

A Multilingual Dataset of Student Answers, Human Grading, and Multi-LLM Evaluations for Automated Assessment Research Using JorGPT

Jorge Cisneros-González , Natalia Gordo-Herrera , Iván Barcia-Santos , Yolanda Cerezo and Javier Sánchez-Soriano * 

Advanced Artificial Intelligence Group (A²IG), Escuela Politécnica Superior, Universidad Francisco de Vitoria, Pozuelo de Alarcón, 28223 Madrid, Spain; j.cisneros.prof@ufv.es (J.C.-G.); natalia.gordo@ufv.es (N.G.-H.); i.barcia@ufv.es (I.B.-S.); y.cerezo@ufv.es (Y.C.)

* Correspondence: javier.sanchez@ufv.es

Abstract

The increasing adoption of Large Language Models (LLMs) in higher education has created a need for high-quality, publicly available benchmarks for automated assessment. Existing datasets often rely on synthetic responses or lack detailed human feedback. This paper presents a multilingual dataset of 3041 authentic student answers to 50 open-ended Computer Science questions, collected from real university assessments during the 2025–2026 academic year. The dataset includes the original student responses (Spanish) and their parallel translations (English), instructor (or teacher) defined ideal answers, blind human grading with qualitative feedback, and structured evaluations from three state-of-the-art LLMs (DeepSeek-chat-V3.2, Qwen-flash-2025-07-28, Gemini-2.5-flash-lite-001) using a unified JSON schema. This resource enables reproducible research in automated grading, feedback generation, and cross-lingual educational NLP.

Dataset: <https://doi.org/10.5281/zenodo.18369061> (accessed on 12 March 2026)

Dataset License: Creative Commons Attribution 4.0 International

Keywords: automated assessment; educational data mining; large language models; open-ended questions; automated grading; multilingual datasets; higher education



Academic Editor: Antonio Sarasa Cabezuelo

Received: 30 January 2026

Revised: 14 March 2026

Accepted: 16 March 2026

Published: 17 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The use of artificial intelligence techniques for automated assessment has gained increasing attention in higher education, particularly for the grading of open-ended questions and the generation of formative feedback. Large language models (LLMs) have demonstrated strong capabilities in natural language understanding and generation, making them attractive candidates for supporting or partially automating assessment processes. LLMs have recently been explored as tools for automated assessment across multiple educational domains, showing promising results while also highlighting challenges related to consistency, transparency, and alignment with human grading [1]. Their adoption promises to reduce instructors' workload, improve feedback timeliness, and enable scalable evaluation in large or resource-constrained courses.

Despite these advances, research on automated assessment remains limited by the availability of high-quality, publicly accessible datasets that reflect authentic educational

settings. Many existing resources rely on synthetic or heavily curated student responses, provide only numerical grades without accompanying feedback, or omit information about evaluation criteria and grading rationale. Moreover, the rapid evolution of LLM-based graders has outpaced the release of datasets that include structured evaluations generated by multiple contemporary models under comparable conditions.

In this context, realistic datasets that combine human and automated assessments are essential for reproducible research, fair benchmarking, and the systematic analysis of alignment, bias, and reliability in automated grading systems. Such datasets are particularly valuable when they include multilingual content, as higher education is inherently multilingual and cross-lingual robustness is a key requirement for educational AI systems. Multilingual automated open-ended grading remains a challenging problem, as grading performance and evaluation criteria may vary significantly across languages and cultural [2].

Despite advances in educational NLP, datasets combining authentic student responses, human feedback, and machine-generated evaluations remain [2]. This work introduces a multilingual dataset of open-ended student answers collected from real higher education assessments. The dataset is released as an open resource with detailed documentation and is intended to support research in automated assessment, educational data mining, and natural language processing for education. This dataset builds upon prior work on LLM-assisted grading frameworks, such as the JorGPT system, which demonstrated the practical integration of multiple large language models to support instructors during the correction of open-ended assessments [3].

2. Related Work

Research on automated assessment intersects with several key domains, including Educational Data Mining (EDM), Natural Language Processing (NLP) for Education, and Automated Short Answer Grading (ASAG). Historically, initiatives such as the ASAP dataset [4] and SemEval shared tasks [5] have driven progress in these fields by providing public benchmarks for feature-based models and transfer learning techniques.

In the specific context of Automated Short Answer Grading (ASAG), the availability of high-quality multilingual data remains a critical challenge. While several datasets like ASAP and ScAA have been proposed, they typically focus on monolingual settings and provide limited qualitative feedback [6]. As noted in recent cross-lingual studies, multilingual grading warrants realistic datasets for systematic study to overcome language-specific barriers [2]. In addition, existing benchmarking resources often differ widely in scope and annotation depth [7], frequently exhibiting limitations such as the use of artificially constructed answers, the absence of qualitative instructor feedback, or the reliance on a single aggregated score per response.

More recently, within Educational NLP and AI-assisted assessment, the integration of Large Language Models (LLMs) into educational evaluation has shifted the research focus. Studies indicate that LLM-based graders can produce scores comparable to human evaluators under controlled conditions, particularly for short and well-defined answers [8]. Comparative analyses of LLM-based grading systems report strong correlations with human scores, although variability persists depending on question type and rubric definition [9]. However, research in this domain frequently relies on private or proprietary datasets that are not released publicly, limiting reproducibility. Additionally, most existing datasets do not include structured evaluations from multiple LLMs using a unified grading schema, making systematic comparison across models difficult.

To address these limitations, recent approaches have proposed structured evaluation schemes and rubric-based scoring to improve the interpretability and robustness of auto-

mated assessment systems [10]. The dataset presented in this paper bridges these gaps by combining authentic student answers, instructor-defined ideal responses, detailed human grading with feedback, and structured evaluations generated by several state-of-the-art LLMs. Unlike prior system-focused work such as JorGPT [3], the present contribution centers on the public release and documentation of the underlying data to support reproducible research in EDM, NLP, and automated assessment.

To better contextualize the contribution of the proposed resource, Table 1 provides a comparison with prominent datasets in the literature. While large-scale datasets such as ASAP-SAS [4] offer a higher volume of responses (approx. 17,000), they often lack qualitative feedback and are restricted to monolingual settings. The classic Mohler dataset [11], frequently used in Computer Science education, is comparable in size (approx. 2273 responses) but offers only numerical scores.

Table 1. Comparison with existing open-ended grading datasets.

Dataset	Domain	Language	Size (Responses)	Evaluation Metadata
Mohler et al. (2011) [11]	Computer Science	English	2273	Human score (0–5 scale)
ASAP (AES) (2012) [4]	Science/ELA	English	12,978	Human score
ScAA (2020) [12]	Science	Hindi/Marathi	12,943	Human label (2-way)
Ours (2026)	Computer Science	Spanish/English	3041	Human Score + Feedback + Multi-LLM Structured Evals (JSON)

In contrast, our dataset (comprising 3041 responses) aligns with the scale of domain-specific datasets while significantly expanding the depth of annotation. Uniquely, it includes instructor-generated qualitative feedback, detailed evaluations from three modern LLMs following a unified JSON schema, and a full parallel translation (Spanish–English), features that are absent in established benchmarks.

3. Dataset Summary

This dataset was created to support research on automated assessment of open-ended questions in higher education using both human and LLM-based evaluators. It consists of student responses collected during regular higher education courses in computer science and software engineering, where students answered theoretical questions in free-text form as part of their formal assessment.

The dataset includes original student answers, instructor-defined ideal answers, numerical grades and qualitative feedback provided by human teachers, and structured evaluations produced by multiple LLMs following a common grading schema. All content was originally produced in Spanish, the language of instruction, and is provided in the file “dataset_es.csv”. A second file, “dataset_en.csv”, contains an English translation of all pedagogical content, aligned with the original dataset at the record level, resulting in a fully bilingual dataset. The data were generated within the operational context of the JorGPT project, an automated assessment framework previously described in the literature, which provided the real classroom environment used for data collection [3].

The dataset was generated within the context of the JorGPT project, an automated assessment framework designed to assist instructors during the grading process. While the framework itself is not the focus of this paper, it provided the operational setting in which the data were collected and curated. To the best of our knowledge, no prior publications based on this dataset have been released.

Table 2 provides a quantitative overview of the dataset, summarizing its main characteristics. These statistics are intended to facilitate rapid understanding of dataset scale and support reproducibility and benchmarking in future studies.

Table 2. Summary statistics of the dataset.

Statistic	Value
Total number of student answers	3041
Number of distinct questions	50
Number of anonymized students	79
Courses/academic years	2025–2026
Average answer length (characters)	542.43
Grading scale	0–10
Number of LLM evaluators	3: DeepSeek (deepseek-chat-V3.2), Qwen (qwen-flash-2025-07-28), Gemini (gemini-2.5-flash-lite-001)
Number of Judges (meta evaluator)	1: OpenAI (gpt-4o-mini-2024-07-18)
Languages	Spanish (original), English (translation)

These statistics reflect real classroom assessment scenarios rather than synthetic or artificially balanced datasets, reinforcing the ecological validity of the resource. Both CSV files share the same structure, column names and number of records (3041 entries), differing only in the language of textual fields and in the language of JSON evaluation keys and fields.

4. Data Description

Each row in the dataset corresponds to a single student answering a specific question. The dataset is distributed in two CSV files: “dataset_es.csv”, containing the original Spanish data, and “dataset_en.csv”, containing the English translations. Both files share the same schema and are aligned at the row level. Tables 3 and 4 summarize all variables included in each dataset file, their descriptions, data types, and language coverage.

Table 3. Description of dataset variables (Spanish version—“dataset_es.csv”).

Field Name	Description	Data Type	Language
entry_id	Unique identifier for each dataset entry.	Integer	-
question_id	Unique identifier of the associated question.	String	-
student_anom_id	Irreversible anonymous identifier for each student.	String	-
question_text	Original question text as presented to students.	Text	Spanish
student_answer	Student’s original free-text answer.	Text	Spanish
ideal_answer	Instructor-defined reference answer (original language).	Text	Spanish
student_answer_length	Length of the student’s answer in characters.	Integer	-
deepseek_raw_json	Structured evaluation generated by the DeepSeek LLM.	JSON	Spanish
deepseek_grade	Numerical grade assigned by DeepSeek (0–10).	Float	-
qwen_raw_json	Structured evaluation generated by the Qwen LLM.	JSON	Spanish
qwen_grade	Numerical grade assigned by Qwen (0–10).	Float	-
gemini_raw_json	Structured evaluation generated by the Gemini LLM.	JSON	Spanish
gemini_grade	Numerical grade assigned by Gemini (0–10).	Float	-

Table 3. *Cont.*

Field Name	Description	Data Type	Language
judge_status	Indicator of internal validation status.	Categorical	-
judge_raw_json	Consolidated structured justification of the final grade.	JSON	Spanish
judge_grade	Final consolidated grade (0–10).	Float	-
teacher_grade	Grade assigned by the instructor.	Float	-
teacher_feedback	Qualitative feedback written by the instructor.	Text	Spanish
teacher_corrected_at	Timestamp of manual correction.	Datetime	-
teacher_corrected	Flag indicating whether manual correction was performed.	Boolean	-

Table 4. Description of dataset variables (English version—“dataset_en.csv”).

Field Name	Description	Data Type	Language
entry_id	Unique identifier for each dataset entry.	Integer	-
question_id	Unique identifier of the associated question.	String	-
student_anom_id	Irreversible anonymous identifier for each student.	String	-
question_text	English translation of the question text.	Text	English
student_answer	English translation of the student’s answer.	Text	English
ideal_answer	English translation of the reference answer.	Text	English
student_answer_length	Length of the student’s answer in characters.	Integer	-
deepseek_raw_json	Structured evaluation generated by the DeepSeek LLM.	JSON	English
deepseek_grade	Numerical grade assigned by DeepSeek (0–10).	Float	-
qwen_raw_json	Structured evaluation generated by the Qwen LLM.	JSON	English
qwen_grade	Numerical grade assigned by Qwen (0–10).	Float	-
gemini_raw_json	Structured evaluation generated by the Gemini LLM.	JSON	English
gemini_grade	Numerical grade assigned by Gemini (0–10).	Float	-
judge_status	Indicator of internal validation status.	Categorical	-
judge_raw_json	Consolidated structured justification of the final grade.	JSON	English
judge_grade	Final consolidated grade (0–10).	Float	-
teacher_grade	Grade assigned by the instructor.	Float	-
teacher_feedback	Qualitative feedback written by the instructor.	Text	English
teacher_corrected_at	Timestamp of manual correction.	Datetime	-
teacher_corrected	Flag indicating whether manual correction was performed.	Boolean	-

The file “dataset_en.csv” mirrors the structure of “dataset_es.csv”, replacing Spanish textual fields with their English translations. The structured JSON fields produced by LLMs follow a common schema that includes multiple evaluation dimensions, such as coverage accuracy, quality of justification and reasoning, terminology clarity, and optional learning transfer indicators. This unified structure facilitates systematic comparison across models and supports alignment analysis with human grading.

Three additional columns contain the raw evaluation output produced by different large language models (DeepSeek, Qwen, and Gemini). These columns store JSON-formatted responses that include detailed scoring dimensions such as coverage accuracy, reasoning quality, terminology clarity, learning transfer assessment, and a final aggregated score. The JSON structure is identical across models to facilitate automated parsing and comparative analysis.

For convenience, the dataset also includes the final numerical score extracted from each model's evaluation, allowing researchers to perform quantitative analyses without parsing the JSON fields:

- “coverage_accuracy”: Degree to which the answer covers the key concepts.
- “justification_reasoning”: Quality of reasoning and explanation.
- “terminology_clarity”: Correctness and clarity of technical terminology.
- “learning_transfer”: Whether the student demonstrates conceptual transfer.
- “total_score”: Final aggregated score.

Table 5 contains a representative example of a single record in the “dataset_es.csv”, whereas Table 6 contains the same example stored in “dataset_en.csv”, showing the structure and richness of information included. The example illustrates the combination of student responses in both languages, multiple LLM evaluations, and instructor grading with feedback, highlighting the dataset's multimodal evaluation capability.

Table 5. Representative record example—“dataset_es.csv”.

Field	Example
entry_id	000123
question_id	Q799823558
student_anom_id	495b02083ab12972a416b93c9c6e776abfdadb1a87fd450a82cf2b6f97d89074d
question_text	Compara TDD con un enfoque tradicional de testing en el desarrollo de software. El enfoque tradicional de testing era que primero se desarrollaba el código del proyecto y luego se hacían test para probar que el código funcionase correctamente. Sin embargo, el enfoque TDD, es al contrario, primero se hacen los test y luego se desarrolla el código del proyecto que va a hacer que esos test los pasen.
student_answer	El enfoque de desarrollo del TDD hace que el código sea mas limpio, fiable y modular. Ya que esta estructura junto con los frameworks de testing pueden confirmar que se ha desarrollado exclusivamente el código justo y necesario para cada funcionalidad. TDD y el enfoque tradicional de testing difieren fundamentalmente en su secuencia y propósito. En TDD, los tests se escriben antes del código, sirviendo como especificación y guía para el desarrollo. En contraste, en un enfoque tradicional, primero se desarrolla el código y luego se realizan pruebas, lo que puede llevar a una verificación reactiva. Además, en TDD, los tests son obligatorios y ayudan a prevenir errores desde el inicio, mientras que en el enfoque tradicional, los tests pueden ser opcionales y se utilizan para validar el trabajo ya realizado.
ideal_answer	
student_answer_length	577
deepseek_raw_json	{“coverage_accuracy”: 7, “justification_reasoning”: 6, “terminology_clarity”: 5, “learning_transfer”: {“applies”: false, “bonus”: 0}, “total_score”: 6.5, “coments”: “Cubres la secuencia básica. Falta precisar propósito y diferencias conceptuales. Mejora terminología y ortografía. Explica por qué TDD produce código más limpio.”}
deepseek_grade	6.5
qwen_raw_json	{“coverage_accuracy”: 8, “justification_reasoning”: 7, “terminology_clarity”: 6, “learning_transfer”: {“applies”: false, “bonus”: 0}, “total_score”: 7.7, “coments”: “Cubres bien la secuencia y ventajas de TDD. Mejora precisión en términos técnicos y redacción.”}
qwen_grade	7.7
gemini_raw_json	{“coverage_accuracy”: 7, “justification_reasoning”: 7, “terminology_clarity”: 8, “learning_transfer”: {“applies”: false, “bonus”: 0}, “total_score”: 7.6, “coments”: “Buena comparación inicial. Explica correctamente la secuencia. El razonamiento sobre la limpieza y fiabilidad del código en TDD es acertado. Claridad adecuada.”}
gemini_grade	7.6
judge_status	OK

Table 5. Cont.

Field	Example
	Nota final: 7.2/10
	Qué has hecho bien: -Has descrito correctamente la secuencia de desarrollo en el enfoque tradicional y en TDD. -Mencionaste que TDD produce código más limpio, fiable y modular, lo cual es un punto importante. En qué puedes mejorar: -Tu respuesta carece de una comparación más profunda entre ambos enfoques; por ejemplo, podrías haber mencionado cómo TDD ayuda a detectar errores más temprano en el proceso. -Hay errores de ortografía y redacción que dificultan la comprensión, como “priemo” en lugar de “primero” y “fucnionase” en lugar de “funcionase”. -Sería útil que explicaras más detalladamente por qué TDD resulta en un código más limpio y modular, ya que esto fortalecería tu argumento. Cómo podrías mejorar tu respuesta: -Intenta incluir ejemplos concretos que ilustren las diferencias entre TDD y el enfoque tradicional. -Revisa tu redacción y ortografía antes de entregar tus respuestas para asegurar que sean claras y precisas. -Considera explicar cómo la práctica de escribir pruebas primero puede influir en el diseño del código y en la reducción de bugs.
judge_raw_json	
judge_grade	7.2
teacher_grade	9.0
teacher_feedback	Comparas bien el orden (tradicional: primero código; TDD: primero tests) y añades efectos positivos en limpieza y modularidad, pero te falta comentar más explícitamente que en TDD los tests actúan como especificación/guía y que en el enfoque tradicional el testing suele ser más reactivo y a veces opcional.
teacher_corrected_at	7 December 2025 22:31:15.297000
teacher_corrected	True

Table 6. Representative record example—“dataset_en.csv”.

Field	Example
entry_id	000123
question_id	Q799823558
student_anom_id	495b02083ab12972a416b93c9c6e776abfddb1a87fd450a82cf2b6f97d89074d
question_text	Compare TDD with a traditional testing approach in software development. The traditional approach to testing was that the project code was developed first and then tests were conducted to verify that the code functioned correctly. However, the TDD approach is the opposite; tests are created first, and then the project code is developed to ensure that those tests pass. The TDD development approach results in cleaner, more reliable, and modular code. This structure, along with testing frameworks, can confirm that only the necessary code has been developed for each functionality. TDD and the traditional testing approach fundamentally differ in their sequence and purpose. In TDD, tests are written before the code, serving as specifications and guides for development. In contrast, in a traditional approach, the code is developed first and then tested, which can lead to reactive verification. Additionally, in TDD, tests are mandatory and help prevent errors from the outset, while in the traditional approach, tests may be optional and are used to validate work that has already been completed.
student_answer	
ideal_answer	

Table 6. Cont.

Field	Example
student_answer_length	577
deepseek_raw_json	{“coverage_accuracy”: 7, “justification_reasoning”: 6, “terminology_clarity”: 5, “learning_transfer”: {“applies”: false, “bonus”: 0}, “total_score”: 6.5, “comments”: “You cover the basic sequence. The purpose and conceptual differences are not clearly specified. Improve terminology and spelling. Explain why TDD produces cleaner code.”}
deepseek_grade	6.5
qwen_raw_json	{“coverage_accuracy”: 8, “justification_reasoning”: 7, “terminology_clarity”: 6, “learning_transfer”: {“applies”: false, “bonus”: 0}, “total_score”: 7.7, “comments”: “You clearly cover the sequence and advantages of TDD. Improve precision in technical terms and writing.”}
qwen_grade	7.7
gemini_raw_json	{“coverage_accuracy”: 7, “justification_reasoning”: 7, “terminology_clarity”: 8, “learning_transfer”: {“applies”: false, “bonus”: 0}, “total_score”: 7.6, “comments”: “Good initial comparison. You correctly explain the sequence. The reasoning about code cleanliness and reliability in TDD is accurate. Adequate clarity.”}
gemini_grade	7.6
judge_status	OK
	Final grade: 7.2/10
	What you did well:
	-You correctly described the development sequence in the traditional approach and in TDD.
	-You mentioned that TDD produces cleaner, more reliable, and modular code, which is an important point.
	Areas for improvement:
	-Your answer lacks a deeper comparison between both approaches; for example, you could have mentioned how TDD helps detect errors earlier in the process.
judge_raw_json	-There are spelling and writing errors that hinder comprehension, such as “priemo” instead of “primero” and “fucnionase” instead of “funcionase”.
	-It would be useful to explain more clearly why TDD results in cleaner and more modular code, as this would strengthen your argument.
	How you could improve your answer:
	-Try to include concrete examples that illustrate the differences between TDD and the traditional approach.
	-Review your writing and spelling before submitting your answers to ensure they are clear and precise.
	-Consider explaining how the practice of writing tests first can influence code design and bug reduction.
judge_grade	7.2
teacher_grade	9.0
teacher_feedback	You correctly compare the order of development (traditional: code first; TDD: tests first) and add positive effects related to cleanliness and modularity. However, you should state more explicitly that in TDD tests act as a specification/guide, and that in the traditional approach testing is usually more reactive and sometimes optional.
teacher_corrected_at	7 December 2025 22:31:15.297000
teacher_corrected	True

5. Methods

This section describes the procedures applied to collect, process, evaluate, and validate the data included in the dataset. The focus is on data generation and curation methods

rather than on the performance of specific models, in accordance with the scope of a data descriptor.

5.1. Data Collection Context

The dataset was collected from undergraduate university courses in computer science and software engineering. The assessments consisted of theoretical open-ended questions covering topics such as software development methodologies, software testing strategies, and fundamental programming concepts. Students provided free-text answers as part of regular course assessments conducted during the academic term, ensuring that the data reflect authentic educational settings rather than experimental or synthetic scenarios.

The original language of instruction was Spanish. All questions, instructor-defined ideal answers, and student responses were initially produced in Spanish during the normal course activities. No modifications were introduced to the original student submissions prior to data processing, beyond standard anonymization procedures.

5.2. Data Collection and Anonymization

Student answers were collected through the institutional learning and assessment infrastructure used in the participating courses. After data collection, all records were anonymized prior to inclusion in the dataset. Each student was assigned a non-reversible anonymous identifier that is consistent across questions but cannot be linked back to personal identities.

No personally identifiable information is included in the dataset. The data collection and release process complies with applicable ethical standards and data protection regulations for educational research, ensuring that student privacy is preserved while enabling meaningful reuse of the data for research purposes.

5.3. Translation Process

All pedagogical content in the dataset was originally authored in Spanish and subsequently translated into Standard American English to create a fully bilingual resource. The translation process aims to preserve semantic equivalence between the original and translated versions of questions, student answers, and ideal solutions.

Manual inspections were performed to verify alignment between the Spanish and English texts, with particular attention to technical terminology and domain-specific expressions. This validation step was necessary to ensure that both language versions can be reliably used for multilingual and cross-lingual research on automated assessment.

5.4. Automated Evaluation with Large Language Models

Each student answer was evaluated using multiple large language models (DeepSeek, Qwen, and Gemini) following a unified grading schema. To ensure the independence and integrity of the automated evaluations, a zero-shot prompting strategy was employed. The input provided to each model consisted strictly of three elements: the question text, the student's response, and the instructor-defined ideal answer.

Crucially, the models were blind to the human assessment: they were not exposed to the instructor's numerical grades, the instructor's feedback, or the identity of the students at any point during the generation process. This aligns with the human grading protocol, where the instructor also evaluated student answers anonymously within the learning management system to prevent potential bias. Furthermore, the automated evaluation was performed independently for each response; models did not have access to other students' answers, preventing any comparative bias or data leakage. No prior calibration or fine-tuning on a held-out subset of this dataset was performed; the models were applied using their base capabilities to adhere to the provided JSON grading schema. The

schema comprises several evaluation dimensions, including coverage accuracy, quality of reasoning and justification, and clarity of technical terminology. In addition to dimension-level assessments, each model produces a final numerical grade on a standardized 0–10 scale. Prior work has shown that large language models can approximate the grading behavior of human teaching assistants, supporting their use as auxiliary evaluators when combined with appropriate validation mechanisms [13]. Studies on automated grading of programming and theoretical assignments further support the feasibility of LLM-assisted evaluation in real educational settings [14].

The use of a common evaluation structure across models enables systematic comparison between automated graders and facilitates downstream analyses, such as agreement studies and meta-evaluation. The raw structured outputs generated by the models are included in the dataset in JSON format to support transparency and detailed inspection.

5.5. Data Preprocessing, Validation, and Quality Control

Several preprocessing and validation steps were applied to ensure data quality and internal consistency. All numerical grades, whether produced by human instructors or automated evaluators, were normalized to a common 0–10 scale. Entries with missing critical metadata (e.g., identifiers or timestamps) were excluded from the final dataset. Blank or empty student answers were intentionally retained to allow analysis of LLM behavior in response to non-responses.

Automated JSON outputs were validated to ensure syntactic correctness and compliance with the expected evaluation schema. Additional consistency checks were performed to verify the correct alignment between questions, student answers, ideal answers, and evaluations across both language versions. Manual inspections were conducted when necessary to resolve ambiguities and confirm semantic equivalence after translation.

6. User Notes and Potential Applications

The dataset is intended for research and educational purposes. Users are encouraged to consider potential biases arising from subject domain, language, and grading criteria when applying the data to new contexts. The dataset can be accessed openly through the associated repository and reused under the terms of the provided license.

The dataset supports multiple research and practical applications in the fields of educational data mining, natural language processing, and artificial intelligence for education. Its combination of real student answers, human grading, and multi-LLM evaluations enables analyses that may be challenging to conduct using currently available public resources.

A primary application of the dataset is benchmarking and comparative evaluation of LLMs for automated grading of open-ended questions. Because all models are evaluated using a common grading schema and are directly comparable to human judgments, the dataset allows systematic analysis of grading accuracy, consistency, and variability across automated evaluators.

The dataset is also appropriate for the training and evaluation of judge or meta-evaluation models, which aim to aggregate or arbitrate between multiple automated graders. The presence of both individual LLM evaluations and consolidated judge decisions provides a suitable context for studying ensemble strategies and decision aggregation.

Another potential application is the analysis of alignment between human instructors and LLM-based assessment systems. Researchers can study score distributions, qualitative feedback differences, and disagreement patterns to identify potential biases, over- or under-grading tendencies, and failure cases of automated assessment.

Although the dataset includes both human grades and LLM-generated evaluations, the present work does not aim to perform a detailed performance comparison between automated and human grading.

Future research using this dataset may include statistical analyses of the alignment between human and LLM assessments, such as correlation studies (e.g., Pearson or Spearman coefficients) or agreement metrics. Such analyses may help quantify the reliability of automated grading systems and identify potential sources of bias.

In addition, the dataset enables research on automated feedback generation, allowing qualitative analysis of LLM-generated comments in comparison with authentic teacher feedback. This permits investigation of feedback usefulness, tone, specificity, and pedagogical value.

Finally, the bilingual nature of the dataset facilitates multilingual and cross-lingual studies, such as evaluating grading performance across languages, analyzing the impact of translation on assessment quality, and developing language-agnostic educational AI systems.

7. Limitations

All pedagogical content was originally produced in Spanish and later translated into English. While care was taken to ensure semantic equivalence between language versions, minor translation artifacts may remain, such as differences in idiomatic expressions, variations in syntactic structure, or subtle shifts in lexical frequency. These aspects should be considered when conducting fine-grained linguistic analyses.

Future extensions of this dataset may include the incorporation of additional cohorts of students and the collection of data from other universities. Such expansions would enable the analysis of the robustness and generalizability of automated grading systems across different educational contexts.

Author Contributions: Conceptualization, J.C.-G. and J.S.-S.; methodology, J.C.-G. and J.S.-S.; software, J.C.-G.; validation, J.C.-G., N.G.-H., I.B.-S., Y.C. and J.S.-S.; formal analysis, J.C.-G.; investigation, J.C.-G., N.G.-H., I.B.-S., Y.C. and J.S.-S.; resources, J.C.-G. and J.S.-S.; data curation, J.C.-G., N.G.-H., I.B.-S. and Y.C.; writing—original draft preparation, J.C.-G. and J.S.-S.; writing—review and editing, J.C.-G., N.G.-H., I.B.-S., Y.C. and J.S.-S.; visualization, J.C.-G.; supervision, J.S.-S.; project administration, J.C.-G. and J.S.-S.; funding acquisition, J.C.-G. and J.S.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by 2025–2026 Innovation in Learning program at Universidad Francisco de Vitoria through the project titled “JorGPT: Accelerating assessment in programming through advanced language models (LLMs), improving the quality of feedback and teaching efficiency”.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki. The research protocol was reviewed and approved by the Research Ethics Committee of Universidad Francisco de Vitoria (Project identification code: 31/2026) on 2 March 2026.

Informed Consent Statement: Informed consent for participation was obtained from all subjects involved in the study. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Data Availability Statement: All data can be accessed without logging in at <https://doi.org/10.5281/zenodo.18369061> (accessed on 12 March 2026). The repository contains two CSV files: dataset_es.csv (Spanish) and dataset_en.csv (English).

Acknowledgments: The authors would like to thank Universidad Francisco de Vitoria for their support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Miller, I.; Choe, Y.; Emirtekin, E. Large Language Model-Powered Automated Assessment: A Systematic Review. *Appl. Sci.* **2025**, *15*, 5683. [CrossRef]
2. Schneider, J.; Richner, R.; Riser, M. Towards Trustworthy AutoGrading of Short, Multi-lingual, Multi-type Answers. *Int. J. Artif. Intell. Educ.* **2022**, *33*, 88–118. [CrossRef]
3. Cisneros-González, J.; Gordo-Herrera, N.; Barcia-Santos, I.; Sánchez-Soriano, J. JorGPT: Instructor-Aided Grading of Programming Assignments with Large Language Models (LLMs). *Future Internet* **2025**, *17*, 265. [CrossRef]
4. Shermis, M.D. State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration. *Assess. Writ.* **2014**, *20*, 53–76. [CrossRef]
5. Dzikovska, M.O.; Nielsen, R.; Brew, C.; Leacock, C.; Giampiccolo, D.; Bentivogli, L.; Clark, P.; Dagan, I.; Dang, H.T. SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. 2013. Available online: <https://aclanthology.org/S13-2045/> (accessed on 17 February 2026).
6. Seneviratne, H.M.T.W.; Manathunga, S.S. Artificial intelligence assisted automated short answer question scoring tool shows high correlation with human examiner markings. *BMC Med. Educ.* **2025**, *25*, 1146. [CrossRef] [PubMed]
7. del Gobbo, E.; Guarino, A.; Cafarelli, B.; Grilli, L. GradeAid: A framework for automatic short answers grading in educational contexts—Design, implementation and evaluation. *Knowl. Inf. Syst.* **2023**, *65*, 4295–4334. [CrossRef] [PubMed]
8. Mendonça, P.C.; Quintal, F.; Mendonça, F. Evaluating LLMs for Automated Scoring in Formative Assessments. *Appl. Sci.* **2025**, *15*, 2787. [CrossRef]
9. Byun, G.; Rajwal, S.; Choi, J.D. LLM-as-A-Grader: Practical Insights from Large Language Model for Short-Answer and Report Evaluation. 2025. Available online: <https://arxiv.org/pdf/2511.10819> (accessed on 25 January 2026).
10. Wang, Y.; Ding, Z.; Wu, X.; Sun, S.; Liu, N.; Zhai, X. AutoSCORE: Enhancing Automated Scoring with Multi-Agent Large Language Models via Structured Component Recognition. 2025. Available online: <https://arxiv.org/pdf/2509.21910> (accessed on 25 January 2026).
11. Mohler, M.; Bunesu, R.; Mihalcea, R. Learning to Grade Short Answer Questions Using Semantic Similarity Measures and Dependency Graph Alignments. 2011. Available online: <https://aclanthology.org/P11-1076/> (accessed on 17 February 2026).
12. Agarwal, D.; Gupta, S.; Baghel, N. ScAA: A Dataset for Automated Short Answer Grading of Children’s Free-Text Answers in Hindi and Marathi. 2020. Available online: <https://aclanthology.org/2020.icon-main.58/> (accessed on 17 February 2026).
13. Poličar, P.G.; Špendl, M.; Curk, T.; Zupan, B. Automated assignment grading with large language models: Insights from a bioinformatics course. *Bioinformatics* **2025**, *41*, i21–i29. [CrossRef]
14. Mohamed, K.; Yousef, M.; Medhat, W.; Mohamed, E.H.; Khoriba, G.; Arafa, T. Hands-on analysis of using large language models for the auto evaluation of programming assignments. *Inf. Syst.* **2025**, *128*, 102473. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.