Metric Learning for Automated Rubric-based Answer Grading: A Bi-Encoder Approach

Pythoneers @ ELOQUENT 2026 Task 3: Topical PISA QUIZ

Zerina Ahmetović[1,*], Mirza Olovčić[1,*]

[1]*Charles University, Malostranské nám. 25, 118 00 Prague, Czechia*

Abstract

Traditional classification approaches often struggle with the nuanced nature of automated short-answer grading (ASAG), motivating the exploration of alternative methods. In this paper, we present our approach for the CLEF 2026 Topical PISA Quiz Task, based on metric learning with pre-trained bi-encoder architectures. Our experiments combine ModernBERT for English grading with the BAAI/bge-m3 model for multilingual evaluation. For the English branch, the strongest development results were obtained with Multiple Negatives Ranking (MNR) loss together with a Question-Aware Batching (QAB) strategy. We also investigate "context dilution", observing that removing long background contexts and focusing on question–answer and question–rubric pairs produced cleaner embedding representations. The final pipeline routes samples to different models depending on the language of the incoming query. While the English metric-learning setup showed strong development performance, multilingual metric learning remained more limited, suggesting that further work is needed to understand whether the bottleneck lies in the multilingual backbone, training data, or cross-lingual transfer of the metric-learning objective.

Keywords

metric learning, short-answer grading, triplet loss, contrastive learning, multiple negatives ranking

1. Introduction

Automated grading remains challenging because of the complexity of the task and subjectivity among graders. Inspired by the principles of metric learning [1], we explore how this idea translates to encoder-based models from the BERT family and how such models perform when used as graders. We train the model to push FC (full credit) answers farther from NC (no credit) answers, while positioning PC (partial credit) answers between the two in embedding space.

During the experiments, the major challenges we faced concerned overfitting, which was reduced by choosing the right loss function, and the unbalanced structure of the input data, where most tokens came from the context. This context proved to be largely unnecessary, as the rubrics contained most of the information needed for successful classification.

The main challenge in automated short-answer grading is that traditional classification approaches often fail to capture nuances in answer quality, mainly because a partial-credit answer may be semantically closer to a full-credit answer than to a no-credit answer. To address this issue, we explore an approach based on **metric learning**, hypothesising that learning a structured semantic embedding space can produce more robust grading decisions.

- We demonstrate a metric-learning, bi-encoder-based approach for rubric-based short-answer grading.
- We introduce Question-Aware Batching (QAB), an algorithmic sampling strategy that enforces within-question semantic resolution during training.

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

*Corresponding authors.

zerina.ahmetovic921@student.cuni.cz (Z. Ahmetović); mirza.olovcic790@student.cuni.cz (M. Olovčić)

https://github.com/inferno73 (Z. Ahmetović); https://github.com/Mirza02 (M. Olovčić)

- We provide an empirical analysis of "context dilution" in transformer-based embedding spaces, showing that excessive background text can be ineffective or even harmful for short-answer evaluation.

2. Methods

2.1. Implementation Details

The entire system is implemented in Python 3.10 with the following widely used libraries:

- **PyTorch** [2] — serves as the underlying deep learning framework, providing GPU-accelerated tensor operations, automatic differentiation, and the foundation on which all model computations run.
- **sentence-transformers** [3] — used for loading and fine-tuning the bi-encoder models. It provides the `SentenceTransformer` wrapper, which applies **mean pooling** followed by ℓ_2 normalisation over the final-layer token representations to produce fixed-size sentence embeddings, as well as the `SentenceTransformerTrainer` training loop and the Multiple Negatives Ranking (MNR) loss implementation [3].
- **Hugging Face transformers** [4] — provides the underlying pre-trained model architectures (ModernBERT, mDeBERTa-v3, BAAI/bge-m3) and training utilities such as learning rate schedulers and callback interfaces used during fine-tuning.
- **Hugging Face datasets** [5] — used to construct and cache `Dataset` objects compatible with the `SentenceTransformerTrainer`, enabling efficient batched data loading and on-disk caching of mined hard negatives.
- **scikit-learn** [6] — used for computing linear-weighted and quadratic Cohen's kappa coefficients as additional evaluation metrics alongside grading accuracy.

2.2. Data Preprocessing & The "Context Dilution" Hypothesis

Before training, the raw English training and development datasets were merged and cleaned to remove artefacts that could distract the model, such as bare URLs and figure/table references. The data was then regrouped by context ID and split into training and held-out development portions with a 90/10 ratio; throughout the remainder of this paper, the 10% held-out portion is referred to as the development set and is used for model evaluation.

During the experiment phase, we noticed that the background context accounted for approximately 70% of the input token space (while the remaining space accounted for the supplied question, answer, and rubrics). The dominance of context degraded the model's ability to focus on grading. The grading-relevant parts of each sample, namely the question, answer, and rubric, were overshadowed by the context, causing the pooled representations to drift toward the semantic centroid of the background passage rather than the task objective. This phenomenon, known as representation dilution [7, 8], supports this claim, and suggests dynamic context truncation as its solution.

Initially, we attempted to use large language models (Phi-3.5 [9] and FLAN-T5 [10]) to either summarise the extensive background contexts or create "enriched rubrics" with the additional information provided by the context. We provided system messages guiding the model to *make each criterion self-contained by replacing vague references, such as "the passage," "the text," or "the article," with the specific facts, names, events, or claims from the passage that the criterion actually refers to.* We also imposed restrictions such as *do not copy the passage*, *do not add new information*, *output only the rewritten rubric in the exact same structure as the input*, and *keep each criterion concise, using one sentence per level.*

Summarising the context introduced noise because the model often failed to capture the essential information, at times producing irrelevant content. Rubric enrichment extended the rubrics in a sensible way, but it did not improve performance over removing the context entirely. In the end, completely omitting the context proved to work best, as most rubrics already contained sufficient information

to evaluate the answer on their own. We trained the model strictly on *[Question + Answer]* versus *[Question + Rubric]* pairs, which drastically cleaned the embedding space.

2.3. Core Architecture: Siamese Network & Metric Learning

The central component of the architecture is the Siamese network setup. Instead of providing the rubric and the answer to the model at the same time, as in a cross-encoder, three separate, independent forward passes are performed through the exact same model weights. This setup aligns with the Anchor, Positive, and Negative structure [11] associated with triplet loss:

1. The Anchor (pass 1): consists of the Question + Rubric string,
2. The Positive (pass 2): consists of the Question + Correct Student Answer string,
3. The Negative (pass 3): consists of the Question + Incorrect Student Answer string.

As the weights are shared, the model learns a unified semantic space. The losses then push the Positive vector closer to the Anchor while keeping the Negative vector farther away.

Note that independent triplets are created for each grade level. For example, a Partial Credit (PC) triplet uses a PC-specific rubric as the Anchor, a known PC student answer as the Positive, and sampled FC or NC answers as Hard Negatives. This diversity encourages the model to further explore and place all three categories (FC, PC, NC) in the correct embedding space.

Before encoding, each input is formatted according to its role in the Siamese setup:

- **Student-answer input:** `Answer: {answer} | Question: {question}`
- **Rubric-anchor input:** `Question: {question} | Rubric: {rubric_text}`

2.4. Model Backbones

Three encoder backbones that cover the main requirements of the task were evaluated: strong English semantic representations, multilingual transfer, and efficient embedding-based retrieval.

ModernBERT-base [12] (`answerdotai/ModernBERT-base`) is a modernized bidirectional encoder-only transformer pre-trained on 2 trillion tokens of English text with a native context window of 8,192 tokens. The model architecture consists of 22 transformer layers and 149 million parameters, and incorporates Rotary Positional Embeddings (RoPE) together with alternating local–global attention, providing efficient long-sequence processing. Its extended context window and strong English representations make it well-suited as the primary encoder for the English grading branch of our system.

mDeBERTa-v3-base [13] (`microsoft/mdeberta-v3-base`) is the multilingual variant of DeBERTa-v3, featuring 12 transformer layers and a hidden size of 768. The model introduces *disentangled attention*, which encodes token content and relative position as separate vectors, improving cross-lingual transfer. It was pre-trained on 2.5 TB of the CC100 multilingual corpus covering 100 languages and has a total of 280 million parameters, of which 190 million reside in the embedding layer due to its large 250K-token multilingual vocabulary.

BAAI/bge-m3 [14] is a fine-tuned variant of XLM-RoBERTa-large with approximately 567 million parameters, 24 transformer layers, and hidden size 1,024. It natively supports over 100 languages and extends the standard XLM-RoBERTa context window from 512 to 8,192 tokens. In our setup, sequences were truncated to 512 tokens to manage memory within the available hardware constraints. We use only its dense retrieval head, which produces L2-normalised `[CLS]` token embeddings for cosine similarity comparison.

2.5. Loss Functions & Optimisation Strategies

Since the evaluated models differed in stability and capacity, we explored two optimisation strategies: one designed for numerically fragile models such as mDeBERTa, and a more refined strategy for our strongest candidate, ModernBERT.

2.5.1. mDeBERTa: Dynamic Margin Triplet Loss

Early experimentation revealed that mDeBERTa-v3 was prone to numerical instability when trained with standard loss functions. To address this, we employed a custom **Dynamic Margin Triplet Loss**. Rather than applying a static margin, this loss function dynamically scales the margin based on the absolute grade difference between the Positive and Negative answers.

Let a, p, and n represent the embeddings of the Anchor, Positive, and Negative texts, respectively. Let g_p and g_n represent their corresponding integer grade values (where NC=0, PC=1, and FC=2). The loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = \max(0, d(a, p) - d(a, n) + m_{dynamic}) \tag{1}$$

where $d(x, y)$ is the cosine distance ($1 - \cos(x, y)$), and the dynamic margin is calculated as:

$$m_{dynamic} = m_{base} \times |g_p - g_n| \tag{2}$$

With a baseline margin $m_{base} = 0.3$, an FC answer ($g_p = 2$) and an NC answer ($g_n = 0$) are forced apart by a margin of 0.6, while an FC and PC answer are forced apart by a margin of 0.3. This explicitly encodes the ordinal nature of the grading scale into the embedding space.

Furthermore, to prevent catastrophic forgetting and reduce overfitting on our relatively small training set, we applied **Layer Freezing** to the mDeBERTa architecture. Specifically, we froze the weights of the bottom half of the transformer layers (layers 0 through 5), restricting gradient updates strictly to the upper, task-specific layers. By doing so, the model stabilised for a few epochs, but in the end, this was not sufficient to fully stabilise the training process.

2.5.2. ModernBERT: MNR Loss and Question-Aware Batching

For our primary English model, ModernBERT, we used the highly efficient **Multiple Negatives Ranking (MNR) Loss**. MNR loss uses all other positive samples in a given training batch as implicit negative samples, optimising the model to maximise the similarity between the true (a_i, p_i) pair while simultaneously minimising its similarity to all p_j (where $j \neq i$).

To further enhance MNR loss, we developed a sampling strategy named **Question-Aware Batching (QAB)**. In standard random batching, in-batch negatives often belong to completely different questions, rendering them mathematically "easy" to push away. QAB dynamically organises the dataset so that a single batch explicitly contains FC, PC, and NC answers *for the exact same question.* This forces the MNR loss to resolve within-question semantic contradictions in every gradient step, leading to significantly sharper decision boundaries.

To further enhance QAB, we implemented a **Dynamic Hard-Question Tracking** callback. During scheduled evaluation steps, questions from the training set that fell below a 67% accuracy threshold were dynamically flagged as “hard”, and their sampling frequency was multiplied by a factor of 3 in subsequent training epochs. This forced the model to prioritise hard-to-classify training questions and contributed to the peak accuracy of 91.33% observed for **v2**. The held-out development set was evaluated during these steps only to monitor training progress; its scores did not trigger early stopping, checkpoint selection, or changes to the training procedure, and all runs completed the full three-epoch training schedule.

2.6. Model Selection, Baselines & Hyperparameters

We primarily used state-of-the-art English models from the BERT family due to their numerical stability, extended context windows, and deep semantic understanding.

To establish a performance floor before applying any metric-learning techniques, we evaluated the pre-trained models (without any task-specific fine-tuning) of ModernBERT (both base and large) [12] and mDeBERTa-v3 [13] directly on the development set using zero-shot cosine similarity.

To prevent overfitting on the training distribution, we regularised our models by tuning hyperparameters, specifically using Attention Dropout, MLP Dropout, Weight Decay, and Cosine Learning Rate Schedulers.

2.7. Multilingual Optimisation: BAAI/bge-m3

To address the non-English portions of the dataset, we fine-tuned the state-of-the-art multilingual encoder BGE-M3 (`BAAI/bge-m3`) [14], a 567M-parameter XLM-RoBERTa-based architecture. The model was fine-tuned for cross-lingual semantic grading using a Multiple Negatives Ranking (MNR) loss objective.

To maximise the efficiency of the MNR loss, we applied offline hard negative mining, extracting the top five hardest negative samples for each anchor-positive pair using the base model's initial embeddings. Training was conducted exclusively on the curated, strictly context-separated human gold-standard dataset in order to prevent context leakage and synthetic domain shifts.

Training was performed for two epochs using the AdamW optimiser with a learning rate of 2×10^{-5} and a cosine learning rate scheduler. To accommodate the model's large parameter count while maintaining a sufficiently large batch size for contrastive learning, we used gradient accumulation over 16 steps with a physical per-device batch size of 4, yielding an effective batch size of 64. The maximum sequence length was strictly capped at 512 tokens, and both attention and MLP dropout rates were kept at 0. The BAAI/bge-m3 model was fully fine-tuned, without layer freezing.

3. Results

3.1. Evaluation Metric: Argmax Cosine Similarity

Because our architecture relies on a metric-learning bi-encoder, the model does not output discrete classification logits. Instead, evaluation must be performed by analysing the geometric distances within the generated semantic space.

To evaluate the model's grading performance on an unseen dataset, we used an *Argmax Cosine Similarity* approach. For a given question, the model independently encodes the Full Credit (FC), Partial Credit (PC), and No Credit (NC) rubric texts to establish three semantic anchor vectors. The student's answer is then encoded to produce a corresponding answer vector.

The predicted grade $\hat{y}$ is determined by calculating the cosine similarity between the answer vector $\mathbf{v}_{\text{ans}}$ and the three rubric vectors $\mathbf{v}_{FC}$, $\mathbf{v}_{PC}$, and $\mathbf{v}_{NC}$, selecting the class with the maximum similarity:

$$\hat{y} = \underset{c \in \{FC, PC, NC\}}{\operatorname{argmax}} \cos(\mathbf{v}_{\text{ans}}, \mathbf{v}_c). \tag{3}$$

Our primary task metric, *Grading Accuracy*, is defined as the strict percentage of samples where the predicted grade $\hat{y}$ matches the human-annotated ground truth. Throughout the experiments, label 0 corresponds to No Credit (NC), label 1 to Partial Credit (PC), and label 2 to Full Credit (FC).

Before evaluating trained models, we computed a majority-class baseline from the training set. The train-set label distribution was 5,875 NC examples (34.6%), 5,596 PC examples (33.0%), and 5,509 FC examples (32.4%), making label 0 (NC) the most frequent class with a train-set majority accuracy of 0.3460. Applying this same majority-class predictor to the development set, whose label distribution was 613 NC examples (35.2%), 564 PC examples (32.4%), and 564 FC examples (32.4%), yielded a development accuracy of 0.3521 (35.21%).

As shown in Table 1, the majority-class baseline achieved 35.21% accuracy on the English development split, while the zero-shot baseline ModernBERT achieved an accuracy of 60.60%. Fine-tuning the English model with Multiple Negatives Ranking (MNR) loss and random batching (**v1**) substantially improved performance to 90.81%. Finally, combining the MNR loss with our Question-Aware Batching (QAB) strategy (**v2**) pushed the overall accuracy to its peak of 91.33%, representing an absolute improvement of 56.12 percentage points over the majority-class baseline and over 30 percentage points over the zero-shot baseline.

Regularisation proved highly influential in this configuration; removing the dropout parameters (**v3**) resulted in a 1.90 percentage point drop in accuracy (down to 89.43%), showing that the combination of QAB and dropout was beneficial for accuracy.

Additionally, although ModernBERT natively supports a context window of up to 8,192 tokens [12], we restrict the maximum sequence length to 512 to ensure efficient training and inference. Because the memory requirements of the standard self-attention mechanism scale quadratically with sequence length, incorporating the full attention window is highly susceptible to out-of-memory (OOM) errors, which would severely restrict the effective batch size and diminish overall training.

All ModernBERT-base configurations share the following training hyperparameters: 3 epochs, batch size of 16 with gradient accumulation over four steps (effective batch size 64), learning rate 2×10^{-5} with a cosine scheduler and warmup ratio 0.15, weight decay 0.05, maximum sequence length 512, and top-k hard negatives $= 5$, yielding 78,667 training triplets in total. Configurations **v1–v4** employ `MultipleNegativesRankingLoss` (scale 20), whereas **v5** and **v6** use Dynamic Margin Triplet Loss (DMT; margin 0.5). Dynamic Hard-Question Tracking is used in **v2**, **v3**, **v4**, and **v6**. The six variants differ in their loss function, batching strategy, regularisation, and layer freezing, as described below.

ModernBERT Variant Configurations

v1 Trained with MultipleNegativesRankingLoss, random batching, and attention/MLP dropout of 0.15. Serves as the fine-tuned baseline against which the effect of Question-Aware Batching is measured.

v2 Introduces Question-Aware Batching (QAB) with MultipleNegativesRankingLoss, dropout 0.15, and Dynamic Hard-Question Tracking. QAB constructs each batch to contain answers from multiple grade levels (FC, PC, NC) of the same question, while hard training questions are oversampled by a factor of 3. This forces the model to discriminate within-question grade boundaries at every update step. This configuration achieves the best overall performance.

v3 Identical to **v2**, including Dynamic Hard-Question Tracking, but with dropout set to zero (ablation). This variant isolates the contribution of dropout regularisation; the observed accuracy decrease of approximately 1.9 percentage points confirms that dropout is a meaningful component of the optimal configuration.

v4 Extends **v2**, including Dynamic Hard-Question Tracking, by freezing the bottom half of the transformer layers during training. Layer freezing was previously found to stabilise training for architecturally fragile models such as mDeBERTa-v3; here we test whether it benefits ModernBERT on the corrected, harder context-based split. Performance remains competitive but does not surpass **v2**, suggesting that full end-to-end fine-tuning is preferable for ModernBERT on this task.

v5 Evaluates DMT under a random batching strategy while keeping dropout at 0.15. This serves as the primary margin-based baseline.

v6 Combines DMT with Question-Aware Batching and Dynamic Hard-Question Tracking. Interestingly, incorporating QAB with the margin loss leads to a slight decrease in overall accuracy, suggesting potential interactions between the loss formulation and the structured batching constraint.

To assess the statistical robustness of the English development results, we estimated 95% confidence intervals (CIs) for grading accuracy using non-parametric bootstrapping. 1,000 bootstrap iterations were performed by randomly sampling the development set with replacement and recomputing accuracy for each resample. The resulting 95% accuracy CIs are as follows: **v1** [0.8937, 0.9213], **v2** [0.9001, 0.9265], **v3** [0.8788, 0.9092], **v4** [0.8960, 0.9236], **v5** [0.8817, 0.9087], and **v6** [0.8765, 0.9047]. These intervals are consistent with the overall ranking observed in Table 1, but indicate that the strongest ModernBERT variants are not clearly separated.

The routed development-set evaluation in Table 2 combines the fine-tuned ModernBERT English branch with the fine-tuned BAAI/bge-m3 multilingual branch. The combined system achieved 55.86% accuracy with a 95% confidence interval of [0.5432, 0.5740] and QWK of 0.4680. For comparison, Table 3

Table 1
Performance on the English development split. κ_{lin} and κ_{quad} denote linear and quadratic Cohen's kappa, respectively. Per-label accuracy is reported as NC/PC/FC. Bold marks the best ModernBERT configuration.

Model	Loss	Batching	Acc.	κ_{lin}	κ_{quad}	NC/PC/FC Acc.
Majority Class (NC)	–	–	35.21%	0.0000	0.0000	100.00/0.00/0.00
ModernBERT-base (Zero-shot)	–	–	60.60%	0.5176	0.6170	80.67/26.77/73.25
mDeBERTa-v3 (Zero-shot)	–	–	45.19%	–	–	–
ModernBERT-base (v1)	MNR	Random	90.81%	0.8800	0.9069	88.09/89.36/95.21
ModernBERT-base (v2)	**MNR**	**QAB**	**91.33%**	**0.8905**	0.9111	90.05/89.36/94.68
ModernBERT-base (v3)	MNR	QAB (No Dropout)	89.43%	0.8667	0.8919	88.58/86.52/93.26
ModernBERT-base (v4)	MNR	QAB (Frozen Layers)	90.98%	0.8888	**0.9128**	90.05/87.59/95.39
ModernBERT-base (v5)	DMT	Random	89.49%	0.8620	0.8815	89.72/86.17/92.55
ModernBERT-base (v6)	DMT	Question-Aware	89.09%	0.8592	0.8820	88.42/85.99/92.91

Table 2
Development-set performance of the routed system. QWK denotes quadratic weighted kappa. The English split reports only English samples, the non-English split reports all multilingual samples, and all-language rows report the combined routed evaluation. Per-label accuracy is reported as NC/PC/FC. The BAAI/bge-m3 baseline is detailed separately in Table 3.

Model	Split	Acc.	95% CI	QWK	NC/PC/FC Acc.	Samples
BAAI/bge-m3 (baseline routed)	Non-English	56.58%	[0.5497, 0.5805]	0.4115	44.86/34.68/77.96	3,802
ModernBERT-base (v2)	English	67.44%	[0.6250, 0.7238]	0.5774	72.04/59.18/69.93	344
BAAI/bge-m3 (fine-tuned)	Non-English	54.81%	[0.5334, 0.5623]	0.4582	61.45/39.93/60.10	3,802
Routed system	**All languages**	**55.86%**	**[0.5432, 0.5740]**	**0.4680**	**62.31/41.55/60.92**	**4,146**

reports the corresponding BAAI/bge-m3 baseline setup using the same English branch, while Table 4 provides the per-language breakdown of the fine-tuned routed pipeline. Language identifiers such as `en`, `cs`, and `ga` follow the ISO 639 language-code convention.[1]

Overall, the fine-tuned multilingual branch produced relatively balanced but modest non-English scores. Swedish (`sv`) and Irish Gaelic (`ga`) were among the lowest-performing non-English splits, with the latter being understandable given that Irish Gaelic is comparatively difficult for multilingual models from the perspective of orthography, diacritics, and tokenisation/font representation. At the same time, the gap between development and official test performance should be interpreted cautiously: it may reflect overfitting, the limited domain and size of the development data, or answer/rubric patterns that are relatively easy for the embedding model to learn. These results leave open an important direction for future work: further improving metric learning for multilingual grading and determining whether the current limitation comes primarily from the multilingual backbone, the training setup, or the metric-learning approach itself. This distinction is important because the same general approach appeared highly effective on English data, whereas its multilingual transfer remains substantially weaker.

3.2. The Impact of Context

Table 5 empirically supports our "Context Dilution" hypothesis. Including the full raw context resulted in the lowest accuracy. While summarised context provided a noticeable improvement, completely removing the context and forcing the model to rely on the semantic relationship among the question, answer, and rubric yielded the best performance and was also the fastest configuration to train. Because some context-based variants were evaluated on a different split, this comparison should be treated as indicative rather than a fully controlled ablation.

3.3. Ensemble Routing and Final Inference Pipeline

The final inference pipeline involves a model routing mechanism based on the incoming evaluation sample's language, which was designed to maximise performance across the entire multilingual CLEF benchmark. Evaluation samples were parsed for their language metadata tag.

[1] See https://en.wikipedia.org/wiki/List_of_ISO_639_language_codes.

Table 3
Per-language development-set performance of the BAAI/bge-m3 baseline routed setup. Both routed evaluations use the same English branch. QWK denotes quadratic weighted kappa. Per-label accuracy is reported as NC/PC/FC.

Language	Model Used	Acc.	95% CI	QWK	NC/PC/FC Acc.	Samples
en	EN Model	67.44%	[0.6250, 0.7238]	0.5774	72.04/59.18/69.93	344
cs	Multi Model	56.93%	[0.5181, 0.6205]	0.4078	43.01/34.78/79.59	332
da	Multi Model	57.32%	[0.5159, 0.6274]	0.4442	45.45/31.82/81.16	314
de	Multi Model	56.67%	[0.5100, 0.6233]	0.4183	48.19/32.94/77.27	300
el	Multi Model	56.29%	[0.5066, 0.6159]	0.4389	45.35/32.14/78.79	302
fi	Multi Model	57.62%	[0.5231, 0.6358]	0.4199	45.35/33.33/81.06	302
ga	Multi Model	52.32%	[0.4636, 0.5795]	0.3316	50.00/41.86/60.61	302
hu	Multi Model	56.87%	[0.5125, 0.6219]	0.3893	42.53/34.78/80.14	320
pt	Multi Model	55.83%	[0.5031, 0.6104]	0.4051	42.86/29.67/80.56	326
ro	Multi Model	56.44%	[0.5091, 0.6196]	0.4083	45.05/34.07/77.78	326
sr	Multi Model	56.33%	[0.5120, 0.6175]	0.4015	41.94/35.87/78.23	332
sv	Multi Model	57.06%	[0.5153, 0.6228]	0.4167	40.66/38.46/79.17	326
uk	Multi Model	59.06%	[0.5375, 0.6438]	0.4545	48.86/36.26/80.14	320
Overall	**Combined**	**57.48%**	**[0.5598, 0.5890]**	**0.4254**	**47.05/36.74/77.29**	**4,146**

Table 4
Per-language development-set performance of the fine-tuned routed inference pipeline. QWK denotes quadratic weighted kappa. Per-label accuracy is reported as NC/PC/FC.

Language	Model Used	Acc.	95% CI	QWK	NC/PC/FC Acc.	Samples
en	EN Model	67.44%	[0.6250, 0.7238]	0.5774	72.04/59.18/69.93	344
cs	Multi Model	55.12%	[0.4969, 0.6084]	0.4778	60.22/34.78/64.63	332
da	Multi Model	53.18%	[0.4777, 0.5828]	0.4488	57.95/35.23/61.59	314
de	Multi Model	56.00%	[0.5067, 0.6167]	0.5067	63.86/43.53/59.09	300
el	Multi Model	56.95%	[0.5166, 0.6225]	0.5291	65.12/39.29/62.88	302
fi	Multi Model	55.63%	[0.5033, 0.6126]	0.4342	56.98/48.81/59.09	302
ga	Multi Model	52.65%	[0.4735, 0.5828]	0.3926	57.14/38.37/59.09	302
hu	Multi Model	53.44%	[0.4813, 0.5875]	0.4268	63.22/36.96/58.16	320
pt	Multi Model	56.44%	[0.5122, 0.6166]	0.4982	65.93/38.46/61.81	326
ro	Multi Model	55.83%	[0.5061, 0.6104]	0.4699	64.84/39.56/60.42	326
sr	Multi Model	55.72%	[0.5030, 0.6114]	0.4503	60.22/41.30/61.90	332
sv	Multi Model	52.45%	[0.4755, 0.5768]	0.3992	60.44/37.36/56.94	326
uk	Multi Model	54.37%	[0.4875, 0.5969]	0.4684	61.36/46.15/55.32	320
Overall	**Combined**	**55.86%**	**[0.5432, 0.5740]**	**0.4680**	**62.31/41.55/60.92**	**4,146**

English queries are routed directly to the fine-tuned ModernBERT model. All non-English queries are routed to the fine-tuned BAAI/bge-m3 model. For both branches, the selected bi-encoder computes the normalised dense embeddings for the student's answer and the three language-matched rubric texts. The final prediction is assigned by taking the argmax of the cosine similarities. This decoupled routing strategy ensures that our highly optimised English predictions are not mathematically degraded by the compromises inherent to cross-lingual embedding spaces.

3.4. Final Test Set Results

Table 6 reports the official final test set results provided by the organisers in the final phase of the lab experiment. The submitted system followed the same general routing structure as our best development configuration, combining a ModernBERT-based English branch similar to **v2** with a BAAI/bge-m3

Table 5
Ablation study: Impact of context on embedding quality. Evaluated on the English development set only. Full Context and Summarised Context were evaluated on a different development split that may contain minor information leakage because the model had previously encountered similar contextual information through the rubrics; therefore, these values should be interpreted as indicative rather than as a fully controlled ablation. Later analysis showed that this had little influence on the reported accuracy.

Data Preprocessing Strategy	Accuracy
Full Raw Context Included	51.37%
LLM-Summarised Context	88.13%
No Context (Stripped)	**91.33%**

multilingual branch. However, the exact model weights differed from the development experiments reported above. The official test set scores are therefore reported separately and not directly compared since they were produced by different models.

The exact development-set accuracy of the submitted model pipeline on an unbiased split is not known. The previous development split used during the experiment was later found to be affected by context and rubric leakage, which inflated internal scores to approximately 93% accuracy for the English ModernBERT branch and 88% for the multilingual branch. The test-set scores available in Table 6 therefore represent the submitted, leakage-affected model pipeline. Although the models we consider to be our strongest final variants were not evaluated on the official test set, the official test set results in Table 6 remain the most reliable available estimate of generalisation performance for the submitted models.

Table 6
Final test set performance across evaluation splits. QWK denotes quadratic weighted kappa.

Evaluation Split	English Test Set		Multilingual Test Set	
	Accuracy	QWK	Accuracy	QWK
Overall (All Data, No Filters)	43.80%	0.2270	44.00%	0.2940
PISA Subset Only	35.80%	0.2340	35.90%	0.1850
Cleaned Rubrics (Type 1)	46.90%	0.2280	49.40%	0.3720
Adversarial Rubrics (Type 0)	27.90%	0.1810	26.00%	0.0270
Manual Answers	50.50%	0.2510	53.50%	0.4490
GPT-Generated Answers	35.80%	0.2340	35.90%	0.1850

Several patterns emerge from the official test scores. First, within the submitted model pipeline, the multilingual branch achieved a higher QWK (0.2940) than the English branch (0.2270) on the overall official evaluation, despite the weaker multilingual development results reported above. Second, adversarial rubrics substantially reduced performance, with accuracy dropping to 27.90% for English and 26.00% for multilingual data, indicating that the model remains sensitive to confusing or inconsistent rubric criteria. Finally, performance was higher on manually written answers than on GPT-generated answers, suggesting that synthetic answers may differ from human responses in ways that are not fully captured by the learned embedding space.

4. Discussion

The results validate our hypothesis that metric learning is an effective approach for automated short-answer grading. The strongest development performance was obtained by combining Multiple Negatives Ranking with Question-Aware Batching, suggesting that the model benefits from seeing semantically close answers from different grade levels within the same question. This setup forces the encoder to resolve fine-grained distinctions during training, rather than relying on easier negatives drawn from unrelated questions.

The experiments also highlight the sensitivity of metric-learning models to noisy or dominant input components. In spatial embedding models, token volume can strongly influence vector direction, which is consistent with our findings on context dilution. When a large share of the input consists of repeated

background context, the resulting representations may drift towards the shared context signal rather than the grading-relevant information. Removing context therefore allowed the model to focus more directly on the semantic relationship between the answer and rubric.

The multilingual results show that applying the same metric-learning idea to a multilingual backbone still leaves substantial room for improvement. In particular, fine-tuning BAAI/bge-m3 with the current objective did not appear to consistently nudge the multilingual embedding space in the desired direction. This suggests that future work should investigate whether the limitation lies in the multilingual model itself, the training data and sampling strategy, or the way the metric-learning objective transfers across languages.

At the same time, the final official test results were substantially lower than the internal development scores, suggesting that the development setup overestimated generalisation. The sharp drop on adversarial rubrics indicates that the current approach depends strongly on clean and logically consistent rubric formatting. Further analysis of why the development set produced such high accuracy, and why this performance dropped on the official test set, is left for future work.

Data and Code Availability

The SenseMaking dataset was provided by the CLEF 2026 organizers. The complete source code for our training pipeline, including both failed and successful experiments, is available on GitHub: https://github.com/inferno73/sensemaking-metric_learning_approach.

Acknowledgments

We would like to thank our colleagues and the organizers of the CLEF 2026 conference for providing the dataset, guidance, and advice during the task experiments.

Declaration on Generative AI

During the preparation of this work, the authors used the following generative AI tools: **Google Gemini** (via the Antigravity coding assistant), **OpenAI ChatGPT** (LaTeX editor Prism), and **Anthropic Claude Sonnet**. These tools were employed for the following purposes, in accordance with the CEUR-WS GenAI Usage Taxonomy:

- **Drafting content.** AI assistance was used in drafting and structuring portions of the methodology and experimental sections.
- **Paraphrase and reword.** AI tools were used to rephrase and clarify selected sentences to improve precision and readability.
- **Improve writing style.** Suggestions for sentence structure, word choice, and overall flow were obtained from AI assistants during the revision of the manuscript.
- **Grammar and spelling check.** AI tools were used to identify and correct grammatical and typographical errors throughout the manuscript.

All AI-generated content was reviewed, verified, and edited by the authors. The authors take full responsibility for the content of the publication.

References

[1] A. Muise, A. Yang, Metric Learning, Springer Nature, 2022.

[2] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems 32 (NeurIPS), 2019, pp. 8024–8035.

[3] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2019.

[4] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.

[5] Q. Lhoest, A. V. del Moral, Y. Jernite, A. Thakur, P. von Platen, S. Patil, J. Chaumond, M. Drame, J. Plu, L. Tunstall, J. Davison, M. Sasko, G. Chhablani, B. Malik, S. Brandeis, T. L. Scao, V. Sanh, C. Xu, N. Patry, A. McMillan-Major, P. Schmid, S. Gugger, C. Delangue, T. Matussiere, L. Debut, S. Bekman, P. Cistac, T. Goehringer, V. Mustar, F. Lagunas, A. M. Rush, T. Wolf, Datasets: A community library for natural language processing, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2021, pp. 175–184.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[7] A. Sinha, O. Umakanthan, S. Gajendran, Dr-cot: dynamic recursive chain of thought with meta reasoning for parameter efficient models, Scientific Reports 15 (2025). doi:10.1038/s41598-025-18622-6.

[8] M. Doshi, A. Trivedi, V. Kumar, P. Awasthy, Y. Li, J. Sen, R. Florian, S. Joshi, Lmk > cls: Landmark pooling for dense embeddings (2026). doi:10.48550/arXiv.2601.21525.

[9] Microsoft, Phi-3.5-mini-instruct, Hugging Face model card, 2024. URL: https://huggingface.co/microsoft/Phi-3.5-mini-instruct.

[10] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, D. Valter, S. Narang, G. Mishra, A. W. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei, Scaling instruction-finetuned language models, 2022. URL: https://arxiv.org/abs/2210.11416. arXiv:2210.11416.

[11] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: A unified embedding for face recognition and clustering , in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2015, pp. 815–823. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298682. doi:10.1109/CVPR.2015.7298682.

[12] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, N. Cooper, G. Adams, J. Howard, I. Poli, Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL: https://arxiv.org/abs/2412.13663. arXiv:2412.13663.

[13] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, in: International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=XPZIaotutsD.

[14] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 2318–2335. URL: https://aclanthology.org/2024.findings-acl.137/. doi:10.18653/v1/2024.findings-acl.137.