of language from structured data interesting, but also the extraction of structured data from user utterances. Therefore, the implementation of an NLU component will be addressed in future work. Additionally, in this paper, the *UNO* domain was used to fine-tune the models. Future work will use other domains outside of board games, for example, technical artifacts, to fine-tune the LLMs in order to evaluate whether the system

is able to perform hallucination-free language generation from the provided triples.

Finally, we return to our thoughts from the beginning of the paper. With the NLG component score, we have introduced a metric that can test one of the components of the system during development, thus preparing it for a final interaction study. The development of comparable metrics for the remaining components is future work.

## 7. Limitations

This paper has several limitations, which we discuss below. First, the weights used to calculate the NLG component score have not yet been tested and optimized through studies. Perspectively, these weights require empirical testing in the future. Since this paper is based on a student thesis, the hallucination metric is only optimized based on agreement with one annotator. A comparison with several human annotators would be desirable. Although the focus of our work is not on the evaluation of specific models, we are aware that the language models discussed here are no longer up to date. Many scientists are currently confronted with this limitation, as the time required for careful analysis is difficult to reconcile with the rapid developments in the field of research.

## 8. Acknowledgments

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/3 2026 – 438445824, project A01.

## 9. Bibliography

Sourav Banerjee, Ayushi Agarwal, and Eishkaran Singh. 2024. The Vulnerability of Language Model Benchmarks: Do They Accurately Reflect True LLM Performance? doi:10.48550/arXiv.2412.03597 arXiv:2412.03597 [cs].

Susan E. Brennan and Joy E. Hanna. 2009. Partner-Specific Adaptation in Dialog. *Topics in Cognitive Science* 1, 2 (April 2009), 274–291. doi:10.1111/j.1756-8765.2009.01019.x

Trishna Chakraborty, Udita Ghosh, Xiaopan Zhang, Fahim Faisal Niloy, Yue Dong, Jiachen Li, Amit K. Roy-Chowdhury, and Chengyu Song. 2025. HEAL: An Empirical Study on Hallucinations in Embodied Agents Driven by Large Language Models. In *Findings of the Association for Computational Linguistic*. Association for Computational Linguistics, Suzhou, China, 21226–21243. https://aclanthology.org/2025.findings-emnlp.1158.pdf

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2025. Unleashing the potential of prompt engineering for large language models. *Patterns* 6, 6 (June 2025), 101260. doi:10.1016/j.patter.2025.101260

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLORA: Efficient Finetuning of Quantized LLMs. In *Proceedings of the 37th Conference on Neural Information Processing System*. Association for Computing Machinery, New Orleans, 10088–10115. doi:10.5555/3666122.3666563

Sherzod Hakimov, Yan Weiser, and David Schlangen. 2024. Evaluating Modular Dialogue System for Form Filling Using Large Language Models. In *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*. Association for Computational Linguistics, St. Julians, Malta, 36–52. doi:10.18653/v1/2024.scichat-1.4

Emily Herron, Junqi Yin, and Feiyi Wang. 2024. SciTrust: Evaluating the Trustworthiness of Large Language Models for Science. In *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, Atlanta, GA, USA, 72–78. doi:10.1109/SCW63240.2024.00017

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55, 12 (Dec. 2023), 1–38. doi:10.1145/3571730

Haoqiang Kang, Terra Blevins, and Luke Zettlemoyer. 2024. Comparing Hallucination Detection Metrics for Multilingual Generation. arXiv:2402.10496 [cs.CL] https://arxiv.org/abs/2402.10496

Evan Miller. 2024. Adding Error Bars to Evals: A Statistical Approach to Language Model Evaluations. doi:10.48550/arXiv.2411.00640 arXiv:2411.00640 [stat].

Xuanfan Ni and Piji Li. 2023. A Systematic Evaluation of Large Language Models for Natural Language Generation Tasks. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics*. Chinese Information Processing Society of China, Harbin, China, 40–56. https://aclanthology.org/2023.ccl-2.4/

Libo Qin, Wenbo Pan, Qiguang Chen, Lizi Liao, Zhou Yu, Yue Zhang, Wanxiang Che, and Min Li. 2023. End-to-end Task-oriented Dialogue: A Survey of Tasks, Methods, and Future Directions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 5925–5941. doi:10.18653/v1/2023.emnlp-main.363

Ehud Reiter. 2018. A Structured Review of the Validity of BLEU. *Computational Linguistics* 44, 3 (Sept. 2018), 393–401. doi:10.1162/coli_a_00322

Amelie Robrecht and Stefan Kopp. 2023. SNAPE: A Sequential Non-Stationary Decision Process Model for Adaptive Explanation Generation:. In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence*. SCITEPRESS - Science and Technology Publications, Lisbon, Portugal, 48–58. doi:10.5220/0011671300003393

Amelie S. Robrecht-Hilbig, Christoph Kowalski, and Stefan Kopp. 2026. Generation and evaluation of adaptive explanations based on dynamic partner-modeling and non-stationary decision making. *Frontiers in Computer Science* 8 (Feb. 2026), 1558674. doi:10.3389/fcomp.2026.1558674

Sashank Santhanam and Samira Shaikh. 2019. A Survey of Natural Language Generation Techniques with a Focus on Dialogue Systems - Past, Present and Future Directions. doi:10.48550/arXiv.1906.00500 arXiv:1906.00500 [cs].

Ashley Suh, Isabelle Hurley, Nora Smith, and Ho Chit Siu. 2025. Fewer Than 1% of Explainable AI Papers Validate Explainability with Humans. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–7. doi:10.1145/3706599.3719964

Kun Sun, Rong Wang, and Anders Søgaard. 2024. Comprehensive Reassessment of Large-Scale Evaluation Outcomes in LLMs: A Multifaceted Statistical Approach. doi:10.48550/arXiv.2403.15250 arXiv:2403.15250 [cs].

Saad Obaid Ul Islam, Anne Lauscher, and Goran Glavaš. 2025. How Much Do LLMs Hallucinate across Languages? On Realistic Multilingual Estimation of LLM Hallucination. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 29065–29086. doi:10.18653/v1/2025.emnlp-main.1481

Chris Van Der Lee, Albert Gatt, Emiel Van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language* 67 (May 2021), 101151. doi:10.1016/j.csl.2020.101151

Hongru Wang, Lingzhi Wang, Yiming Du, Liang Chen, Jingyan Zhou, Yufei Wang, and Kam-Fai Wong. 2025. A Survey of the Evolution of Language Model-Based Dialogue Systems: Data, Task and Models. doi:10.48550/arXiv.2311.16789 arXiv:2311.16789 [cs].

Zecheng Wang, Chunshan Li, Zhao Yang, Qingbin Liu, Yanchao Hao, Xi Chen, Dianhui Chu, and Dianbo Sui. 2024. Analyzing Chain-of-thought Prompting in Black-Box Large Language Models via Estimated V-information. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*. ELRA and ICCL, Torino, 893–903.

Johnny Wei and Robin Jia. 2021. The statistical advantage of automatic NLG metrics at the system level. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 6840–6854. doi:10.18653/v1/2021.acl-long.533

Xiao-Kun Wu, Min Chen, Wanyi Li, Rui Wang, Limeng Lu, Jia Liu, Kai Hwang, Yixue Hao, Yanru Pan, Qingguo Meng, Kaibin Huang, Long Hu, Mohsen Guizani, Naipeng Chao, Giancarlo Fortino, Fei Lin, Yonglin Tian, Dusit Niyato, and Fei-Yue Wang. 2025. LLM Fine-Tuning: Concepts, Opportunities, and Challenges. *Big Data and Cognitive Computing* 9, 4 (April 2025), 87. doi:10.3390/bdcc9040087

Weizhe Xu, Serguei Pakhomov, Patrick Heagerty, Eric Horvitz, Ellen R. Bradley, Josh Woolley, Andrew Campbell, Alex Cohen, Dror Ben-Zeev, and Trevor Cohen. 2025. Perplexity and proximity: Large language model perplexity complements semantic distance metrics for the detection of incoherent speech. *Journal of Biomedical Informatics* 170 (Oct. 2025), 104899. doi:10.1016/j.jbi.2025.104899

Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. 2025. A Survey on Recent Advances in LLM-Based Multiturn Dialogue Systems. *Comput. Surveys* 58, 6 (2025), 1–38. doi:10.1145/3771090

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, Chunyang Li, Zheyuan Zhang, Yushi Bai, Yantao Liu, Amy Xin, Nianyi Lin, Kaifeng Yun, Linlu Gong, Jianhui Chen, Zhili Wu, Yunjia Qi, Weikai Li, Yong Guan, Kaisheng Zeng, Ji Qi, Hailong Jin, Jinxin Liu, Yu Gu, Yuan Yao, Ning Ding, Lei Hou, Zhiyuan Liu, Bin Xu, Jie Tang, and Juanzi Li. 2024. KoLA: Carefully Benchmarking World Knowledge of Large Language Models. doi:10.48550/arXiv.2306.09296 arXiv:2306.09296 [cs].

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. doi:10.48550/arXiv.1904.09675 arXiv:1904.09675 [cs].

Yihan Zheng, Weixing Tan, Qian Li, Lei Liu, Zhongmin Yan, and Hongjun Dai. 2025. A Hybrid Pipeline and Large Language Model System for Task-Oriented Dialogue. In *2025 IEEE International Conference on High Performance Computing and Communications (HPCC)*. IEEE, Exeter, United Kingdom, 391–398. doi:10.1109/HPCC67675.2025.00069

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a Unified Multi-Dimensional Evaluator for Text Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics,

Abu Dhabi, United Arab Emirates, 2023–2038.
doi:10.18653/v1/2022.emnlp-main.131

# 10. Appendix

## 10.1. Prompts and Code

Listing 1: Custom Metric Prompt

```
STATE_INFORMATION_REQ ="""- **State Information Requirement:** Extract and
    present only the information explicitly provided in Triple 1. Do not add
     any details beyond what is stated."""
CONFIRM_CLARIFY_REQ = """- **Sentence Start Requirement:** The sentence **
    must** begin with either **'Ja,'** or **'Nein,'**âĂŤthis is **crucial**!
- **Sentence Content Requirement:** After the required start, extract and
    present the information explicitly provided in Triple 1."""
GENERATE_COMPARISON_REQ ="""- **Comparison Structure Requirement:** Compare
     and extract information from both triples, ensuring that both specified
     domains are mentioned explicitly in the sentence.
Comparison Information Requirement:** Extract and present only the
    information explicitly provided in Triple 1 or Triple 2. Do not add any
    details beyond what is stated."""

PROMPT_TEMPLATE = """Please evaluate the following German sentence
    according to the requirements and scoring guidelines.

### CONTEXT:
- **Triple 1:** {triple1} (Domain: \textit{UNO} Cardgame)
{context}

### SENTENCE TO EVALUATE:
{sentence}
### END SENTENCE

### SPECIFIC REQUIREMENTS:
{requirements}

### ADDITIONAL REQUIREMENTS:
- Synonyms and reformulations are acceptable as long as the original
    meaning is preserved.
- Use domain-specific terminology appropriately.
- Carefully evaluate all requirements!

### SCORING GUIDELINES:
- **Score 0:** The requirements are correctly fulfilled, and the sentence
    contains only information directly stated in the triple(s) without any
    hallucinations.
- **Score 1:** The requirements are not fulfilled; the sentence includes
    deviations or additional unsupported details (hallucinations) not
    present in the triple(s).

### REQUIRED OUTPUT FORMAT:
{{
  "score": <integer 0 or 1>,
  "reasoning": <reasoning for score>
}}
"""
```

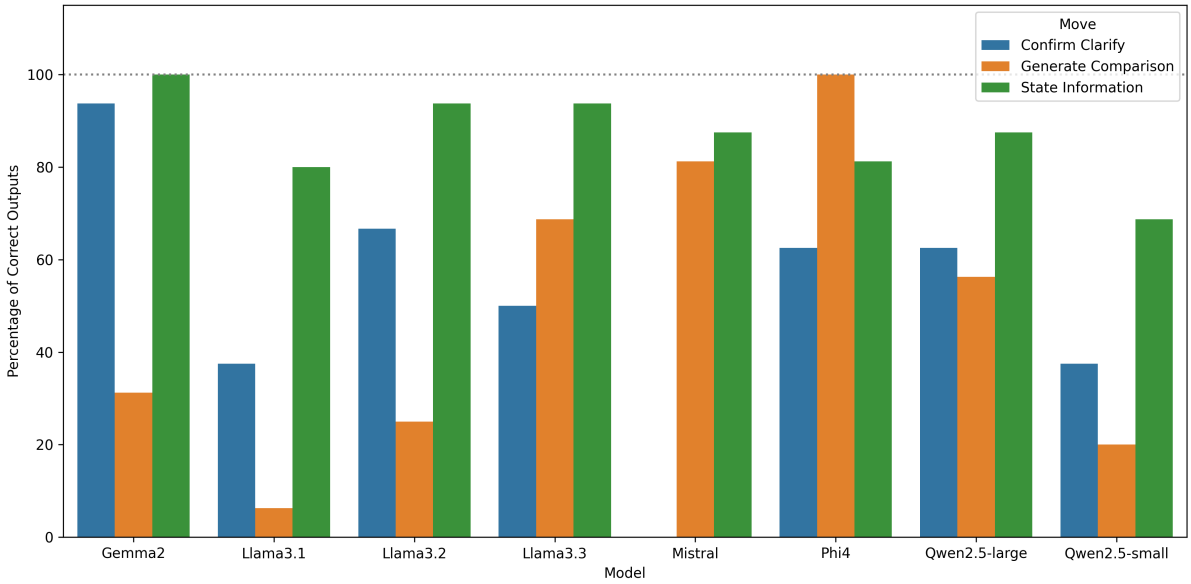


Figure 8: Hallucination metric results for baseline.

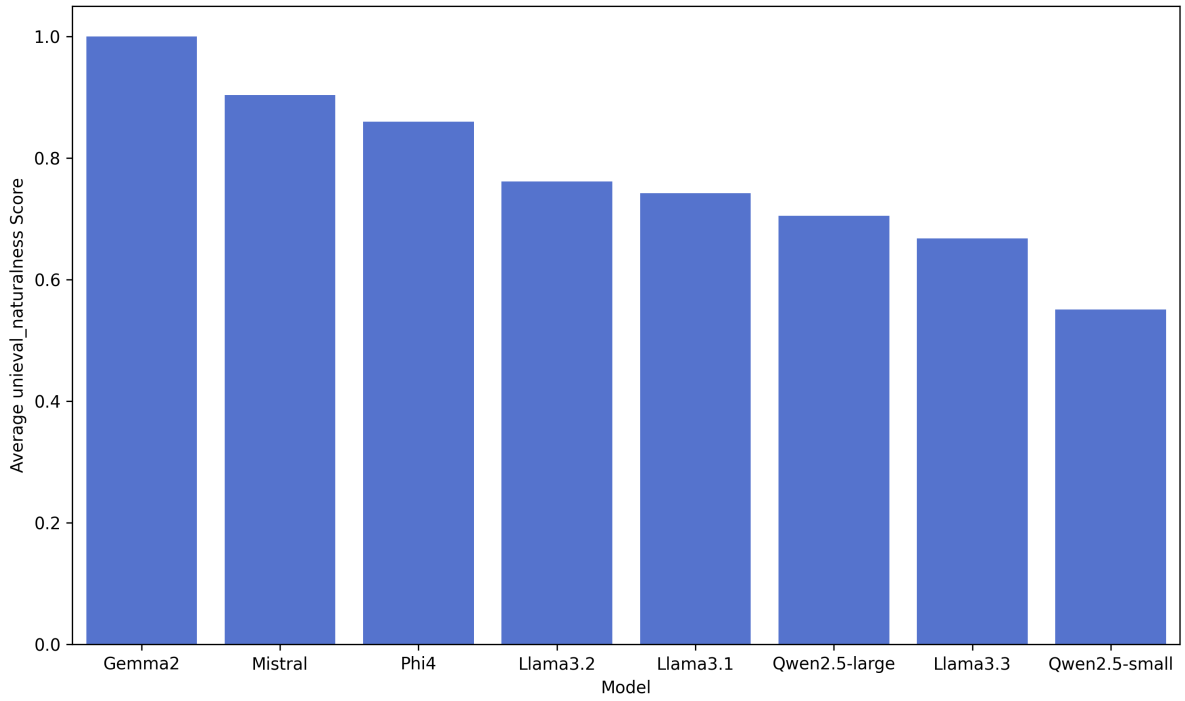


Figure 9: UniEval - Naturalness metric results for baseline.

## 10.2. Additional Figures

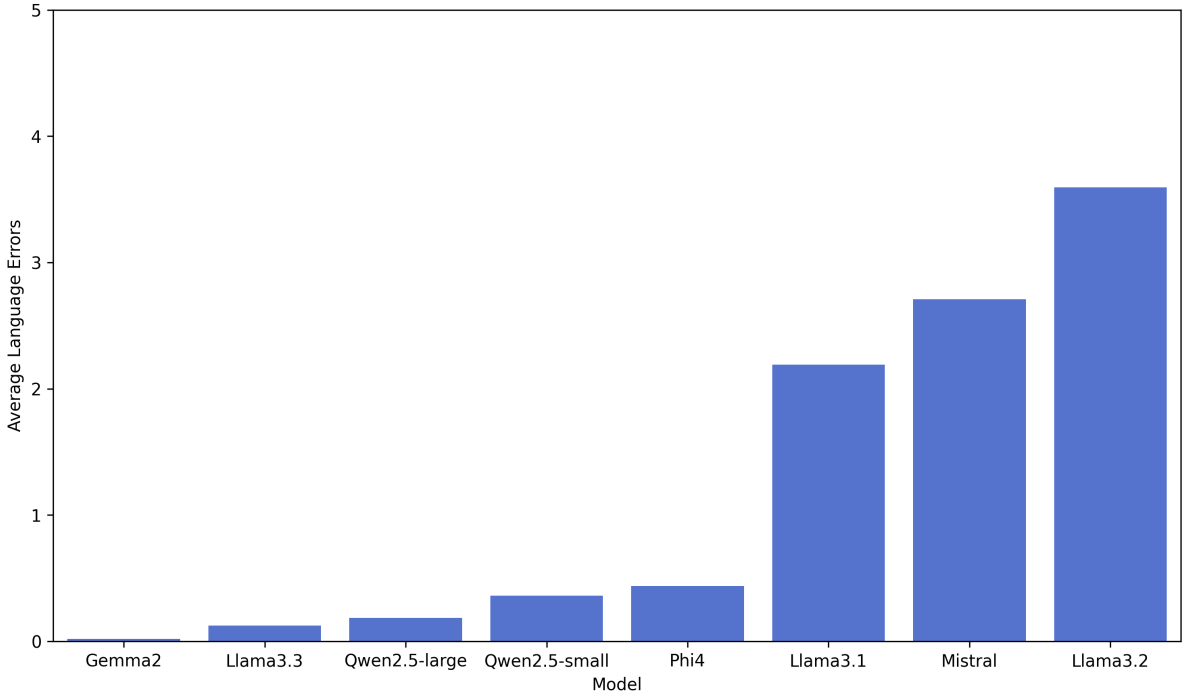


Figure 10: Language error results for baseline.

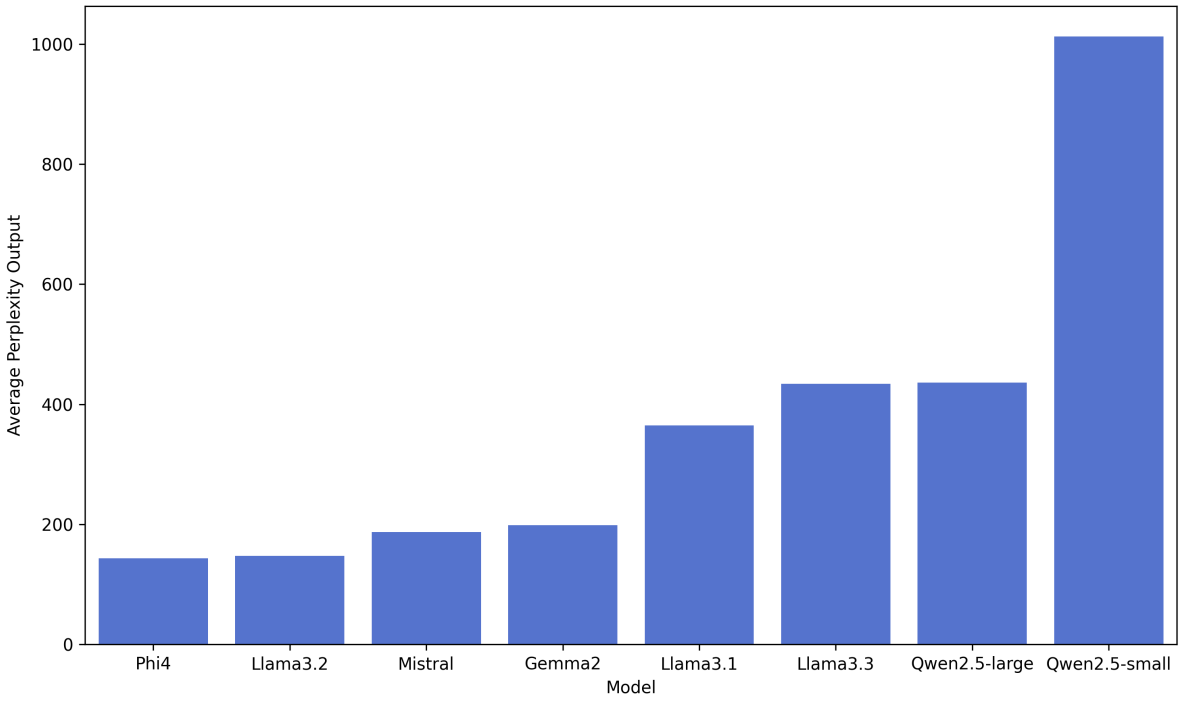


Figure 11: Perplexity results for baseline.

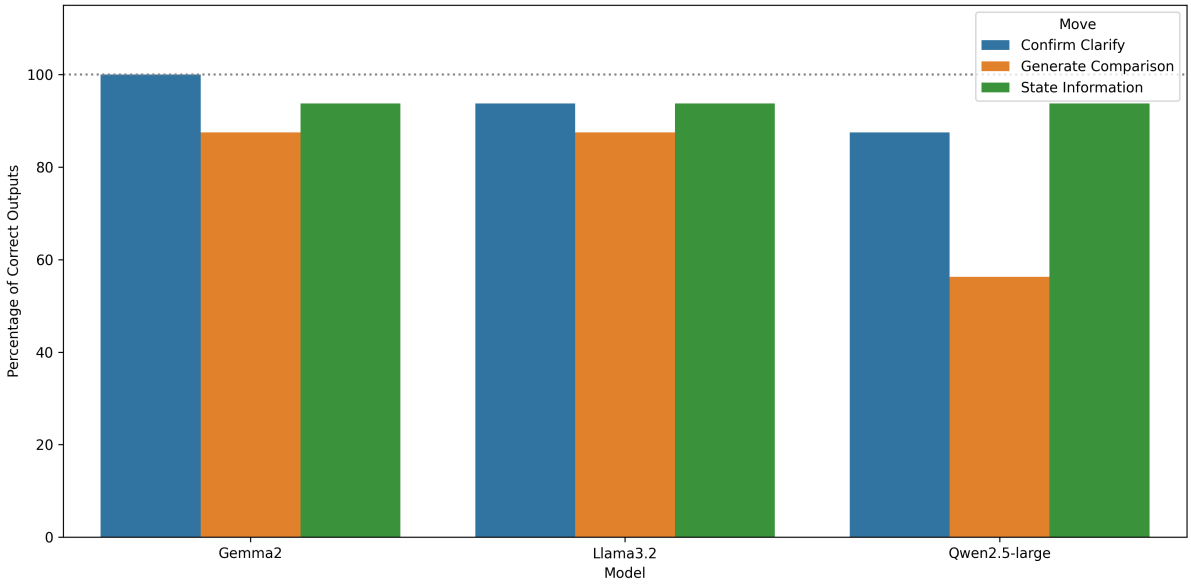


Figure 12: Hallucination metric results for dspy.

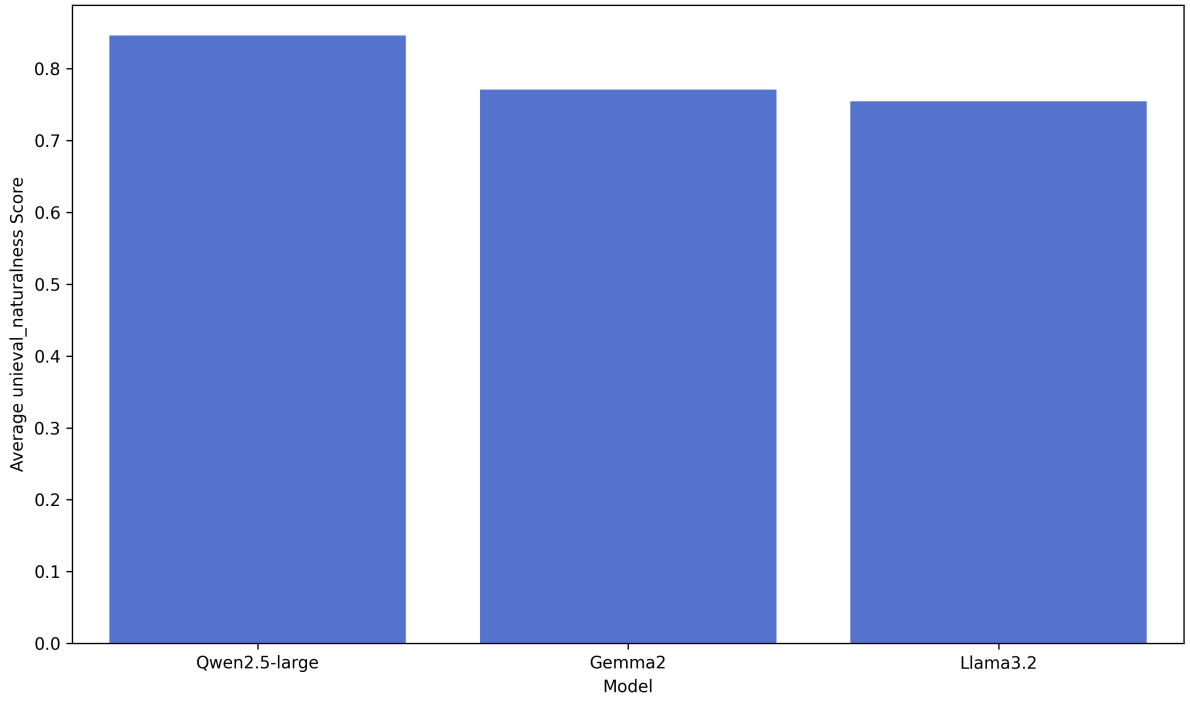


Figure 13: UniEval - Naturalness metric results for dspy.

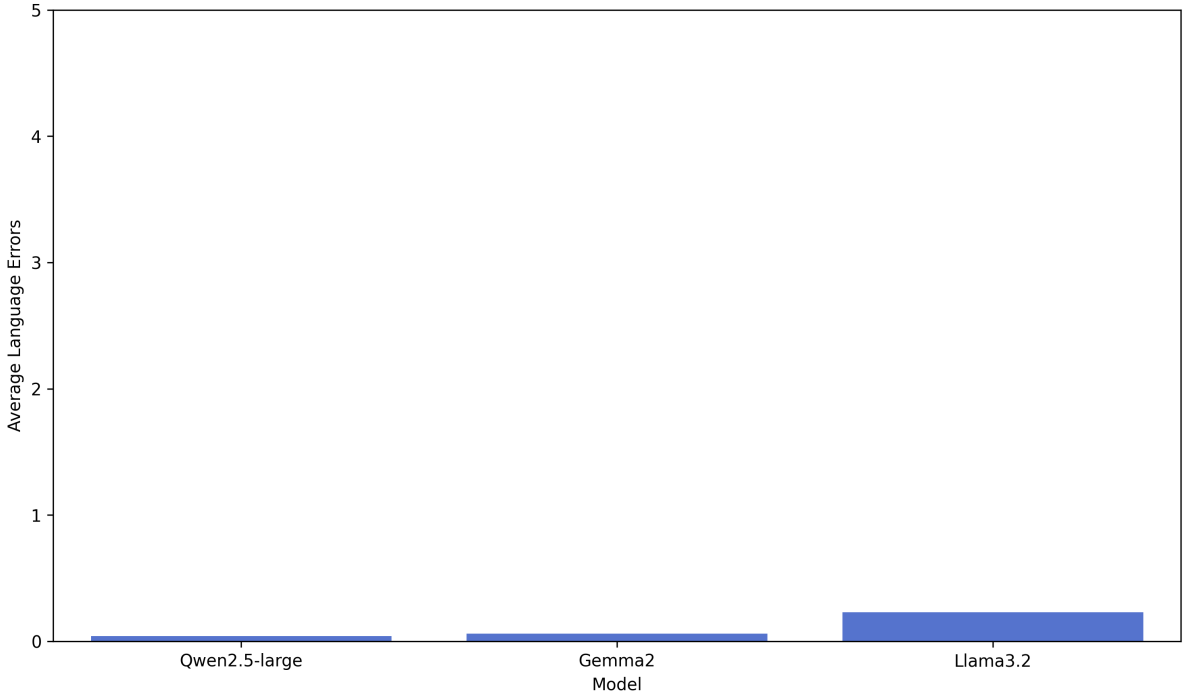


Figure 14: Language error results for dspy.

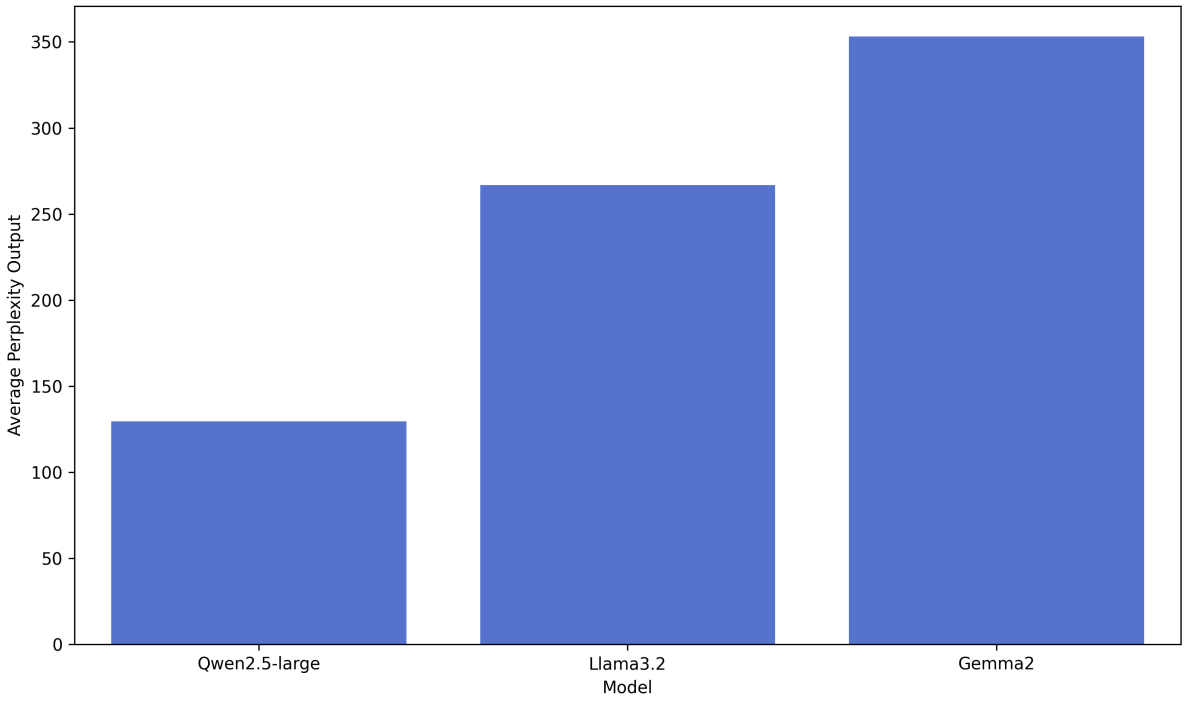


Figure 15: Perplexity results for dspy.

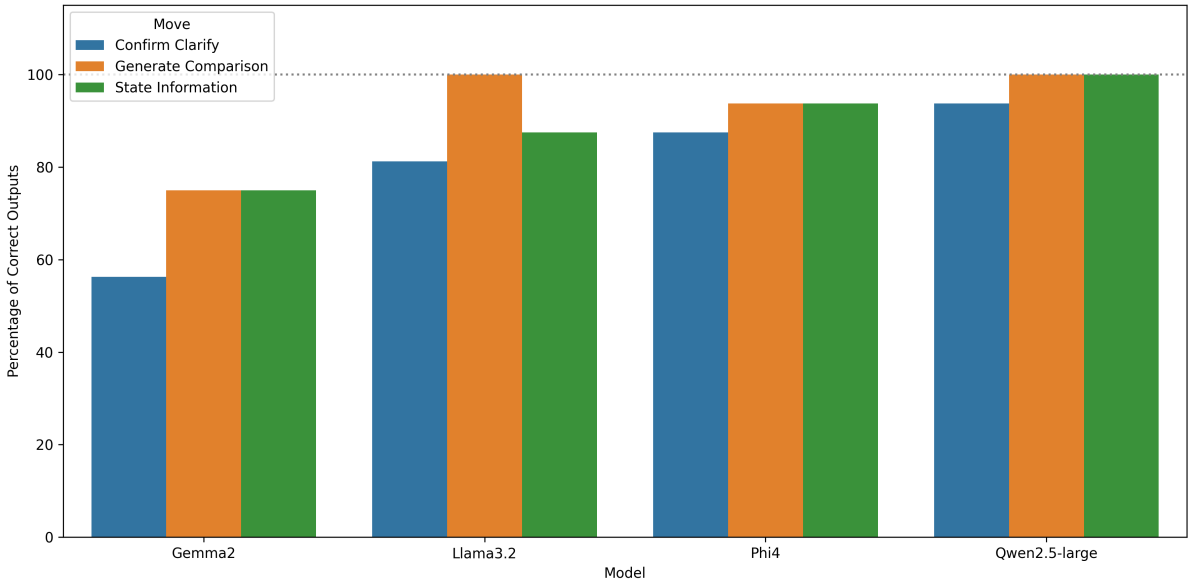


Figure 16: Hallucination metric results for finetuned.

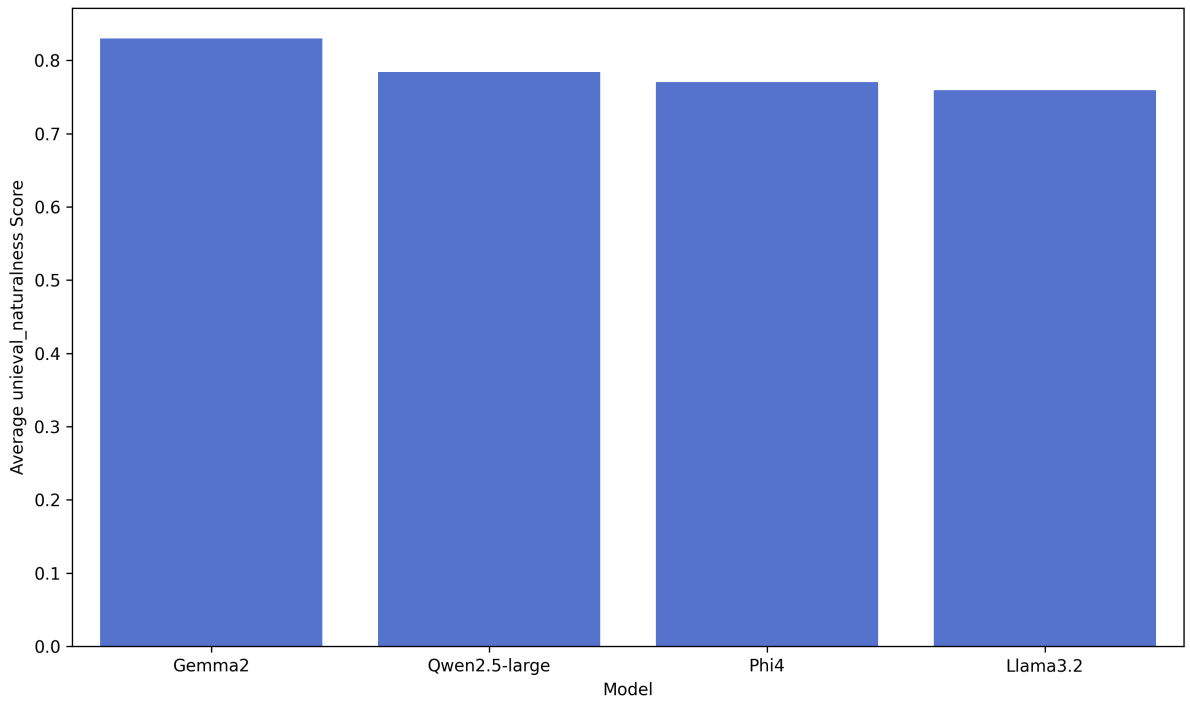


Figure 17: UniEval - Naturalness metric results for finetuned.

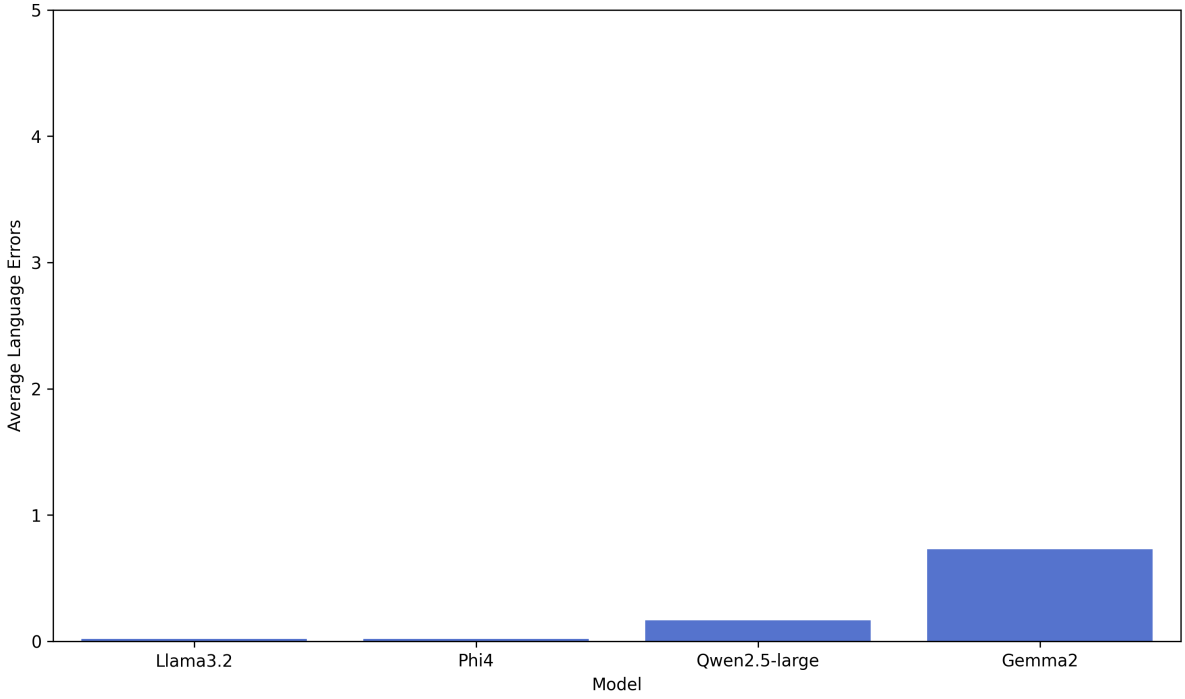


Figure 18: Language error results for finetuned.

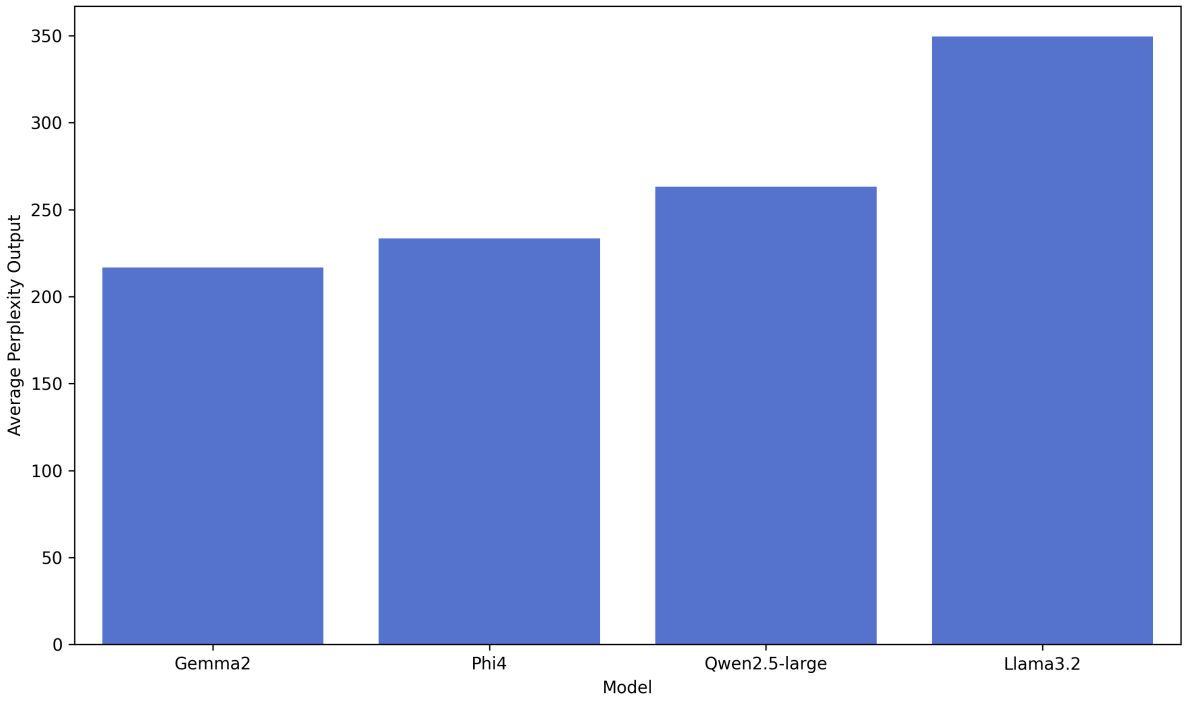


Figure 19: Perplexity results for finetuned.