An approach to assess the content relevance of LLM-based chatbots in software testing education

Gabriel F. M. Rodrigues, Universidade de São Paulo, gabriel.rodrigues.aluno@gmail.com
https://orcid.org/0009-0004-3157-8697
Pedro Henrique Dias Valle, Universidade de São Paulo, pedrohenriquevalle@usp.br
https://orcid.org/0000-0002-6929-7557
Leo Natan Paschoal, Pontifícia Universidade Católica do Paraná, leo.paschoal@pucpr.br
https://orcid.org/0000-0003-1076-9174

Abstract: Large Language Model (LLM)-based chatbots have been increasingly adopted by students across educational levels as support tools for solving exercises, clarifying doubts, and deepening conceptual understanding. In the context of software testing education, recent studies have explored the potential of these systems as mechanisms to enhance learning. However, as LLMs are integrated into educational activities, ensuring the reliability and conceptual accuracy of their responses becomes essential. One major challenge is that poorly formulated or ambiguous questions can lead to conceptually incorrect answers. Moreover, LLMs are prone to hallucinations, producing linguistically coherent yet factually inaccurate responses, which poses a significant risk of misinformation. This paper proposes a systematic approach based on BERTScore to assess the domain relevance of responses generated by LLM-based chatbots in conversations about software testing. The approach semantically compares chatbot responses with a specialized conceptual reference, enabling the identification of domain drift and potential conceptual errors.

Keywords: BERTScore; Chatbot; Software engineering education; Software testing.

Uma abordagem para avaliar a relevância do conteúdo de chatbots baseados em LLM no ensino de teste de software

Resumo: Chatbots baseados em grandes modelos de linguagem (LLMs) têm sido amplamente utilizados como ferramentas de apoio à aprendizagem em diversas áreas do conhecimento. No contexto do ensino de teste, esses sistemas podem auxiliar estudantes na compreensão de conceitos técnicos e na resolução de dúvidas sobre processos, técnicas e critérios de teste. Entretanto, a confiabilidade das respostas geradas ainda representa um desafio, especialmente diante do risco de alucinações e da dificuldade em distinguir respostas corretas de conceitualmente imprecisas. Este artigo apresenta uma abordagem sistemática para avaliar a relevância do domínio das respostas produzidas por chatbots baseados em LLMs em diálogos de teste de software. A proposta utiliza a métrica BERTScore para medir a similaridade semântica entre as respostas e um referencial conceitual especializado da área, permitindo identificar desvios e inconsistências conceituais.

Palavras-chave: BERTScore; Chatbot; Educação em engenharia de software; Teste de software.

1. Introduction

Conversational agents, or chatbots, are software systems designed to interact with users through natural language, whether by text or speech (ALLOUCH; AZARIA; AZOULAY, 2021). Owing to their ability to emulate human dialogue, these systems have been adopted across a wide range of domains, with education emerging as one of the most promising areas of application. In this context, chatbots function as continuously available learning companions—capable of clarifying doubts, scaffolding reasoning, and adapting interactions to individual learners' needs (LO; HEW; JONG, 2024). This potential has been particularly explored in technical disciplines, such as software testing education (PASCHOAL *et al.*, 2023).

Teaching software testing—a foundational yet challenging component of software engineering education—has increasingly incorporated innovative pedagogical models such as flipped classrooms and Massive Open Online Courses (MOOCs), which require greater learner autonomy (PASCHOAL *et al.*, 2023). To support this autonomy, early initiatives introduced

domain-specific chatbots, such as TOB-STT, a pattern-matching agent designed to assist students in understanding testing concepts (MUSICO *et al.*, 2025). While such systems showed promise, they also exposed critical limitations, especially in their ability to interpret nuanced questions and varied linguistic formulations from students (MUSICO *et al.*, 2025).

These comprehension challenges were largely overcome with the advent of Large Language Models (LLMs), which have driven a new generation of chatbots, such as ChatGPT (MEHRVARZ *et al.*, 2025). LLMs are capable of producing coherent and contextually appropriate responses across diverse situations, offering substantial potential for education through personalized learning, immediate feedback, and cognitive support, all of which contribute to increased student engagement and autonomy (LO; HEW; JONG, 2024; GARG; SHARMA, 2023). However, this same generative capacity introduces new risks, notably the phenomenon of hallucinations, where models produce responses that are linguistically fluent but factually incorrect (HUANG *et al.*, 2025; ZHANG *et al.*, 2025).

This issue becomes particularly critical in software testing education, where terminological precision is non-negotiable. Even minor ambiguities in student questions can lead LLMs to conflate closely related but distinct concepts—such as fault, error, and failure—or to blur differences among terms like testing criteria, technique, and test; testing phases and testing levels; or test requirements, test data, and test cases. A recent empirical study found that ChatGPT produced correct or partially correct answers in only 55.6% of conceptual questions related to software testing, revealing persistent gaps in its grasp of formal definitions (JALIL *et al.*, 2023).

Such findings highlight an issue: the lack of systematic approaches for evaluating the semantic and conceptual relevance of chatbot responses in specialized domains. Off-topic or imprecise answers can mislead students, perpetuating misconceptions and undermining learning outcomes, especially in technical education, where conceptual rigor is crucial. The lack of mechanisms to verify whether an LLM's output remains aligned with the target knowledge domain poses a risk of misinformation.

This study proposes a structured approach to evaluate the conceptual relevance of LLM-generated responses in the context of software testing education. The approach measures semantic similarity between student questions, chatbot responses, and a specialized conceptual reference to detect when a model deviates from the expected content. Acting as an early-warning mechanism, it allows educators and researchers to assess the conceptual alignment of responses, identify potential domain drift, and promote the safe and pedagogically reliable use of LLM-based educational tools.

The key contributions of this work are as follows: (i) the proposal of a six-step approach to quantify the domain relevance of LLM-generated chatbot responses; and (ii) the presentation of empirical evidence showing that prompt contextualization—explicitly framing the model as a domain expert—enhances the conceptual relevance and consistency of its responses.

2. Chatbots in Software Engineering Education

The academic community's interest in using chatbots to support Software Engineering education predates the rise of large language models (LLMs). Early research focused on developing expert chatbots based on structured, domain-specific knowledge bases. In Requirements Engineering education, for instance, conversational agents were designed to simulate the role of a client during elicitation interviews, allowing students to practice communication and analytical skills in realistic scenarios (PASCHOAL; OLIVEIRA; CHICON, 2018). Other initiatives introduced virtual assistants for project management (MA *et al.*, 2023) and code review activities (FARAH *et al.*, 2022).

While these studies highlight the pedagogical value of chatbots as learning support tools, they also expose a persistent limitation: developing and maintaining comprehensive knowledge

bases is a complex and resource-intensive task (ABDELLATIF *et al.*, 2024; MA *et al.*, 2023).

The rigidity of these rule-based systems prompted researchers to explore the potential of LLMs, whose linguistic generalization capabilities eliminate the need for predefined domain ontologies. The emergence of tools such as ChatGPT opened new possibilities for educational applications, offering more flexible and adaptable chatbot interactions across diverse learning contexts (ANDERSEN-KIEL; LINOS, 2024). Empirical evidence suggests that students perceive value in using ChatGPT for debugging, code generation, and exam preparation—viewing it as a valuable aid for continuous learning. However, the same body of work consistently reports conceptual inaccuracies and inconsistent responses, raising concerns about the reliability of generative models in technical domains.

These concerns are particularly significant in software testing education, where terminology carries precise and interrelated meanings. Recent investigations have revealed that general-purpose LLMs perform inconsistently in this domain. Jalil *et al.* (2023) found that ChatGPT produced correct or partially correct answers in only 55.6% of questions, while Shloul *et al.* (2024) reported an even lower accuracy rate of 37.5%.

The literature on educational chatbots can therefore be divided into two major phases: (i) a pre-LLM phase, focused on the development of expert, knowledge-based systems; and (ii) a post-LLM phase, centered on performance evaluation, user perceptions, and the reliability of generative models in educational contexts. This latter phase highlights the need for complementary evaluation methods that can detect when a chatbot deviates from its intended knowledge domain.

In this context, the present study directly addresses that gap by proposing a systematic approach for measuring the conceptual relevance of responses generated by LLMs in software testing dialogues. Unlike studies that merely quantify correctness or error rates, this approach functions as an alert mechanism, capable of identifying domain drift and supporting the pedagogically reliable use of chatbots in educational settings.

3. Proposed Approach

The proposed approach aims to measure the conceptual relevance of responses generated by LLM-based chatbots by comparing them against a predefined domain reference. The method is structured into six interdependent stages, spanning from the definition of a semantic similarity model to the validation of the conceptual adequacy of the resulting vocabulary. Each stage is detailed below.

Stage 1 — Defining the semantic similarity metric

Given the reliability challenges associated with LLMs, designing an evaluation framework required a methodological process capable of systematically quantifying the conceptual relevance of chatbot responses. The central challenge lay in translating an abstract notion—"relevance"—into something objective and measurable.

While human evaluation provides precision, it lacks scalability. Conversely, keyword-based approaches are too shallow: a response might include correct terms yet misrepresent conceptual relationships, or employ valid synonyms that simple lexical matching fails to capture.

To overcome these limitations, this study employed a semantic similarity metric to quantify the degree to which two texts align in meaning. This metric embeds texts into a semantic vector space, where distances between vectors represent degrees of semantic proximity.

We employed BERTScore, a widely recognized metric in natural language evaluation that measures token-level similarity using embeddings from pre-trained BERT models (HU; ZHOU, 2024). BERTScore requires two input texts:

- A reference text, representing the expected "within-domain" knowledge; and

- A candidate text, corresponding to either the student's question or the chatbot's response.

The metric outputs a numerical score reflecting the degree of semantic similarity between the two. This study used the paraphrase-MiniLM-L6-v2 model[1], optimized for semantic search tasks and known for its balance between accuracy and computational efficiency.

Once the metric was established, the next stage focused on constructing a robust representation of the software testing domain, which served as the conceptual benchmark for all subsequent analyses.

Stage 2 — Building the software testing domain representation

After defining the similarity metric, the next step involved building the domain reference. This representation consists of a set of terms that capture the technical vocabulary of software testing, forming a foundation to assess whether a text remains within its conceptual boundaries.

To ensure reliability and currency, the knowledge base was derived from the IEEE International Conference on Software Testing, Verification and Validation (ICST)[2], the field's leading venue. Three factors guided the choice of ICST: (i) its international recognition as the primary scientific forum in software testing, organized by IEEE and supported by world-class technical committees; (ii) its rigorous peer-review process, ensuring terminological precision and conceptual accuracy—both critical for building a dependable semantic foundation; and (iii) its broad coverage of subdomains, which enhances vocabulary representativeness and minimizes thematic bias.

The 2023 edition was selected because it represented the most up-to-date state-of-the-art information available during the development of this approach. Since the 2024 and 2025 proceedings had not yet been released, using ICST 2023 ensured temporal consistency between the corpus and the methodological execution.

All 54 papers from the conference proceedings were included, avoiding selective sampling and ensuring a comprehensive, representative dataset of technical vocabulary.

From this corpus, 4,546 unique words were extracted. Because raw text from PDF files often contains linguistic noise, a lexical refinement step was performed using the PyEnchant library[3] in Python. Each extracted term was validated against an English dictionary, and invalid entries were discarded, resulting in 3,404 valid words. This refinement improved linguistic integrity and minimized distortions in similarity computation.

These 3,404 words were then ranked by semantic proximity to the expression "software testing." The 100 most semantically similar terms were selected to form the domain's core vocabulary—a methodological compromise between breadth and conceptual focus. This size is broad enough to capture key concepts yet concise enough to prevent semantic dispersion and maintain interpretability. It also facilitates visual exploration (e.g., word clouds) and expert validation.

Figure 1 illustrates the most semantically relevant terms representing the software testing domain.

Finally, an experienced software testing researcher, independent of this study's design, reviewed and validated the extracted terms, ensuring the conceptual soundness of the reference vocabulary.

Stage 3 — Collecting student–chatbot interactions

This stage focused on compiling real student–chatbot interactions in the context of software testing education. The goal was to assemble a representative dataset of authentic student

[1]More information available at: ⟨https://bit.ly/3SE1iS0⟩

[2]More information available at: ⟨https://conf.researchr.org/home/icst-2025⟩.

[3]More information available at: ⟨https://pyenchant.github.io/pyenchant/index.html⟩.

Figure 1 - Word cloud showing the 100 most semantically similar terms representing the software testing domain.

questions that capture typical misunderstandings, conceptual gaps, and linguistic patterns in real learning environments. These questions were later used as prompts for the LLM to generate responses for conceptual relevance analysis.

The decision to use authentic student questions, rather than textbook-derived ones, was based on a key rationale: spontaneously formulated questions reveal how students actually comprehend, confuse, or articulate concepts in authentic learning situations. Using such data allows the evaluation of the LLM's ability to manage linguistic variability and conceptual imprecision characteristic of genuine educational dialogues—unlike controlled, expert-crafted prompts.

The dataset was sourced from publicly available collections published by Paschoal *et al.* (2023), which document student interactions with TOB-STT, a rule-based chatbot designed to support software testing education. The datasets, available on GitHub[4], include CSV files containing both student questions and TOB-STT responses. A copy of the repository was used to extract 820 unique student-generated questions, representing a diverse range of inquiries about software testing. These questions were employed as prompts for generating new responses via the LLM. This approach eliminated the need for additional data collection, reducing cost and variability while ensuring reproducibility and alignment with prior research.

Stage 4 — Generating LLM responses

Using the 820 collected prompts, responses were generated with Gemini Pro[5], accessed via Google GenAI API. The model was selected for its operational stability and transparent documentation.

Each question was automatically submitted to the model, and the generated responses were stored for further analysis.

To examine the impact of prompt contextualization on model performance, each question was tested under two configurations:

- Mode 1: the original student question submitted directly;
- Mode 2: the same question prefaced with explicit context: "Suppose you are a software testing specialist. Knowing this, answer the following question."

This second mode implements a role-based prompting strategy (QIAN, 2025), prompting the model to adopt the role of a domain expert before generating its response. The goal is to enhance conceptual alignment and minimize generic or off-topic outputs.

Stage 5 — Semantic similarity evaluation

In this stage, LLM-generated responses were compared with the domain vocabulary defined in Stage 2 to quantify domain relevance and detect conceptual drift—instances where linguistically coherent responses diverged from the expected technical scope.

[4]More information available at: ⟨https://github.com/?locale=pt-br⟩.
[5]More information available at: ⟨https://gemini.google.com/⟩.

Responses were first standardized by removing line breaks and special characters to ensure textual consistency. BERTScore was then used to compute the mean similarity between each response and the domain vocabulary, producing a continuous value between 0 and 1. Scores near 1 indicate high conceptual alignment, while lower values reflect weaker domain relevance or off-topic content.

Scores were grouped according to prompt mode (with or without context) to enable direct comparison and to assess the effect of contextualization on conceptual coherence.

Stage 6 — Analysis and interpretation of results

The final stage involved systematically analyzing similarity scores and defining rules to infer the likelihood that an interaction remained within the software testing domain.

All BERTScore results for each question–response pair were consolidated into a single dataset to ensure traceability. Outliers were removed using the interquartile range method to prevent distortion of the overall score distribution.

Because continuous similarity values are difficult to interpret directly, scores were converted into discrete, interpretable categories using standard statistical segmentation based on the mean and standard deviation of the distribution.

Each question and response was categorized as Dissimilar, Low Similarity, Medium Similarity, or High Similarity, representing its relative position within the dataset. These categories provide an intuitive conceptual interpretation—e.g., a "High Similarity" score indicates strong domain alignment. Table 1 presents the classification intervals used.

Table 1 - Classification intervals for BERTScore values.

Category	Score range
Dissimilarity	$score < mean - std$
Low similarity	$mean - std \leq score < mean$
Medium similarity	$mean \leq score < mean + std$
High similarity	$score \geq mean + std$

Based on these categories, a set of inference rules was developed to estimate whether a student–chatbot exchange remained within the intended domain. Rules were defined by combining the similarity classes of each question–response pair. Interactions in which both sides exhibited high similarity were classified as within domain, whereas those with low similarity on either side were flagged as potential conceptual deviations.

Table 2 presents all possible combinations between question and response similarity classes.

Table 2 - Inference rules combining question and response similarity classes.

Question similarity	Response similarity	Conceptual inference
High	High	Within domain
Medium	High	Partially aligned
High	Medium	Partially aligned
Low	High	Potential deviation
High	Low	Potential deviation
Medium	Medium	Potential deviation
Low	Medium	Out of domain
Low	Low	Out of domain

These rules do not constitute a definitive qualitative assessment but rather serve as an

early-warning mechanism to flag responses that may have semantically drifted from the reference domain.

4. Results and Discussion

This section presents the main results obtained from applying the proposed approach. The findings discussed here reflect the practical execution of the steps described in Section 3—that is, the application of the methodology to a dataset composed of real student questions and responses generated by the Gemini Pro model.

4.1 General analysis of student questions and LLM responses

Applying the BERTScore metric to the different textual datasets—student-generated questions on software testing and responses produced by the Gemini 1.5 Pro model—enabled the quantification of each text's semantic proximity to the specialized conceptual reference. This analysis made it possible to compare how closely the LLM's responses remained aligned with the software testing domain relative to the students' original questions. The responses were evaluated in two distinct modes: (1) without additional contextualization and (2) with explicit contextualization, where the model was instructed to adopt the role of a domain expert.

Figure 2 presents the distribution of similarity scores for the three analyzed categories: (a) student questions; (b) responses generated without contextualization; and (c) responses generated with explicit contextualization.

The distribution shows that student questions are predominantly concentrated in the low and medium similarity ranges. This result indicates that, although the questions relate to the software testing domain, the vocabulary used by students does not always coincide with the field's core conceptual terminology. This pattern reflects the naturally variable language of learners, who often employ less technical expressions when describing domain concepts.

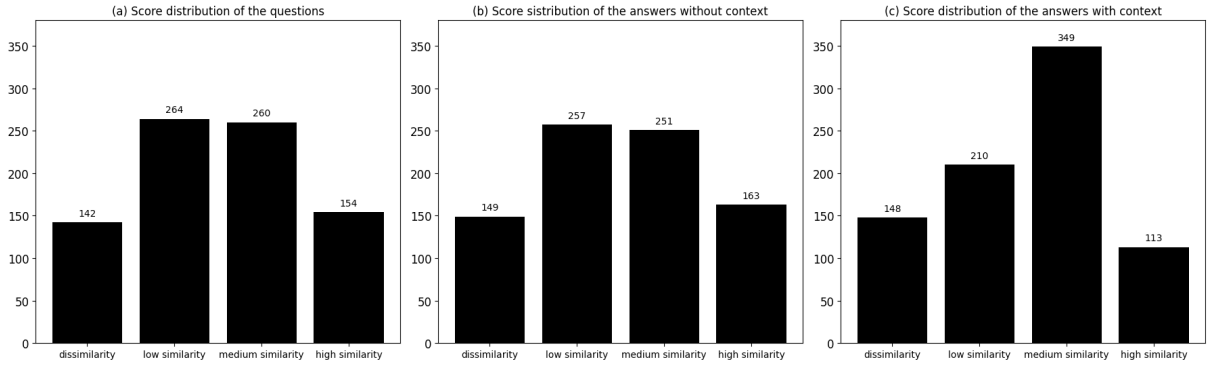

Figure 2 - Distribution of semantic similarity scores for student questions and LLM responses.

The similarity distributions for student questions and responses generated without context (Figures 2a and 2b) exhibit comparable patterns. In both cases, low and medium similarity levels prevail, suggesting that both students' questions and non-contextualized responses share only partial alignment with the software testing vocabulary.

In contrast, Figure 2(c) reveals a perceptible shift in the distribution of responses generated with contextualization. There is a simultaneous reduction in low- and high-similarity categories and an increase in medium-similarity instances. This shift suggests two distinct effects. First, some responses shifted from lower to higher similarity ranges, indicating an improvement in quality due to contextualization. Second, specific high-similarity responses slightly dropped to the medium range—likely reflecting BERTScore's natural sensitivity to minor semantic variations in generated text.

4.2 Effect of prompt contextualization

Comparing the two prompting modes, without and with explicit contextualization, revealed significant differences in both the distribution and semantic quality of generated responses. As shown in Figures 2(b) and 2(c), responses without context display a relatively balanced spread between low and medium similarity, while contextualized responses cluster more strongly around medium similarity, with proportional reductions in the extremes. This redistribution suggests that contextual prompts guided the LLM-based chatbot to produce responses that were both more stable and more semantically centered.

Table 3 quantitatively summarizes these differences, presenting the means and standard deviations of each set. The average BERTScore for contextualized responses rose from 0.30 to 0.37 compared to the non-contextualized mode, accompanied by a drop in standard deviation (from 0.15 to 0.11). This pattern indicates that the model not only produced responses more closely aligned with the software testing domain but also achieved greater semantic consistency, exhibiting less variability across similar instances.

Table 3 - Mean and standard deviation of BERTScores for responses.

Score types	Mean	Standard Deviation
Score of answers without context	0.30	0.15
Score of answers with context	0.37	0.11

Consistency is a key factor in assessing the reliability of LLM-based chatbots in educational contexts, as it reduces the likelihood of the model alternating between accurate and off-topic responses to similar questions. In this sense, the prompt—by instructing the model to "assume the role of a software testing expert"—acted as a semantic constraint, narrowing the model's generative space to a conceptually more stable region. This prompt-driven constraint contributed to aligning the chatbot's vocabulary with the characteristic terminology of the software testing domain, thereby mitigating issues such as hallucinations and overly generic responses.

Although the average similarity gain may seem modest, the result holds educational significance: instructing the chatbot to adopt a specialized role enhances the conceptual coherence of its answers. It demonstrates a viable strategy for adapting general-purpose LLMs to technical domains without requiring structural reconfiguration.

4.3 Conceptual inference

While similarity scores offer a quantitative foundation for analysis, they must be translated into discrete, interpretable categories to function as a practical evaluation tool (see Table 1).

However, classifying a question or response in isolation is insufficient for conclusively assessing domain adherence. For example, a "High Similarity" response is not necessarily good—if it answers a "Low Similarity" question, the chatbot may be forcing a topic or hallucinating a non-existent connection. Likewise, a "Low Similarity" response is not inherently poor—if paired with a "Low Similarity" question, it simply indicates that the chatbot avoided responding outside its scope.

Meaningful insights thus arise not from isolated classifications but from the relationship between question and response categories. This relationship reveals behavioral patterns of the LLM-based chatbot. Accordingly, the final stage of the approach defined a set of inference rules to assess the relevance of each interaction. These rules use the question's category as a predictor of expected behavior and evaluate whether the response aligns with that expectation—transforming the analysis into a practical alert mechanism. The two general rules were:

Rule 1: Questions with low domain relevance tend to generate equally irrelevant responses.

The analysis showed that questions categorized as "Dissimilar" or "Low Similarity," when submitted without explicit context, tend to elicit responses in the same low-relevance range. This pattern suggests that such questions are inherently unsuitable for obtaining reliable information—and that the chatbot's reaction is consistent and expected.

Figure 3(a) illustrates this case. A vague question such as "How to do the test?" has low semantic similarity with the software testing domain. Unable to identify specific technical terms, the chatbot provides a generic response applicable to any type of test. Thus, both the question and response fall into the low-similarity category. The "Low Question + Low Response" combination correctly leads the approach to infer that the interaction lies outside the domain.

Rule 2: Questions with high domain relevance tend to produce aligned responses.

Conversely, questions classified as "Medium" or "High Similarity" are generally well understood by the chatbot, which tends to remain within the expected domain of knowledge. Significant drops in similarity—where the response falls sharply—are rare and signal possible misinterpretation by the model.

This ideal case is illustrated in Figure 3(b), where a technically precise question ("What is the difference between black-box testing and white-box testing?") receives a high-similarity response, indicating that the interaction remained within the domain.

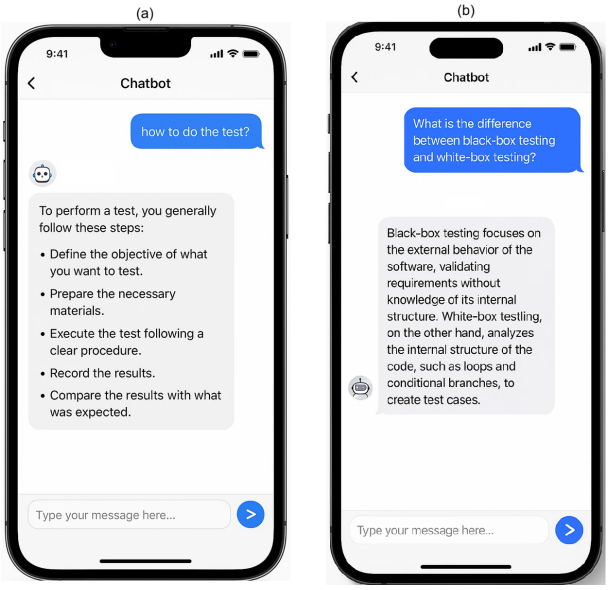

Figure 3 - Examples of chatbot interactions illustrating domain retention inference.

Applying these rules collectively turns the approach into a diagnostic mechanism. By classifying both the question and the response and analyzing their combination, it becomes possible to infer whether an interaction remained within the domain, is partially aligned, or represents a potential deviation. This inference does not replace qualitative analysis, but it provides an automated, scalable method to flag responses that merit closer review, helping mitigate misinformation risks in the educational use of LLMs.

5. Limitations

Although the results are promising, this study presents some limitations.

First, the approach was evaluated solely in the context of software testing education, using English-language interactions and a single LLM (Gemini Pro). Replications in other domains, models, and languages are needed to confirm its generalizability.

Second, the domain representation was derived from ICST 2023 papers, forming a simplified textual vocabulary rather than a formal ontology. Despite previous attempts to identify a publicly available ontology for software testing (e.g., Barbosa, Nakagawa and Maldonado (2006)), no complete and accessible textual version was found that could be directly used in this research. Consequently, the chosen corpus served as a practical approximation of domain concepts, later validated by an expert.

Third, BERTScore captures semantic proximity but not pedagogical adequacy; thus, a high similarity score does not necessarily imply instructional relevance. Finally, the dataset originated from interactions with a rule-based chatbot (TOB-STT), which may not fully reflect authentic LLM-based dialogues. Future classroom experiments could validate the approach in direct human–LLM interactions.

6. Conclusions

This paper presented an approach for analyzing the conceptual relevance of responses generated by Large Language Model (LLM)-based systems within the context of software testing education. The primary goal was to explore the use of semantic inference mechanisms to assess the conceptual alignment of responses produced by educational chatbots. To this end, a six-stage methodology was proposed, encompassing the construction of a domain-specific vocabulary, the application of the BERTScore metric, and the formulation of inference rules correlating the similarity categories of questions and answers. This methodological structure establishes a reproducible and adaptable framework that can be applied to other knowledge domains.

The findings indicate that prompt contextualization, specifically assigning the model a specialist role explicitly, enhances conceptual coherence and reduces variability in responses. The analysis demonstrated that the proposed approach can serve as an effective tool for evaluating the reliability of LLM-based chatbots, thereby helping to mitigate misinformation risks in educational environments. Although the study presents certain limitations, such as relying on a single model and lacking validation with data directly obtained from students, the results suggest that the methodology holds strong potential for expansion and can be applied across various LLM-assisted learning scenarios.

The identified limitations also open avenues for future research. Among the possible extensions is the development of plugins or browser-based tools capable of applying the proposed inference rules in real time to monitor interactions between students and chatbots. Such mechanisms could function as early-warning systems, flagging potentially inaccurate responses and prompting students and instructors to engage in critical verification. However, implementing solutions of this nature requires careful adherence to ethical and legal principles, particularly in terms of protecting privacy and using educational data responsibly.

Acknowledgments

The authors gratefully acknowledge the support of the Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico do Estado do Paraná.

References

ABDELLATIF, A. *et al.* A transformer-based approach for augmenting software engineering chatbots datasets. In: *International Symposium on Empirical Software Engineering and Measurement*, 2024. p. 359–370.

ALLOUCH, M.; AZARIA, A.; AZOULAY, R. Conversational agents: Goals, technologies, vision and challenges. *Sensors*, MDPI, v. 21, n. 24, p. 8448, 2021.

ANDERSEN-KIEL, N.; LINOS, P. P. Using chatgpt in undergraduate computer science and software engineering courses: A students' perspective. In: *IEEE Frontiers in Education Conference*: IEEE, 2024. p. 1–9.

BARBOSA, E. F.; NAKAGAWA, E. Y.; MALDONADO, J. C. Towards the establishment of an ontology of software testing. In: *International Conference on Software Engineering and Knowledge Engineering*San Francisco, CA: , 2006. p. 522–525.

FARAH, J. C. *et al.* Impersonating chatbots in a code review exercise to teach software engineering best practices. In: *IEEE Global Engineering Education Conference*. Tunis, Tunisia: IEEE, 2022. p. 1634–1642.

GARG, A.; SHARMA, D. Generative ai for software test modeling with a focus on erp software. In: *International Conference on Advances in Computation, Communication and Information Technology*Faridabad, India: , 2023. p. 187–193.

HU, T.; ZHOU, X. Unveiling llm evaluation focused on metrics: Challenges and solutions. *arXiv preprint arXiv:2404.09135*, 2024.

HUANG, L. *et al.* A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, v. 43, n. 2, p. 1–55, 2025.

JALIL, S. *et al.* Chatgpt and software testing education: Promises & perils. In: *IEEE International Conference on Software Testing, Verification and Validation Workshops*, 2023. p. 4130–4137.

LO, C. K.; HEW, K. F.; JONG, M. S.-y. The influence of chatgpt on student engagement: A systematic review and future research agenda. *Computers & Education*, v. 219, p. 1–21, October 2024.

MA, S.-P. *et al.* Tabot: A teaching assistant chatbot for software engineering courses. In: *Asia-Pacific Software Engineering Conference*. Seoul, Korea, Republic of: IEEE, 2023. p. 627–631.

MEHRVARZ, M. *et al.* How does students' perception of chatgpt shape online learning engagement and performance? *Computers and Education: Artificial Intelligence*, v. 9, p. 1–15, December 2025.

MUSICO, T. *et al.* Um chatbot treinado para entender intenções de estudantes em diálogos sobre teste de software. In: SBC. *Simpósio Brasileiro de Informática na Educação*. Curitiba, PR, 2025. p. 873–886.

PASCHOAL, L. N. *et al.* An experimental study on a conversational agent in software testing lessons. *Informatics in Education*, v. 22, n. 1, p. 99–120, 2023.

PASCHOAL, L. N.; OLIVEIRA, M. M. de; CHICON, P. M. M. A chatterbot sensitive to student's context to help on software engineering education. In: *Latin American Computer Conference*. Sao Paulo, Brazil: IEEE, 2018. p. 839–848.

QIAN, Y. Prompt engineering in education: A systematic review of approaches and educational applications. *Journal of Educational Computing Research*, SAGE Publications Sage CA: Los Angeles, CA, p. 07356331251365189, 2025.

SHLOUL, T. A. *et al.* Role of activity-based learning and chatgpt on students' performance in education. *Computers and Education: Artificial Intelligence*, Elsevier, v. 6, p. 100219, 2024.

ZHANG, Z. *et al.* Llm hallucinations in practical code generation: Phenomena, mechanism, and mitigation. *Proceedings of the ACM on Software Engineering*, v. 2, n. ISSTA, p. 481–503, July 2025.