Lightweight interpretable pipeline for automated short answer grading

Notebook for the ELOQUENT Lab at CLEF 2026

Filip Prášil[1,*,†], Antonín Otmar[1,*,†]

[1]Charles University Prague

Abstract

This paper presents the submission of the FiAnSo team to the CLEF 2026 Sensemaking shared task, which focuses on evaluating context-grounded student answers against specific rubrics across 13 European languages. To address the transparency and security limitations of generative Large Language Models (LLMs), we propose a computationally efficient and interpretable three-step classification pipeline. Rather than utilizing end-to-end generation, we embed the text fields independently using open-weight multilingual dense embedding models, extract an interpretable information bottleneck of 13 cosine similarity-based features, and train lightweight classifiers to predict the final discrete grade. This architecture allows for auditable grading decisions and adds inherent robustness against adversarial prompt injections. We evaluate multiple embedding–classifier combinations, achieving a Quadratic Weighted Kappa (QWK) of 0.521 and 0.420 on standard and PISA test sets respectively, by using an instructed Qwen3-Embedding-8B model paired with Logistic Regression, showing that our approach is competitive.

Keywords

short answer grading, semantic similarity, interpretable NLP, machine learning

1. Introduction

Automated grading of open-ended student responses remains a significant challenge in educational technology. A reliable automated grader must evaluate answers accurately, fairly, and transparently based on the provided reference materials. While generative Large Language Models (LLMs) can provide satisfactory results, they typically lack the transparency required to audit grades or identify biases, and are highly susceptible to adversarial attacks.

This paper details our participation as the team FiAnSo in the Sensemaking 2026 shared task [1], which explores the challenges of creating systems to automatically evaluate multilingual, context-grounded student answers.

To address the transparency issues inherent in generative LLMs, we propose a computationally efficient three-step pipeline. Rather than relying on end-to-end generation, we decouple the text representation step from the final grading decision. First, we embed the text fields independently using open-weights multilingual dense embedding models. Next, we extract 13 cosine similarity features from the embeddings. Finally, we train lightweight classifiers on these similarity features and use them to predict the final grade.

This architecture helps to ensure that the grading decisions are auditable and secure against adversarial prompt injections. Furthermore, it proved effective: our best configuration achieved a peak Quadratic Weighted Kappa (QWK) of 0.521 and 0.420 on standard and PISA test sets respectively.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

†These authors contributed equally.

fi.prasil@gmail.com (F. Prášil); antonin.otmar322@student.cuni.cz (A. Otmar)

2. Related Work

An overview of approaches to solve tasks similar to ours is given in Haller et al. [2]. While the earliest systems usually used syntactic overlap, this breaks down with more open questions or with multilingualism, where semantic similarity scoring is necessary.

The first works to use semantic similarity to grade answers (e.g. Mohler and Mihalcea [3]) relied on latent semantic analysis (LSA) semantic similarity. These approaches often required significant per question learning.

A notable work similar to our approach of embedding pieces of text independently with a generalist non-finetuned backbone and clustering them later in the pipeline, although without using it to grade answers, is Reimers and Gurevych [4]. We are applying a method inspired by this work to a new domain in combination with newest embedding models.

3. Task and Dataset

The Sensemaking 2026 task focuses on the automatic evaluation of student responses to open-ended questions based on input materials. It is divided into two tracks:

- **Simple Rating track**: graders score student answers on a scale 0-4 based on whether it is an answer to the question that correctly reflects information in the context, without explicit guidelines saying how good an answer needs to be to get a particular grade.
- **Rubric-Based Rating track**: graders score student answers on a scale 0-2, against rubrics that contain grading guidelines.

Our participation focuses entirely on the Rubric-Based Rating track.

3.1. Dataset Structure

The Rubric-Based Rating Track formulates the grading as a ternary classification problem. Each answer is labeled as No Credit (0), Partial Credit (1), or Full Credit (2). Each input instance contains the following fields:

- **Context**: text containing all the necessary information to answer the question.
- **Question**: the asked question.
- **Answer**: the student's answer to be graded.
- **Full Credit (FC) Rubric**: text describing what a full credit answer should look like.
- **Partial Credit (PC) Rubric**: text describing what a partial credit answer should look like.
- **No Credit (NC) Rubric**: text describing what a no credit answer should look like.

The dataset has a couple of notable properties. Mainly, it is **highly multilingual**. It contains 13 European languages, including low-resource ones like Irish Gaelic. Another notable property is that part of the test set is **confidential**. This constraint explicitly prohibits the use of commercial APIs, meaning all the evaluation must be done locally, which substantially limits the available computational resources and model sizes.

The dataset is divided into four splits: *training*, *development*, *test*, and *PISA test*. An overview of the dataset statistics across these splits is presented in Table 1.

4. Methodology

We designed our approach with these limitations in mind. Specifically, our focus was on creating an approach that is computationally efficient and handles multilinguality natively. Additionally, we wanted our approach to be interpretable and robust to adversarial inputs. The resulting approach consists of three sequential steps:

Split	# Samples	# Domains	Language Distribution	Confidential
Training	∼13k	2	English slightly overrepresented	No
Development	∼4k	5	Roughly uniform	No
Test	∼47k	8	English slightly overrepresented	No
PISA Test	—	—	—	Yes

Table 1
Statistical overview of the dataset splits, including sample counts, number of covered domains, language distribution, and confidentiality.

1. **Embedding Creation**: Generating dense vector representations for individual text fields.
2. **Similarity Feature Extraction**: Computing explicit similarity metrics from the embeddings.
3. **Classification**: Using a simple prediction model on these similarity features to make the final prediction.

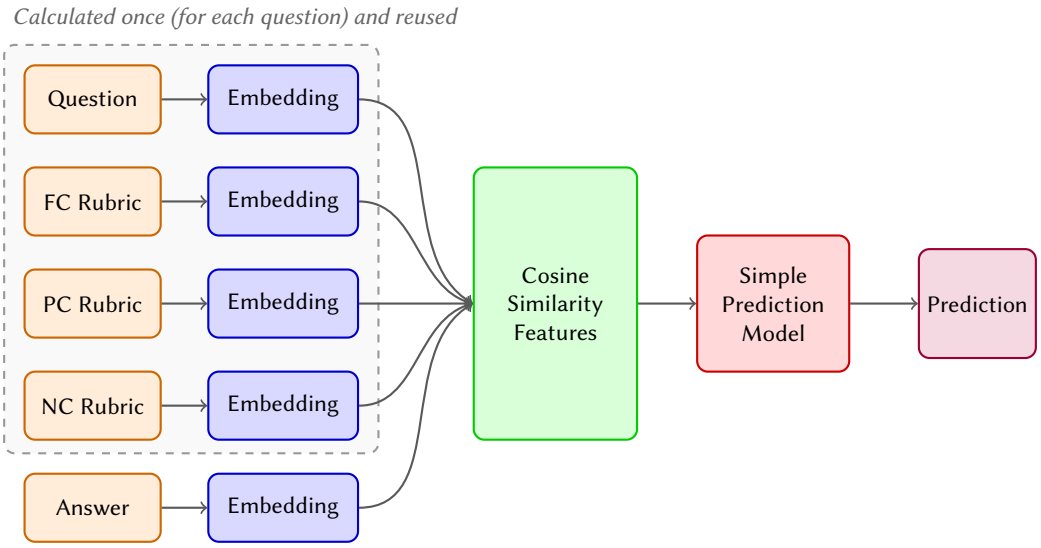

Figure 1: Diagram of the proposed pipeline.

4.1. Embedding Creation

The first step in our pipeline converts the raw text data into vector representations. Instead of combining the individual text fields into a single prompt, we decided to embed each field separately. While a cross-field encoding would allow for cross-referencing of potentially relevant information, it would also be substantially more difficult to interpret. Single-field encoding also allows for a substantial reduction in the required compute. Because the dataset contains multiple answers for each question, we can save on most of the computational resources by simply caching the question-specific fields and recomputing the embedding only for the answer itself.

For the embedding process, we utilize the question, answer, and each of the rubrics (FC, PC, and NC). We explicitly discard the context field, as preliminary testing indicated that its presence does not significantly improve accuracy, while its length presents substantial additional memory and computational overhead.

We rely on state-of-the-art multilingual dense embedding models to create the embeddings. Specifically, we evaluate two model families:

- **BGE-M3** [5]: an XLM-RoBERTa-based [6] encoder-only transformer model trained via multi-stage contrastive learning.

- **Qwen3-Embedding** [7]: a decoder-only transformer model derived from the Qwen3 [8] language model family, pre-trained on massive multilingual corpora and fine-tuned for instruction following. We test all three available variants of the Qwen3-Embedding models: 0.6 billion [9], 4 billion [10], and 8 billion [11] parameter versions.

The Qwen3-Embedding models are instruction-aware, meaning that their input instructions can be changed to better suit the given task. For these models, we evaluate both the uninstructed baseline and the instructed variant, specifically using the instruction:

> "Given a text, retrieve relevant information for semantic grading"

Model	Context window	Embedding dim	Parameters	Instruction aware
BGE-M3	8,192	1,024	∼570M	No
Qwen3-Embedding-0.6B	32,768	1,024	∼600M	Yes
Qwen3-Embedding-4B	32,768	2,560	∼4B	Yes
Qwen3-Embedding-8B	32,768	4,096	∼8B	Yes

Table 2
Properties of the used embedding models.

4.2. Similarity Feature Extraction

In the second step, the high-dimensional embedding vectors are transformed into more compact and interpretable features. We do so by calculating cosine similarities between each unique pair of the text field embeddings (Question, Answer, FC Rubric, PC Rubric, and NC Rubric). This yields 10 pairwise similarity features. We also include 3 additional features to represent the relative closeness between the answer and each of the rubrics by calculating differences in Answer-Rubric similarities:

$$\cos(\text{Answer}, \text{FC Rubric}) - \cos(\text{Answer}, \text{NC Rubric})$$

$$\cos(\text{Answer}, \text{FC Rubric}) - \cos(\text{Answer}, \text{PC Rubric})$$

$$\cos(\text{Answer}, \text{PC Rubric}) - \cos(\text{Answer}, \text{NC Rubric})$$

Where $\cos$ is the cosine similarity between the embeddings of the two texts.

This gives us a set of 13 features, each with a clear semantic meaning, creating an interpretability bottleneck. A possible interpretation for each of the similarity features is presented in Appendix A.

4.3. Classification

The final stage of the pipeline uses a simple classification model on the 13 similarity features to predict the corresponding grade. To evaluate the discriminative power of our engineered features, we test a progression of three models:

- **K-Nearest Neighbors (KNN)** serves as an example of a non-parametric model.
- **Logistic Regression** serves as a slightly more complex, parametric, but linear model.
- **Multi-Layer Perceptron (MLP)** serves as our most powerful model that can model non-linear relationships. Note that we only used a single hidden layer, to keep the model relatively simple.

For each of the 21 unique configurations (7 embedding methods $\times$ 3 classifiers), we performed a grid search to identify the optimal hyperparameters. The specific parameter spaces explored are detailed in Table 3. Driven by the official evaluation target of the Sensemaking 2026 task, we optimized for Quadratic Weighted Kappa (QWK).

Classifier	Hyperparameter	Values searched
KNN	k (neighbors)	1, 3, 5, 10, 15, 25
	Distance metric	Euclidean (fixed)
Logistic Regression	C (regularization)	0.01, 0.1, 1, 10, 100
	Class weight	none, balanced
MLP	Hidden layer width	32, 64, 128, 256
	Activation	ReLU (fixed)
	α (L2 penalty)	10^{-4}, 10^{-3}

Table 3
Hyperparameter grid search spaces for each of the three classifiers.

5. Results

5.1. Development Set Performance

Table 4 reports the development set performance for all embedding–classifier combinations. Across all configurations, the KNN baseline consistently performed the weakest. When comparing the linear (Logistic Regression) and non-linear (MLP) classifiers, an interesting trend can be seen. For the BGE-M3 embeddings, the MLP classifier (QWK = 0.459) noticeably outperformed Logistic Regression (QWK = 0.434). Conversely, for the Qwen3-Embedding models, Logistic Regression generally matched or exceeded the performance of the MLP across all configurations. This contrast suggests that the similarity feature space produced by BGE-M3 is more complex and cannot be properly modeled by a linear model, whereas the Qwen3-Embedding models produce a simpler, linearly separable feature space where Logistic Regression suffices.

5.1.1. Scaling Effects

Within the Qwen3-Embedding family, performance scaled consistently with model size. When using the uninstructed baseline and Logistic Regression, QWK improved from 0.461 for the 0.6B model to 0.493 for the 4B model, reaching 0.521 for the 8B model.

5.1.2. Instruction Effects

Using the custom instruction prompt revealed a strict capacity-dependent effect. For the smallest Qwen3-Embedding-0.6B model, the prompt caused a significant performance degradation across all classifiers (e.g., Logistic Regression dropped from 0.461 to 0.309). This indicates that the 0.6B model lacks the capacity to follow the instruction without distorting its internal representations.

For the 4B model, the effect was mixed: Logistic Regression improved ($0.493 \rightarrow 0.536$), while KNN worsened, and the MLP gain was marginal. Finally, for the Qwen3-Embedding-8B model, the instruction prompt consistently improved all three classifiers (Logistic Regression: $0.521 \rightarrow 0.581$, KNN: $0.348 \rightarrow 0.414$, MLP: $0.501 \rightarrow 0.548$). This confirms that sufficiently large models can successfully utilize the prompt to create a more task-aligned representation space.

Ultimately, the best overall result was achieved by combining the instructed Qwen3-Embedding-8B model with Logistic Regression, resulting in a peak QWK of 0.581 ± 0.023.

5.2. Test Set Performance and Comparison

Due to development time constraints, our official submission to the shared task was the uninstructed configuration of the Qwen3-Embedding-8B model paired with a Logistic Regression classifier. Post-deadline, we obtained overall test set results for the instructed variant of the same model to evaluate the generalizability of our best-performing development configuration.

Model	Variant	Classifier	QWK	κ	Acc
BGE-M3	—	KNN	0.265 ± 0.028	0.216 ± 0.025	0.443 ± 0.016
		Logistic Regression	0.434 ± 0.024	0.358 ± 0.021	0.503 ± 0.016
		MLP	0.459 ± 0.024	0.390 ± 0.022	0.540 ± 0.015
Qwen3-Embedding-0.6B	no instruct	KNN	0.335 ± 0.027	0.276 ± 0.023	0.481 ± 0.015
		Logistic Regression	0.461 ± 0.027	0.392 ± 0.024	0.550 ± 0.016
		MLP	0.451 ± 0.026	0.390 ± 0.023	0.564 ± 0.015
	instruct	KNN	0.218 ± 0.030	0.183 ± 0.026	0.452 ± 0.016
		Logistic Regression	0.309 ± 0.026	0.272 ± 0.023	0.482 ± 0.015
		MLP	0.320 ± 0.030	0.280 ± 0.026	0.508 ± 0.016
Qwen3-Embedding-4B	no instruct	KNN	0.401 ± 0.027	0.338 ± 0.024	0.524 ± 0.015
		Logistic Regression	0.493 ± 0.026	0.417 ± 0.023	0.564 ± 0.016
		MLP	0.461 ± 0.027	0.392 ± 0.024	0.563 ± 0.015
	instruct	KNN	0.364 ± 0.028	0.296 ± 0.025	0.493 ± 0.016
		Logistic Regression	0.536 ± 0.024	0.443 ± 0.022	0.568 ± 0.015
		MLP	0.476 ± 0.027	0.405 ± 0.024	0.569 ± 0.015
Qwen3-Embedding-8B	no instruct	KNN	0.348 ± 0.029	0.291 ± 0.024	0.499 ± 0.015
		Logistic Regression	0.521 ± 0.025	0.448 ± 0.024	$\mathbf{0.590 \pm 0.015}$
		MLP	0.501 ± 0.027	0.424 ± 0.024	0.580 ± 0.015
	instruct	KNN	0.414 ± 0.026	0.340 ± 0.023	0.516 ± 0.015
		Logistic Regression	$\mathbf{0.581 \pm 0.023}$	$\mathbf{0.482 \pm 0.021}$	0.585 ± 0.015
		MLP	0.548 ± 0.024	0.461 ± 0.023	0.588 ± 0.016

Table 4
Development set results. QWK = quadratic weighted kappa; κ = linear Cohen's kappa; Acc = accuracy. Values are point estimates $\pm$ half-width of the 95% percentile bootstrap CI (1,000 resamples). Bold marks the best value per metric.

On the official test set, our submitted uninstructed model achieved an overall Quadratic Weighted Kappa (QWK) of 0.476 on the standard split and 0.391 on the PISA split. In comparison, the post-hoc evaluation of the instructed configuration demonstrated a clear performance leap, yielding an overall QWK of 0.521 on the standard split and 0.420 on the PISA split. This consistent margin confirms that the benefits of task-specific instructions observed during development successfully transfer to unseen evaluation data. Because per-language and domain results were only provided for our official submission, the subsequent domain and language breakdowns focus on the uninstructed baseline model.

5.2.1. Domain Effects

Comparing the development and test set results (Table 5) reveals significant differences across domains. Although the performance on the test set is uneven, it mirrors trends observed in the organizers' baselines and other submissions, indicating that these variations likely reflect the innate difficulty of each text collection. As expected, most per-domain results deteriorated on the test set due to mild overfitting to the development set during model and hyperparameter selection. However, the `popular_science` and `popular` domains exhibited the opposite trend, suggesting that some performance differences stem from inherent data distribution shifts between the splits rather than pure overfitting.

5.2.2. Language Effects

An analysis of the pipeline's performance across the 13 evaluated languages (Table 6) highlights further variations in model robustness. Notably, while performance on the development set seems to be roughly equal (with the exception of Irish), it varies significantly on the test set. The relatively low performance on Irish can likely be attributed to the fact that it is a low-resource language, which makes the encoder's performance worse, consequently lowering our pipeline's performance. The rest of the performance variation across languages does not seem to have an obvious explanation. However, similar variance can

Table 5
Development and Test set performance of the submitted uninstructed Qwen3-Embedding-8B model separated by domain.

	Dev			Test		
Domain	**Acc.**	κ	**QWK**	**Acc.**	κ	**QWK**
world-history	0.664	0.576	0.673	0.511	0.296	0.503
popular_science	0.569	0.368	0.441	0.652	0.424	0.608
demagog	0.645	0.521	0.587	0.557	0.317	0.523
popular	0.526	0.298	0.395	0.510	0.290	0.508
OECD_public	0.331	0.238	0.358	—	—	—
ukr	—	—	—	0.531	0.320	0.541
counseling	—	—	—	0.500	0.263	0.439
pisa	—	—	—	0.482	0.208	0.391
law_faculty	—	—	—	0.513	0.188	0.272

be seen in other submissions and some of the organizers' baselines, again pointing towards differences in data distribution.

Table 6
Development and Test set performance of the submitted uninstructed Qwen3-Embedding-8B model separated by language.

	Dev			Test		
Lang.	**Acc.**	κ	**QWK**	**Acc.**	κ	**QWK**
ro	0.592	0.444	0.514	0.547	0.335	0.547
de	0.630	0.508	0.585	0.519	0.267	0.481
uk	0.603	0.467	0.537	0.535	0.263	0.475
cs	0.599	0.466	0.544	0.505	0.245	0.460
pt	0.604	0.458	0.526	0.505	0.243	0.459
sr	0.611	0.495	0.580	0.508	0.240	0.454
da	0.627	0.496	0.561	0.514	0.253	0.451
sv	0.583	0.442	0.519	0.510	0.249	0.448
hu	0.578	0.439	0.519	0.505	0.234	0.442
el	0.580	0.421	0.482	0.497	0.225	0.407
fi	0.596	0.451	0.514	0.483	0.214	0.394
en	0.584	0.461	0.561	0.495	0.228	0.386
ga	0.477	0.279	0.326	0.397	0.090	0.239

6. Conclusion

In this paper, we presented a computationally efficient and semantically interpretable pipeline for rubric-based answer grading, developed for the Sensemaking 2026 shared task. Our official submission utilized an uninstructed Qwen3-Embedding-8B model with a Logistic Regression classifier and achieved strong test set results of QWK $= 0.476$ on the standard split and QWK $= 0.391$ on the PISA split, placing at the top of non-generative LLM submissions performance-wise. Our post-hoc evaluation of the instructed model version improved on this, achieving QWK $= 0.521$ and QWK $= 0.420$ on the standard and PISA splits respectively.

Our experiments demonstrated that the use of task-specific instructions exhibits a capacity-dependent effect, degrading smaller models while significantly boosting the 8B parameter model. Furthermore, we observed that the similarity space produced by the Qwen3-Embedding models is highly linearly separable, allowing a simple Logistic Regression classifier to excel. Conversely, the similarity space

generated by BGE-M3 contained non-linearities that required a more expressive Multi-Layer Perceptron to parse effectively.

This presents an interesting direction for future work. It warrants investigating whether this (non)linearity in the similarity spaces is only an artifact of the specific dataset or whether it represents a generalizable property that holds across other educational domains and task formulations.

Acknowledgments

We want to thank Ondřej Bojar and Pavel Šindelář at the Institute of Formal and Applied Linguistics at Charles University for organizing the shared task and providing valuable feedback to the participants. We also thank Sothi Khorn for his contributions to our initial task submission.
This work was partially supported from the Project OP JAK Mezisektorová spolupráce Nr. CZ.02.01.01/00/23_020/0008518 named "Jazykověda, umělá inteligence a jazykové a řečové technologie: od výzkumu k aplikacím", and from the EuroHPC JU grant Czech AI Factory (CZAI), reg. nr. 10131474. Computational resources were partially provided by the e-INFRA CZ project (ID: 90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

Declaration on Generative AI

During the preparation of this work, the authors used Claude AI and Gemini in order to: Draft abstract, Draft content, Paraphrase and reword, and Improve writing style. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] P. Šindelář, O. Bojar, S. Ettejjari, L. F. V. Madriz, M. Piacentini, K. Thomas, Overview of the PISA Sensemaking Task at the ELOQUENT 2026 lab for evaluating generative language model quality, in: E. S. Salido, A. Barrón-Cedeño, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2026), CEUR-WS, 2026.

[2] S. Haller, A. Aldea, C. Seifert, N. Strisciuglio, Survey on automated short answer grading with deep learning: from word embeddings to transformers, 2022. URL: https://arxiv.org/abs/2204.03503. doi:10.48550/arXiv.2204.03503. arXiv:2204.03503.

[3] M. Mohler, R. Mihalcea, Text-to-text semantic similarity for automatic short answer grading, in: A. Lascarides, C. Gardent, J. Nivre (Eds.), Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), Association for Computational Linguistics, Athens, Greece, 2009, pp. 567–575. URL: https://aclanthology.org/E09-1065/.

[4] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410/. doi:10.18653/v1/D19-1410.

[5] BAAI, BGE-M3 model card, https://huggingface.co/BAAI/bge-m3, 2024. Accessed: 2026-05-24.

[6] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747/. doi:10.18653/v1/2020.acl-main.747.

[7] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, et al., Qwen3 embedding: Advancing text embedding and reranking through foundation models, arXiv preprint arXiv:2506.05176 (2025).

[8] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, Z. Qiu, Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[9] Qwen Team, Qwen3-Embedding-0.6B model card, https://huggingface.co/Qwen/Qwen3-Embedding-0.6B, 2025. Accessed: 2026-05-24.

[10] Qwen Team, Qwen3-Embedding-4B model card, https://huggingface.co/Qwen/Qwen3-Embedding-4B, 2025. Accessed: 2026-05-24.

[11] Qwen Team, Qwen3-Embedding-8B model card, https://huggingface.co/Qwen/Qwen3-Embedding-8B, 2025. Accessed: 2026-05-24.

A. Interpretation of Similarity Features

- $\cos(\textbf{Answer}, \textbf{FC Rubric})$: Measures how semantically similar the student answer is to the FC rubric descriptor. A high value indicates the answer uses vocabulary and concepts consistent with what a fully correct response is expected to contain.
- $\cos(\textbf{Answer}, \textbf{PC Rubric})$: Measures how semantically similar the student answer is to the partial credit rubric descriptor. A high value indicates the answer contains at least some part of what a fully correct response is expected to contain but not all of it.
- $\cos(\textbf{Answer}, \textbf{NC Rubric})$: Measures how similar the student answer is to the no credit rubric descriptor. A high value may indicate the answer reflects misconceptions or incorrect reasoning described by the NC rubric. In practice usually less useful than the previous two as the no credit rubric is often very general.
- $\cos(\textbf{Question}, \textbf{FC Rubric})$: Captures how tightly the FC rubric descriptor is tied to the specific question topic. A high value means the FC rubric is question-specific; a low value suggests it contains generic grading language that may not be particularly useful.
- $\cos(\textbf{Question}, \textbf{PC Rubric})$: Captures how tightly the PC rubric descriptor is tied to the specific question topic. A high value means the PC rubric is question-specific; a low value suggests it contains generic grading language that may not be particularly useful.
- $\cos(\textbf{Question}, \textbf{NC Rubric})$: Captures how tightly the NC rubric descriptor is tied to the specific question topic. A high value means the NC rubric is question-specific; a low value suggests it contains generic grading language that may not be particularly useful.
- $\cos(\textbf{Answer}, \textbf{Question})$: Measures whether the student answer is on-topic with regards to the question. A very low value may indicate an off-topic or non-answer response.
- $\cos(\textbf{FC Rubric}, \textbf{PC Rubric})$: Reflects how close the FC and PC rubrics are to each other. A high value means the two rubric levels are semantically close, which reduces the discriminative power of the answer–rubric similarity features for separating FC from PC labels. This can in principle serve as a way to normalize the rubric-question and rubric-answer distances.
- $\cos(\textbf{FC Rubric}, \textbf{NC Rubric})$: Reflects how close the FC and NC rubrics are to each other. A high value means the two rubric levels are semantically close, which reduces the discriminative power of the answer–rubric similarity features for separating FC from NC labels. This can in principle serve as a way to normalize the rubric-question and rubric-answer distances.
- $\cos(\textbf{PC Rubric}, \textbf{NC Rubric})$: Reflects how close the PC and NC rubrics are to each other. A high value means that the two rubric levels are semantically close, which reduces the discriminative power of the answer–rubric similarity features for separating PC from NC labels. This can in principle serve as a way to normalize the rubric-question and rubric-answer distances.

- $\cos(\textbf{Answer}, \textbf{FC Rubric}) - \cos(\textbf{Answer}, \textbf{NC Rubric})$: Measures the relative semantic alignment of the answer toward full credit versus no credit. A strong positive value confidently suggests a correct answer, filtering out any baseline semantic similarity the answer might share with both rubrics.
- $\cos(\textbf{Answer}, \textbf{FC Rubric}) - \cos(\textbf{Answer}, \textbf{PC Rubric})$: Measures the relative closeness of the answer to the full credit rubric compared to the partial credit rubric. This feature explicitly helps the classifier separate a perfect answer from one that is only partially correct.
- $\cos(\textbf{Answer}, \textbf{PC Rubric}) - \cos(\textbf{Answer}, \textbf{NC Rubric})$: Measures whether the answer leans more toward partial credit or no credit. A positive value serves as a strong signal that the student demonstrated at least some valid knowledge, separating partial understanding from a complete lack of knowledge.