

Automated Factual Benchmarking for In-Car Conversational Systems using Large Language Models

Rafael Giebisch¹ and Ken E. Friedl² and Lev Sorokin³ and Andrea Stocco⁴

Abstract—In-car conversational systems bring the promise to improve the in-vehicle user experience. Modern conversational systems are based on Large Language Models (LLMs), which makes them prone to errors such as hallucinations, i.e., inaccurate, fictitious, and therefore factually incorrect information. In this paper, we present an LLM-based methodology for the automatic factual benchmarking of in-car conversational systems. We instantiate our methodology with five LLM-based methods, leveraging ensembling techniques and diverse personae to enhance agreement and minimize hallucinations. We use our methodology to evaluate CarExpert, an in-car retrieval-augmented conversational question answering system, with respect to the factual correctness to a vehicle’s manual. We produced a novel dataset specifically created for the in-car domain, and tested our methodology against an expert evaluation. Our results show that the combination of GPT-4 with the Input Output Prompting achieves over 90% factual correctness agreement rate with expert evaluations, other than being the most efficient approach yielding an average response time of 4.5s. Our findings suggest that LLM-based testing constitutes a viable approach for the validation of conversational systems regarding their factual correctness.

I. INTRODUCTION

Researchers and companies are exploring the potential of Large Language Models (LLMs) to tackle complex tasks across diverse domains, such as content quality classification of Q&A websites [1], code translation [2], security vulnerability detection [3] and autonomous driving [4], [5], [6].

In the automotive domain, LLMs can be utilized to develop modern in-car conversational systems. These systems enhance driver and passenger experiences by enabling natural, context-aware interactions for navigation, entertainment, and vehicle control [7]. These systems are expected to provide real-time information, adapt to user preferences, thereby reducing the need for manual inputs, and enhancing overall driving experience and safety [8]. However, despite their potential advantages, LLM-based in-car conversational systems also present critical development challenges in terms of factual correctness, latency, and privacy.

In this paper, we target the automated testing of factual correctness of answers provided by a LLM-based conversational system, as it represents the vital requirement to be satisfied and thoroughly tested [9] (i.e., the system should

respond with factually accurate information), without which such systems would be hardly accepted in production [10]. Particularly, in this paper we focus on the evaluation of CarExpert [11], an in-car conversational system developed at BMW for quality assessment. Among the many features, CarExpert is designed to engage drivers in natural, multi-turn conversations about the vehicle and its features (Figure 1). The system’s responses are informed by data derived from the owner’s manual and relevant documents, including press articles, which have been parsed, annotated by domain experts, embedded, and stored in a vector database to serve as ground truth for CarExpert.

In literature, there exists several *manual* approaches to help minimize hallucinations and factual inaccuracies in conversational system responses [12], [13], [14]. However, manually debugging these defects is generally impractical for engineers due to the extensive domain knowledge and time required for a comprehensive inspection.

This paper investigates the problem of building a black-box testing framework for the automated benchmarking of the factual correctness of in-car conversational-systems, such as CarExpert [11], to reduce the manual workload for engineers. The black-box approach is necessary as these systems are often developed, fully or partially, by third parties. Our approach leverages LLMs, due to their impressive performance in natural language processing, while minimizing hallucinations through techniques such as ensembling, diverse personae, and majority voting.

In our study, we evaluate the property of factual correctness using two dimensions, namely factual consistency and factual relevance. In the context of CarExpert, a system response is considered as *factually consistent* if it fully aligns with the information in the owner’s manual. The owners’ manual serves as a carefully curated dataset and source of ground truth. No additional information should be introduced, assumed, misinterpreted, taken out of context, or fabricated in the response to the user. A system response is *factually relevant*, if it is adequately addressing the user’s question. A response can be factually consistent, but not relevant, and vice versa (see Figure 1).

In this paper, we first curated a dataset of question and answer pairs for the automotive domain, where questions were generated by humans, while answers were provided by CarExpert. Domain experts performed then a factual correctness evaluation on the answers. We then selected five different types of LLMs and reasoning methods, each with custom prompt templates. Each prompt received the generated answer by CarExpert and the relevant paragraphs

¹Rafael Giebisch is with the Technical University of Munich, Munich, Germany rafael.giebisch@tum.de

²Ken E. Friedl is with the BMW Group, Munich, Germany ken.friedl@bmw.de

³Lev Sorokin is with the BMW Group and the Technical University of Munich, Munich, Germany lev.sorokin@tum.de

⁴Andrea Stocco is with the Technical University of Munich and fortiss GmbH, Munich, Germany andrea.stocco@tum.de

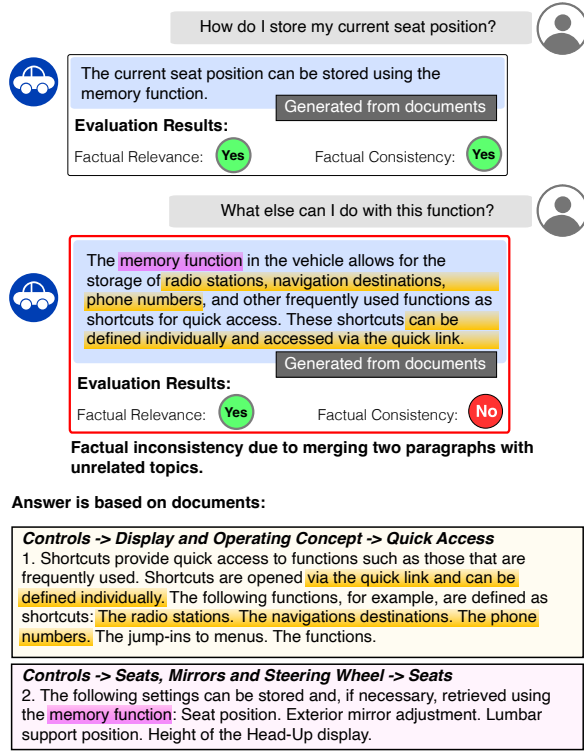


Fig. 1: Example dialogues showing both positive (top) and negative (bottom) interactions between a user and CarExpert. In the second response, CarExpert fails to combine documents from the manual to provide a correct response.

retrieved from the manual by CarExpert as input. The prompt is then passed to our framework LLMs to evaluate the degree of factual correctness.

We assessed the effectiveness of our framework by comparing our systems factual correctness evaluation responses with those collected using human experts. Additionally, we analyzed its efficiency to determine its suitability for real-time use.

Our results show that combining multiple LLMs and methods yields over 90% accuracy in both factual relevance and consistency. GPT-4, utilizing the Input-Output Prompting method, achieved the best tradeoff between effectiveness and efficiency, with accuracy scores of 90% (relevance) and 92% (consistency), and an average execution time of 4.5 seconds per request.

II. TESTING CONVERSATIONAL SYSTEMS

A. Conversational Question Answering Systems

At BMW, CarExpert is an LLM-based question answering system. It can engage with the user in natural, human-like, multi-turn conversations about the car and its functionalities. Its data and ground truth are based on the contents of the owner's manual along with related documents such as press articles. These documents have been parsed and annotated by domain experts, embedded and stored in a vector database.

In this work, we aim to evaluate the answer generation capabilities of CarExpert, which we describe next. Given

a user utterance, CarExpert retrieves in-car-domain-specific relevant documents which may include the potential answer. The system uses Retrieval Augmented Generation (RAG) for information retrieval and answer generation. Specifically, the modular architecture of CarExpert includes four primary sub-components: 1) orchestration, 2) semantic search, 3) answer generation, and 4) answer moderation. To improve factually correct system utterance generation, CarExpert includes control mechanisms on three modules: 1) it uses an input filter as part of the orchestrator, 2) it controls the answer generation by prompt design and 3) it filters it through an output filter. A heuristic is also applied to select the optimal answer, prioritizing both a close match to the user's question and maintaining as much of the original wording as possible to ensure factual accuracy in information extraction and generation. Further, CarExpert integrates text-to-speech and speech-to-text translation, enabling seamless voice-based interaction. For a comprehensive description, we refer the reader to the relevant literature [11].

The use of the retrieved documents for the generation of the output is a critical step which can introduce inconsistencies. For this reason, a validation approach is required to assess whether generated responses are factually consistent and relevant. In the rest of this section, we describe our testing methodology to tackle this task.

B. LLM-based Testing Approach

Figure 2 (left) shows the high level architecture of our testing method. The input to our testing method is a user utterance (e.g., a question), whereas the output is a system utterance. The system utterance is represented as textual description of the factual correctness of the original LLM response. The user utterance is passed to CarExpert, which retrieves first a list of top-k documents (right boxes) and generates an answer. Then, these artifacts are forwarded with the user utterance to our testing framework to evaluate the factual consistency of CarExpert's responses.

Our testing framework is based on LLMs. In this study we evaluated five distinct LLM types and reasoning approaches, for which we designed customized prompt templates. Examples of prompt templates are given in Table I (limited to two relevant methods, due to space constraints).

In the remaining of this section, we give an overview of the LLM-based methods used in our approach.

1) *Input-Output Prompting*: Input-output (IO) prompting, also called *standard prompting* [15], is a straightforward method to query a LLM. A query x is wrapped with predefined task instructions to obtain the system's evaluation [16]. A more refined version consists in providing examples (few-shot prompting), which has been shown to help the LLM to better understand contextual, domain-specific information, resulting in improved accuracy [17].

2) *Chain-of-Thought Prompting*: Chain-of-thought (CoT) prompting introduces intermediate thinking steps to better bridge the gap between the input and output. Aside from simple mathematical problems [16], in practice there is no consolidated definition of what constitutes a step for complex

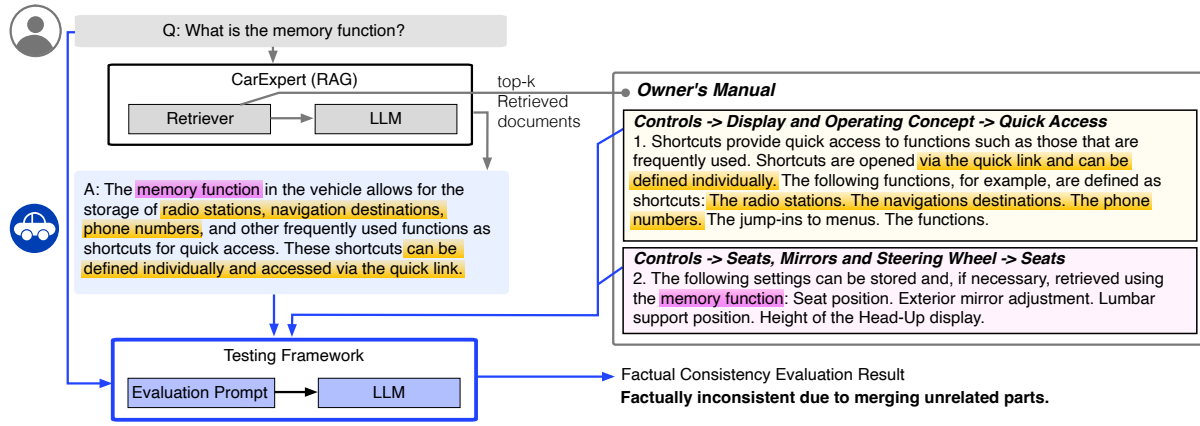


Fig. 2: Overall process of our testing framework for the evaluation of factual correctness of in-car conversational systems.

TABLE I: Examples of prompt templates for factual consistency for IO and RT.

Method	Prompt Template
Input-Output Prompting	Setting: a service to help drivers in their car. Consider following sentences: <retrieved-documents>. Is the text: <generated-answer> factually consistent with the sentences? All information contained in the text has to appear in the sentences. No additional information must be added or assumed. Return your answer in the following format: <output-format>.
	Other agents have already returned their evaluation. Check, whether you agree with them or not.
Round-Table Conference (2nd round only, simplified)	Here are their evaluations in JSON format: <evaluations>. Return your answer in the following format: <output-format>.

domains such as ours. Nevertheless, CoT has been shown to provide accurate results in several domains [18] by providing examples of how a human would sequentially reason about problems. In our case, we request in the prompt to iterate over text units of retrieved documents and build up an argumentation chain.

3) *Self-Consistency with Chain-of-Thought*: Chain-of-thought prompting and similar methods create chains autoregressively in one step. This can lead to error propagation as the number of steps increases due to compound errors [19]. Self-Consistency mitigates this issue, by sampling CoT k-times and then returning the most frequent output. This has been shown to work if the output space is limited. If the outputs are too dissimilar, no output would be generated

twice (or more) [16].

4) *Multi-Persona Self-Collaboration*: Multi-Persona Self-Collaboration (MPSC) dynamically assigns and simulates multiple personas for a task, instructing the LLM to adopt a specific identity. As shown by Xu et al. [20], the restriction to a particular way of thinking may enhance performance. Personas iteratively generate and critique solutions from other personas, refining them through multiple rounds to achieve consensus. This method, called Solo Performance Prompting (SPP), operates in a pure zero-shot manner [21]. In our framework we defined first the personas *Fact Checker*, *Research Analyst*, *Editor*, *Journalist*, *Librarian* and questioned the LLM Claude 3 Haiku [22] do describe the personas. We assume, that these personas possess qualifications which are necessary for a critical review of factual consistency.

5) *Round Table Conference*: While previous methods rely on a single large language model to complete the task, using multiple LLMs—often referred to as agents—offers the potential for even better performance. The approach involves allowing each agent to independently generate its own reasoning and then engage in a collaborative discussion until reaching a consensus.

We use a method similar to ReConcile [23], a Round Table (RT) Conference approach, which adds additional weighting of the responses, based on the returned confidence estimations of the agents, as well as more elaborate examples. If all agents agree on a common evaluation, or if the maximum number of rounds has been reached, the Round Table Conference ends and returns the common answer, otherwise the next round is initiated. In our preliminary experiments, we determined that a limit of five agents was sufficient to produce reliable results, which is why we selected it as our threshold.

III. CASE STUDY AT BMW

A. Research Questions

We consider the following research questions.

RQ₁ (effectiveness): *How accurate is our framework in assessing factual relevance and consistency?*

RQ₂ (efficiency): *How efficient is our framework?*

The first research question investigates the effectiveness of our framework. We evaluate different configurations and influencing factors such as the LLM instance, method or temperature used. The second research question targets the evaluation of the efficiency of different methods to investigate which methods are suitable for real-time processing.

B. Metrics

To evaluate the effectiveness of our framework (**RQ₁**), we compare the factual relevance and consistency evaluation results output by our framework with an expert-based evaluation. Specifically, we assess the *accuracy* between the evaluation of the human and our proposed framework (as outlined in (1)), similarly to previous studies on the performance of LLM-based systems [24].

$$ACC = \frac{\# \text{ agreements expert and our framework}}{\# \text{ samples}} \quad (1)$$

We apply the metric respectively for both, the factual relevance and consistency dimension. To date, we consider our choice as the best available strategy for evaluating our approach for the following reasons: 1) conventional automated evaluation metrics such as ROUGE, BLEU or other n-gram based metrics fail to accurately analyze semantics, which is the focus of our study. 2) conventional metrics often exhibit a low correlation with human judgments [25]. A description of how the factual relevance and consistency labels are created by an expert is outlined in the following Section III-C.

For **RQ₂**, we recorded the execution time for each request and identified the consumed token number including the number of tokens in the prompt for the evaluation task and the response generated by our system.

C. Dataset Creation

We use the owner’s manual as our ground truth. First, we parse the owner’s manual of a specific BMW SUV into a JSON file and divide its content into paragraphs. We call these *documents*. Overall, we retrieve a total of 4914 documents. We then let BMW experts go through the documents and manually create questions, i.e., user utterances. For each question, there is exactly one document sufficient to address the question. Then, we query CarExpert with the questions from the experts and receive responses, including a list of retrieved documents and the generated answer to our question. After this, we pass the question with the outputs from CarExpert to two different experts, both employed in the Car Manual Quality Assurance department with more than 5 years experience to decide whether the generated answer is 1) consistent with the retrieved document(s), and 2) whether it is relevant. In particular, to achieve a reliable dataset, we randomly repeat questions passed to the first reviewer. If the expert provides inconclusive labeling results when questioned multiple times, we pass the evaluation once to the second expert to decide on the label. This process generates us consistency and relevance labels for each question and answer pair, which we consider as our ground truth.

D. Configurations

We evaluate our framework with five different proprietary and open-weight models, also considering their size in terms of number of parameters. Particularly, for our experiments, we chose widely popular models from OpenAI, such as GPT-3.5-turbo, GPT-4 and their to date latest model GPT-4o. As a representative of open models, we adopted two versions of Meta Llama 3, with 8 billion and 70 billion parameters.

For all LLM-based methods and types as described in Section II-B, we executed the experiments varying four different temperature settings in the range [0.0, 0.2, 0.4, 0.6]. The pre-defined prompts were kept the same for all experiments conducted. Overall, we performed 10,300 evaluations (103 samples $\times$ 5 LLM-based methods $\times$ 5 LLMs $\times$ 4 temperatures).

E. Implementation

Our benchmarking framework is implemented in Python and is using the LangChain¹ library to send evaluation requests, containing the initial users utterance and the output of CarExpert, to the considered LLMs. CarExpert is deployed locally, while the to be tested LLMs for our evaluation are deployed in the Cloud. Specifically, the LLAMA models are deployed in Azure Virtual Machines, while GPT-based models are hosted natively by OpenAI. For the expert evaluation and ground truth data creation we create an online application, where the expert is presented the original user utterance, the answer by CarExpert and retrieved document. The accuracy evaluation comparing the experts results with the ones from our framework is performed locally.

F. Results

Table II reports the results for our research questions **RQ₁**. The table presents the accuracy results, for different temperatures, LLMs, and evaluation methods, for both relevance and consistency. As it is challenging to compare methods whose performance is expressed by multiple conflicting metrics/objectives, i.e., relevance and consistency in our case, we have sorted the results based on Pareto-dominance and highlighted the results in boldface. Nevertheless, the use of our benchmarking framework can also target specifically relevance or consistency, which is why we report in addition our observations and analysis for each metric independently. **RQ₁**. Regarding relevance, the highest score is achieved with GPT-3.5-Turbo with 93.2% at temperature 0, while the corresponding consistency value was around 20%. However, GPT-4 achieves with RT a slightly lower relevance accuracy of 92.2% but a significantly higher consistency score of 90.2% at the same temperature. We encountered the exact opposite result with GPT-4 using IO. Solely results from GPT-3.5 Turbo and GPT-4 based methods pose the set of non-dominated results, where GPT-4 scores do not fall below 89.3%, which is the worst score for consistency at temperature 0.2.

¹<https://www.langchain.com/>

TABLE II: RQ₁: Accuracy results of various LLM and method combinations, evaluated by experts for relevance and consistency across different models and temperature settings (percentage-based). The Pareto-optimal solutions, which are not simultaneously dominated in both relevance and consistency, are highlighted in bold.

	RELEVANCE				CONSISTENCY			
	0.0	0.2	0.4	0.6	0.0	0.2	0.4	0.6
GPT-3.5-turbo								
RT	93.2	93.2	92.2	92.2	21.3	20.3	20.3	21.3
CoT	90.2	89.3	88.3	88.3	22.3	25.2	31.0	29.1
CoT-SC	90.2	89.3	90.2	89.3	22.3	22.3	24.2	23.3
IO	90.2	89.3	92.2	89.3	82.5	81.5	76.6	69.9
MPSC	90.2	91.2	93.2	90.2	72.8	74.7	72.8	69.9
GPT-4								
RT	92.2	92.2	92.2	92.2	90.2	89.3	90.2	91.2
CoT	92.2	91.2	90.2	92.2	89.3	88.3	88.3	90.2
CoT-SC	92.2	92.2	91.2	91.2	89.3	89.3	89.3	89.3
IO	90.2	91.2	89.3	90.2	92.2	91.2	90.2	91.2
MPSC	82.5	77.6	77.6	78.6	82.5	87.3	85.4	84.4
GPT-4o								
RT	91.2	89.3	90.2	90.2	85.4	84.4	87.3	85.4
CoT	78.6	80.5	79.6	80.5	65.0	65.0	64.0	64.0
CoT-SC	78.6	79.6	77.6	79.6	65.0	66.9	65.0	57.2
IO	78.6	78.6	81.5	78.6	79.6	79.6	77.6	78.6
MPSC	70.8	71.8	71.8	71.8	78.6	80.5	78.6	80.5
LLAMA 3 8B								
RT	14.5	15.5	14.5	15.5	82.5	82.5	82.5	84.4
CoT	85.4	85.4	89.3	86.4	52.4	79.6	76.6	73.7
CoT-SC	85.4	85.4	86.4	83.4	79.6	79.6	72.8	76.6
IO	85.4	86.4	85.4	83.4	49.5	71.8	75.7	64.0
MPSC	66.9	66.0	72.8	61.1	55.3	67.9	64.0	66.0
LLAMA 3 70B								
RT	91.2	91.2	91.2	91.2	81.5	22.3	20.3	22.3
CoT	91.2	91.2	91.2	92.2	84.4	84.4	84.4	85.4
CoT-SC	90.2	91.2	91.2	92.2	84.4	85.4	85.4	84.4
IO	91.2	91.2	91.2	91.2	83.4	83.4	83.4	84.4
MPSC	82.5	82.5	82.5	84.4	82.5	85.4	83.4	85.4

Impact of different LLMs/methods. For GPT-4 we did not observe a significant impact of the evaluation method used, only MPSC performed worse for almost all temperatures than the best results of GPT-4 for both relevance and consistency. For GPT-3.5 turbo we observed less variance regarding relevance. However, results diverge more for consistency, which is also the case for GPT-4o and LLAMA 3 8B. For instance, RT exhibits a low consistency score of 21.3% at temperature 0 and 22.3% for GPT-3.5-turbo and LLAMA 3 70B, while methods such as IO achieve scores over 80%.

Regarding consistency, the methods' impact depends on the LLM used. For instance, at temperature 0, the methods CoT, CoT-SC exhibit for GPT-3.5-turbo scores around 20%, while for other LLMs the results are significantly better. Another example is the approach MPSC, which achieved the best consistency scores for GPT-4, GPT-4o and LLAMA 3 70B, while showing the lowest score for Llama 3 8B with 55.3% at temperature 0.

TABLE III: Scores for Round Table Conferences with agents consisting of different LLMs with a temperature of 0.0.

Round Table Agents	Consistency	Relevance
5x GPT-4	90.2	92.2
1x GPT-4, 1x LLAMA 3 70B	89.3	90.2
1x GPT-4, 1x GPT-4o, 1x LLAMA 3 70B	83.4	90.2
2x GPT-4, 1x GPT-4o, 2x LLAMA 3 70B	85.4	89.3

Impact of different temperatures. Regarding the impact of the temperature on the results, for relevance we did not observe a significant variation when increasing the temperature. The highest decrease was for LLAMA 3 8B with MPSC by 5.8%, from temperature 0 to temperature 6.

For consistency, we observed the highest decrease from 81.5% to a round 22% with LLAMA 3 70 B and RT. The highest increase was observed for LLAMA 3 8B for CoT from 52.4% to over 73%. For the other LLMs and methods the scores tend slightly to decrease or increase without a concrete trend.

Multi-Agent Round Table. The former Round Table conference approach used multiple instances of the same LLM. In additional, we evaluated Round Table based reasoning using different agents/LLMs. The idea of using various agents is to encourage divergent thinking, which is expected to improve the results.

We performed additional experiments with the RT approach by combining several of the top-performing LLMs (GPT-4, GPT-4o, and LLAMA 3 70B) in various ensemble configurations at temperature 0. The results, as shown in Table III, however indicate that an ensemble of 5 GPT-4 models achieved the highest relevance and consistency scores, which is similar to the performance of a single GPT-4 model employing RT. Combining agents of different types yielded lower relevance and consistency scores, achieving a higher decrease in consistency than for relevance.

RQ₁ (effectiveness): GPT-4 balanced relevance and consistency best, scoring 92.2% vs. 90.2% with RT and 90.2% vs. 92.2% with IO at temperature 0, with minimal variation across methods. GPT-3.5-Turbo with RT achieved the highest relevance (93.2%), while LLAMA 3B had the lowest (14.5%, with 82% consistency). Generally, relevance scores surpassed consistency, which varied by LLM. GPT-3.5-Turbo's CoT and CoT-SC scored around 20%, whereas other LLMs performed better. Temperature changes had little impact, and using multiple agents for GPT-4 RT reduced relevance more than consistency.

RQ₂. Table V reports the average token usage per evaluation and average time per evaluation for methods on GPT-4. Since LLAMA models were deployed on Azure virtual machines, a direct cost and time comparison with GPT models is not feasible. OpenAI's models, being proprietary and available only as serverless functions, use a distinct pricing and

TABLE IV: Comparison of Owner’s Manual Text and Generated Answers for examples of three categories of error types.

Type 1	Manual: To manually turn on standby state, press and hold the thumbwheel on the center console. Generated: To manually turn on idle state, press and hold the thumbwheel on the center console.
Type 2	Manual: Press and hold the unlock button on the vehicle key after unlocking. The windows open for as long as the button on the vehicle key is pressed. Generated: The welcome window is a feature that allows you to open the windows of the vehicle by pressing and holding the unlock button on the vehicle key after unlocking. [...]
Type 3	Manual: The Lane Change Warning system with active return detects vehicles in your blind spot or vehicles approaching from behind in the adjacent lane. Generated: The Lane Change Warning light illuminates when there is a risk of collision with a vehicle crossing from the left [...] and immediate braking or an evasive maneuver is required.

TABLE V: Overview of execution times, token number for one evaluation, and average time per token.

Method	Tokens	Execution Time (s)	Average Time (s/token)
RT	3924	23.7	0.0060
MPSC	2376	17.8	0.0074
CoT	1427	5.5	0.0038
CoT-SC	4281	16.5	0.0039
IO	689	4.5	0.0065

compute structure. We assume that the relative differences between methods are consistent for LLAMA models as well. The best efficiency regarding tokens per time achieved CoT, followed by CoT-SC and RT. Regarding the evaluation time for one request, IO was the fastest method.

RQ₂ (efficiency): The most efficient method in terms of time required per token was CoT (6.5ms/token), followed by CoT-SC and RT. However, when considering the evaluation time for a single request, IO demonstrated the best performance with 4.5s per request compared to 23.7s for CoT.

G. Qualitative Analysis

We analyzed our results to identify specific failure patterns and gain insight as of why our methods did not achieve 100% accuracy for factual relevance and consistency. However, a manual error analysis of the over 23,000 individually generated evaluations would be infeasible, thus we only analyzed the results of the best run of each of the methods. Since relevance tends to be assessed on an intuitive basis, we focus our analysis on factual consistency, as this is based on facts and can therefore be analyzed more objectively. An error is identified when the expert opinion was positive but the implemented methods returned negative results, or vice versa. We ignore errors regarding failed requests, formatting difficulties as they only refer to errors in the results themselves. We identified three categories of errors [26], described next. None of the LLM techniques used in our framework succeeded in adequately answering the questions for the examples in Table IV.

Error Type 1: Confusion of Internal Terminology

One example for error type 1 is the response to a user utterance about the “standby state” as given in Table IV. CarExpert used information about the “idle state”, although this is a different state. The evaluation stated the statement as factual consistent, although the statement is incorrect.

Without a domain-specific understanding of these terms, completing this request is challenging for the system. During LLM model training, terms like “standby” and “idle” may have been treated as synonyms. However, within a brand-specific context, these terms represent distinct functions. LLMs may also struggle with the company’s evolving terminology, as new functions and names are regularly introduced.

Error Type 2: Hallucinations

Error of type 2 involves hallucinations. Hallucinations occur when CarExpert attempts to respond despite retrieving documents from the user manual that either did not address the question or provided inadequate answers. For instance, a question regarding the “welcome window” as shown in Table IV led to an imagined connection with a retrieved document from the owner’s manual, incorrectly associating it with the requested function.

The “welcome window” is a small greeting message on the infotainment, but the CarExpert system used other information here and interpreted that the window in the given extract from the owner’s manual is the ‘welcome window’, although not referenced.

Error Type 3: Common Sense Errors

The final type of error involves a lack of common sense. While it could be considered as a subset of hallucinations, we distinguish it because additional and specialized knowledge the mistake is unlikely to be made by a human. As an example, consider the term “Lane Change Warning light”. As the name implies, this light alerts the driver to a lane change if there is already a vehicle in an adjacent lane.

While the system’s purpose is obviously clear, the CarExpert’s response—to recommend immediate braking—is wrong and safety-critical.

IV. RELATED WORK

Several papers have already outlined LLM-based evaluation methods [27], [28], [29] and its advantages [30], but

these work focus on text summarization. In contrast, our method focuses on evaluating in-car conversational systems in the automotive domain for factual correctness.

Recently, researchers have proposed using LLMs to evaluate other LLMs [31] (i.e., “LLM-as-a-Judge”). The setup includes several LLMs, which generate answers to predefined, open-ended questions. After this, experts are used to evaluate and label the generated answers. The LLMs judges are then given the same question-answer pairs, which they will also have to evaluate on their own. In this experiment the LLM judgements reached an agreement of over 80% with humans, proving a promising way of evaluating LLM generated answers. Explicitly analyzing the factual correctness and consistency of the generated output has been neglected, going even as far as mentioning safety has been disregarded in the analysis, which is a requirement for in-car usage [31].

In the automotive domain, Friedl et al. [32] describes the evaluation of in-car conversational systems using LLMs. Multiple personas were created to let the LLM judge the quality of the generated output. However, the evaluation only included metrics like “follow-up”, “implicit understanding” and “harmful user input”, neglecting yet again factual correctness, factual consistency and relevance of the answer [32]. Our work, on the other hand, focuses on the correctness of the main functionality of conversational systems, thus it complements existing research.

V. CONCLUSIONS

In this work, we have proposed a validation approach using LLMs to test an in-car conversational system, focusing on factual correctness, spanning the dimensions of factual relevance and factual consistency. We have created a human-based curated dataset specific for the in-car domain to evaluate our approach. Our evaluation of five advanced LLM types and methods shows that multiple models achieve over 90% accuracy in relevance and consistency across various hyperparameter settings. The best results regarding the tradeoff between relevance, consistency, and efficiency where achieved with GPT-4 employing the Input-Output prompting. Overall, our study shows the potential of LLM-based methods for the automated evaluation of factual correctness in conversational question answering systems and intends to encourage further studies.

In our future work we want to investigate approaches for the synthetic data generation creating pairs of questions and source documents, thereby avoiding expensive expert-crafted questions. Further, we will investigate the performance of our approach when employing other languages or vehicle types.

REFERENCES

- [1] P. Y. P. Chan and J. Keung, “Validating pretrained language models for content quality classification with semantic-preserving metamorphic relations,” *Natural Language Processing Journal*, vol. 9, p. 100114, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2949719124000621>
- [2] R. Pan, A. R. Ibrahimzada, R. Krishna, D. Sankar, L. P. Wassi, M. Merler, B. Sobolev, R. Pavuluri, S. Sinha, and R. Jabbarvand, “Lost in translation: A study of bugs introduced by large language models while translating code,” in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ser. ICSE ’24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: <https://doi.org/10.1145/3597503.3639226>
- [3] B. Boi, C. Esposito, and S. Lee, “Smart contract vulnerability detection: The role of large language model (llm),” *SIGAPP Appl. Comput. Rev.*, vol. 24, no. 2, p. 19–29, Aug. 2024. [Online]. Available: <https://doi.org/10.1145/3687251.3687253>
- [4] L. Chen, O. Sinavski, J. Hünemann, A. Karnsund, A. J. Willmott, D. Birch, D. Maund, and J. Shotton, “Driving with llms: Fusing object-level vector modality for explainable autonomous driving,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 093–14 100.
- [5] Y. Cui, S. Huang, J. Zhong, Z. Liu, Y. Wang, C. Sun, B. Li, X. Wang, and A. Khajepour, “Drivellm: Charting the path toward full autonomous driving with large language models,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1450–1464, 2024.
- [6] K. Tong and S. Solmaz, “Connectgpt: Connect large language models with connected and automated vehicles,” in *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 581–588.
- [7] S.-C. Lin, C.-H. Hsu, W. Talamonti, Y. Zhang, S. Oney, J. Mars, and L. Tang, “Adasa: A conversational in-vehicle digital assistant for advanced driver assistance features,” in *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 531–542. [Online]. Available: <https://doi.org/10.1145/3242587.3242593>
- [8] H. Du, X. Feng, J. Ma, M. Wang, S. Tao, Y. Zhong, Y.-F. Li, and H. Wang, “Towards proactive interactions for in-vehicle conversational assistants utilizing large language models,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.09135>
- [9] V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, and P. Tonella, “Testing Machine Learning based Systems: A Systematic Mapping,” *Empirical Software Engineering*, 2020.
- [10] I. H. Sarker, “Llm potentiality and awareness: a position paper from the perspective of trustworthy and responsible ai modeling,” *Discover Artificial Intelligence*, vol. 4, no. 1, May 2024. [Online]. Available: <http://dx.doi.org/10.1007/s44163-024-00129-0>
- [11] M. R. A. H. Rony, C. Suess, S. R. Bhat, V. Sudhi, J. Schneider, M. Vogel, R. Teucher, K. Friedl, and S. Sahoo, “CarExpert: Leveraging large language models for in-car conversational question answering,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, M. Wang and I. Zitouni, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 586–604. [Online]. Available: <https://aclanthology.org/2023.emnlp-industry.56>
- [12] J.-Y. Yao, K.-P. Ning, Z.-H. Liu, M.-N. Ning, Y.-Y. Liu, and L. Yuan, “LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples,” 2024. [Online]. Available: <https://arxiv.org/abs/2310.01469>
- [13] S. Banerjee, A. Agarwal, and S. Singla, “LLMs Will Always Hallucinate, and We Need to Live With This,” 2024. [Online]. Available: <https://arxiv.org/abs/2409.05746>
- [14] Z. Zhang, Y. Wang, C. Wang, J. Chen, and Z. Zheng, “LLM Hallucinations in Practical Code Generation: Phenomena, Mechanism, and Mitigation,” 2024. [Online]. Available: <https://arxiv.org/abs/2409.20550>
- [15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” 2023. [Online]. Available: <https://arxiv.org/abs/2201.11903>
- [16] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, “Tree of thoughts: Deliberate problem solving with large language models,” 2023. [Online]. Available: <https://arxiv.org/abs/2305.10601>
- [17] H. Ratnayake and C. Wang, “A prompting framework to enhance language model output,” in *AI 2023: Advances in Artificial Intelligence*, T. Liu, G. Webb, L. Yue, and D. Wang, Eds. Singapore: Springer Nature Singapore, 2024, pp. 66–81.
- [18] Z. Zhang, A. Zhang, M. Li, and A. Smola, “Automatic chain of thought prompting in large language models,” 2022. [Online]. Available: <https://arxiv.org/abs/2210.03493>
- [19] A. Zhou, K. Yan, M. Shlapentokh-Rothman, H. Wang, and Y.-X. Wang, “Language agent tree search unifies reasoning acting and planning in language models,” 2024. [Online]. Available: <https://arxiv.org/abs/2310.04406>

- [20] B. Xu, A. Yang, J. Lin, Q. Wang, C. Zhou, Y. Zhang, and Z. Mao, "Expertprompting: Instructing large language models to be distinguished experts," 2023. [Online]. Available: <https://arxiv.org/abs/2305.14688>
- [21] Z. Wang, S. Mao, W. Wu, T. Ge, F. Wei, and H. Ji, "Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration," 2024. [Online]. Available: <https://arxiv.org/abs/2307.05300>
- [22] "The claude 3 model family: Opus, sonnet, haiku." [Online]. Available: <https://api.semanticscholar.org/CorpusID:268232499>
- [23] J. C.-Y. Chen, S. Saha, and M. Bansal, "Reconcile: Round-table conference improves reasoning via consensus among diverse llms," 2024. [Online]. Available: <https://arxiv.org/abs/2309.13007>
- [24] O. Honovich, R. Aharoni, J. Herzig, H. Taitelbaum, D. Kukliansy, V. Cohen, T. Scialom, I. Szpektor, A. Hassidim, and Y. Matias, "True: Re-evaluating factual consistency evaluation," 2022. [Online]. Available: <https://arxiv.org/abs/2204.04991>
- [25] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-eval: Nlg evaluation using gpt-4 with better human alignment," 2023. [Online]. Available: <https://arxiv.org/abs/2303.16634>
- [26] N. Humbatova, G. Jahangirova, G. Bavota, V. Riccio, A. Stocco, and P. Tonella, "Taxonomy of Real Faults in Deep Learning Systems," in *Proceedings of 42nd International Conference on Software Engineering*, ser. ICSE'20. New York, NY, USA: ACM, 2020, p. 12 pages.
- [27] C.-H. Chiang and H. yi Lee, "Can large language models be an alternative to human evaluations?" 2023. [Online]. Available: <https://arxiv.org/abs/2305.01937>
- [28] Y.-T. Lin and Y.-N. Chen, "Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models," 2023. [Online]. Available: <https://arxiv.org/abs/2305.13711>
- [29] O. Honovich, L. Choshen, R. Aharoni, E. Neeman, I. Szpektor, and O. Abend, " q^2 : Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7856–7870. [Online]. Available: <https://aclanthology.org/2021.emnlp-main.619>
- [30] T. Hosking, P. Blunsom, and M. Bartolo, "Human feedback is not gold standard," 2024. [Online]. Available: <https://arxiv.org/abs/2309.16349>
- [31] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging llm-as-a-judge with mt-bench and chatbot arena," 2023. [Online]. Available: <https://arxiv.org/abs/2306.05685>
- [32] K. E. Friedl, A. G. Khan, S. R. Sahoo, M. R. A. H. Rony, J. Germies, and C. Süß, "Inca: Rethinking in-car conversational system assessment leveraging large language models," 2023. [Online]. Available: <https://arxiv.org/abs/2311.07469>