Failure at BEA 2026 Shared Task 1: One Pipeline, Three L1s: A Unified Language-Agnostic System for Vocabulary Difficulty Prediction

Abid Al Hossain
Chittagong University of Engineering and Technology
Chattogram, Bangladesh
aabid.al.hossain@gmail.com

Kamruzzaman Khan Alve
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
kamruzzamanalve@gmail.com

Abstract

We present team Failure's submission to the BEA 2026 Shared Task on L1-aware vocabulary difficulty prediction for English learners (Felice and Skidmore, 2026). We investigate whether a single, language-agnostic pipeline can achieve competitive performance across typologically distinct L1 settings without per-language manual engineering. We implement a unified notebook, model architecture, and training procedure for Spanish, German, and Mandarin Chinese. Our system fine-tunes `xlm-roberta-base` (Conneau et al., 2020) with attention-mask-weighted mean pooling, augmented with four universal length-based features. We use Optuna-driven hyperparameter search (Akiba et al., 2019) and a leakage-free 5-fold ensemble with fold-specific standardization. Under this unified and compute-constrained setup, our closed-track system outperforms the official baseline on all three evaluation languages: RMSE 1.219 vs. 1.257 for Spanish, 1.118 vs. 1.258 for German, and 1.090 vs. 1.140 for Chinese. These results suggest that a compact, reproducible pipeline can yield consistent improvements without L1-specific customization, providing a practical baseline for multilingual educational NLP.

1 Introduction

Vocabulary difficulty prediction is a canonical educational NLP task, with applications in adaptive assessment, content selection, and learning models. The BEA 2026 Shared Task (Felice and Skidmore, 2026) poses the task as a regression problem: given an English vocabulary test item and its corresponding first-language (L1) context, the objective is to predict the psychometrically calibrated GLMM difficulty score for learners from one of three L1 backgrounds: Spanish, German, and Mandarin Chinese.

One might imagine solving the task in the most intuitive way, by training specialized models for each L1 condition, which requires different feature engineering or heuristics tailored to the language at hand. However, we adopt a language-agnostic stance. The motivation for this work is grounded in the assumption that there is sufficient underlying structure in the task across L1 conditions to allow a generalized approach to perform better than the official baseline on all three languages, even in a setup where all models are independently tuned with identical architecture, features, hyperparameters, and training methodology.

There is more to this line of reasoning than maximizing leaderboard scores. A language-agnostic pipeline is more easily audited and reproduced than a language-specific one. It also is more easily extended to new L1 conditions should the community choose to do so in future tasks. A model that can run on freely available hardware is more widely applicable and usable than one requiring institutional compute. Finally, an approach that relies on consistent modeling assumptions rather than language-specific heuristics is a more scientifically interesting contribution.

In this closed-track paper, we show that this language-agnostic pipeline achieves superior performance on all three languages, as predicted. This contribution goes beyond just improving the score, as it provides a complete pipeline that is reproducible in practice and runnable on free GPU hardware.

2 Task and Data

The BEA 2026 Shared Task (Felice and Skidmore, 2026) provides parallel data for three learner L1 conditions: Spanish (`es`), German (`de`), and Mandarin Chinese (`cn`). The target variable is `GLMM_score`, a psychometric difficulty estimate derived from more than 100,000 learner responses using random-item–random-person Rasch models (Schmitt et al., 2024).

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 1041–1046

The dataset is based on the British Council's Knowledge-based Vocabulary Lists (KVL) (Schmitt et al., 2021) and was adapted into the Extended KVL Dataset for NLP (Skidmore et al., 2025). Each item includes the fields `item_id`, `L1`, `en_target_word`, `en_target_pos`, `en_target_clue`, `L1_source_word`, and `L1_context`; labeled splits additionally include `GLMM_score` as the target variable. Items with the same `item_id` across different L1 files are parallel, meaning they correspond to the same English target word but include L1-specific prompts and difficulty annotations.

Each L1 condition contains 6,091 training items, 677 development items, and 748 test items. We train one model per L1 condition using only the corresponding training data. The closed track does not permit combining data across L1s, using external training data, or relying on large language models. Our system fully complies with these constraints.

3 System Overview

Our system is implemented as a single Kaggle notebook and applied to all three L1 conditions. The only run-specific setting is the target L1 identifier (`es`, `de`, or `cn`); no language-specific preprocessing, feature engineering, or post-processing is used.

The pipeline has three stages controlled by a single `MODE` flag: hyperparameter search, 5-fold training with development evaluation, and test inference from saved fold checkpoints. The same codebase is reused across all languages and stages.

Figure 1 summarizes the overall workflow and the shared per-fold model architecture. In `MODE='optuna_tune'`, Optuna searches over the main training hyperparameters using reduced cross-validation. In `MODE='train_dev'`, the selected hyperparameters are used for full 5-fold training with fold-specific standardization, and the best checkpoint is saved for each fold. In `MODE='test_infer'`, the saved checkpoints are loaded, predictions are generated independently per fold, and the final output is obtained by averaging fold predictions and inverse-transforming them to the original `GLMM_score` scale.

4 Input Representation

Text serialization. For each item, we serialize all available text fields into a single input string using bracket markers:

```
[L1] es [WORD] universal [POS]
adjective [CLUE] u_______ [SOURCE]
global [CONTEXT] La escalada. . .
```

This jointly exposes lexical, syntactic, orthographic, translational, and contextual information to the encoder. The serialization format is identical for all three languages. Text fields are cleaned by collapsing whitespace and replacing null values with empty strings, ensuring the tokenizer never receives null input.

Numerical features. We extract four character-count features from each item: **`word_len`**, **`clue_len`**, **`source_len`**, and **`context_len`**, computed as the character length of `en_target_word`, `en_target_clue`, `L1_source_word`, and `L1_context` respectively. These are universally available across all L1s and require no language-specific processing. Length-based signals are present in the task format itself, particularly through the partial-spelling clue, and can provide lightweight complementary information to the encoder.

5 Model Architecture

We use `xlm-roberta-base` (Conneau et al., 2020) as the encoder backbone. The serialized input is tokenized with a maximum length of 128 tokens and passed through the 12-layer transformer. We apply attention-mask-weighted mean pooling over the last hidden states:

$$\mathbf{h} = \frac{\sum_i m_i \cdot \mathbf{e}_i}{\sum_i m_i} \in \mathbb{R}^{768}$$

where $\mathbf{e}_i$ is the i-th token representation and $m_i \in \{0, 1\}$ is the attention mask value, so that padding tokens do not contribute to the pooled representation.

A dropout layer ($p = 0.1$) is applied to $\mathbf{h}$. The four numerical features are independently Z-score standardized using fold-specific training statistics to form $\mathbf{f} \in \mathbb{R}^4$. We concatenate them:

$$\mathbf{z} = [\mathbf{h}; \mathbf{f}] \in \mathbb{R}^{772}$$

A linear regression head produces the predicted difficulty: $\hat{y} = \mathbf{W}\mathbf{z} + b$. Training is performed in standardized target space using mean squared error (MSE) loss, and predictions are inverse-transformed to the original GLMM scale at evaluation time:

$$\hat{y}_{\text{raw}} = \hat{y} \cdot \sigma_{\text{fold}} + \mu_{\text{fold}}$$

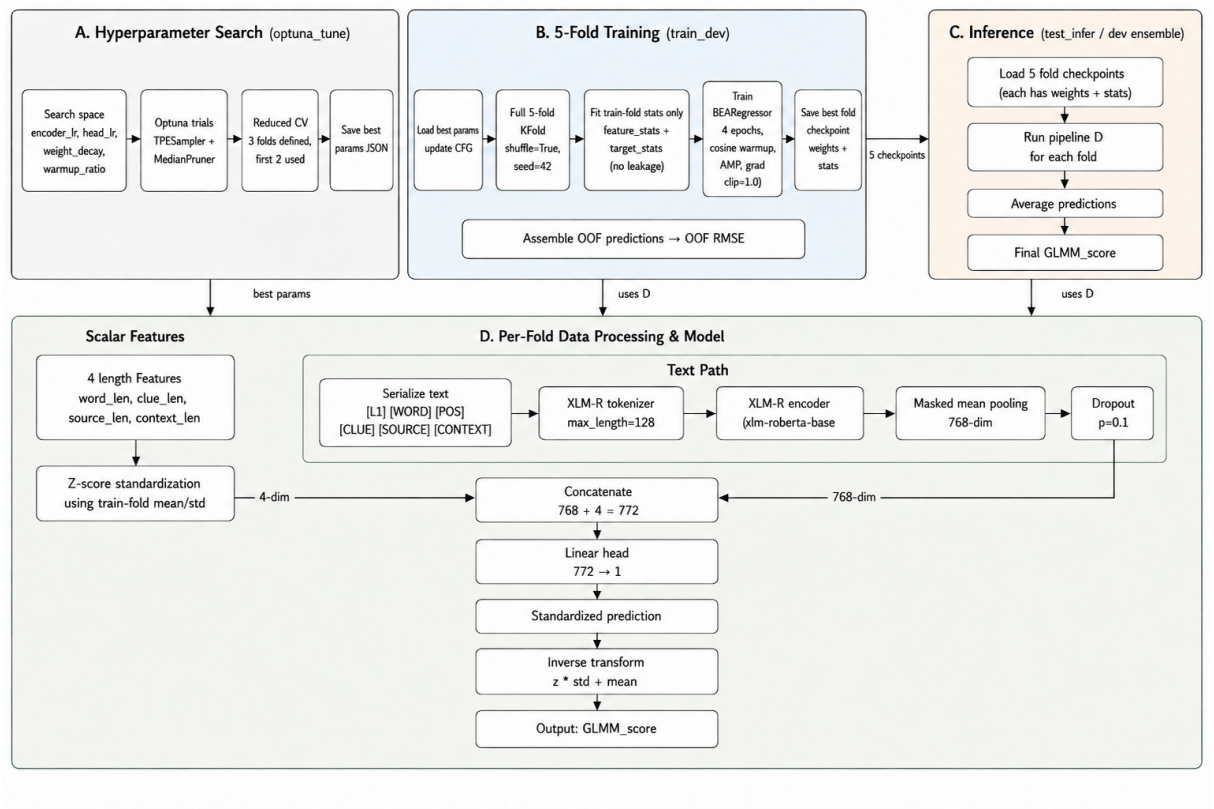


Figure 1: Overview of our unified pipeline for one target L1. The upper panels show the three operating modes, and the lower panel shows the shared per-fold model architecture.

where μ_{fold} and σ_{fold} are the target mean and standard deviation computed on that fold's training partition.

We selected `xlm-roberta-base` as the largest multilingual encoder that fit within the memory budget of our two available NVIDIA T4 GPUs.

6 Training Procedure

Hardware and runtime. All experiments were conducted on two NVIDIA T4 GPUs (16 GB each) available through the Kaggle free compute tier. Multi-GPU training was handled automatically via `nn.DataParallel`. A complete run for one L1 condition—including Optuna search, 5-fold training, development evaluation, and test inference—required approximately 2 hours of wall-clock time, with some variation depending on GPU availability.

Hyperparameter optimization. We use Optuna (Akiba et al., 2019) with a TPE sampler (`TPESampler`, seed 42) and a Median pruner (`MedianPruner`, `n_startup_trials=5`, `n_warmup_steps=1`) to optimize four hyperparameters: encoder learning rate in $[5\times10^{-6}, 5\times10^{-5}]$ (log scale), head learning rate in $[3\times10^{-5}, 5\times10^{-4}]$ (log scale), weight decay in $[10^{-5}, 10^{-1}]$ (log scale), and warmup ratio in $[0.01, 0.20]$. Each of 15 trials is evaluated using reduced 3-fold cross-validation (`KFold`, `shuffle=True`, seed 42), but only the first 2 folds are scored per trial for efficiency (`OPTUNA_N_SPLITS=3`, `OPTUNA_MAX_FOLDS=2`). The objective is the mean validation RMSE across the evaluated folds. Best parameters are saved to a JSON file and automatically loaded before full training, eliminating manual transcription errors.

Leakage-free 5-fold training. For full training, we use 5-fold cross-validation (`KFold`, `n_splits=5`, `shuffle=True`, `random_state=42`). For each fold, both feature standardization statistics and target standardization statistics are computed only on the fold's training partition and stored inside the saved checkpoint. These statistics are reused at inference time, preventing any information flow from held-out data. A zero-variance guard ($\sigma < 10^{-8}$) is applied to avoid division errors in degenerate cases.

Optimization. The model is trained with MSE loss in standardized target space. We use AdamW with three parameter groups: encoder parame-

ters with the tuned weight decay; LayerNorm and bias parameters without weight decay; and the regression head with its own tuned learning rate and no weight decay. We apply a cosine learning rate schedule with linear warmup (`get_cosine_schedule_with_warmup`), train for 4 epochs, and clip gradients to a maximum norm of 1.0. Mixed precision training (AMP) is enabled throughout via `torch.cuda.amp.autocast` and `GradScaler`. Seeds are fixed (`seed=42`) for approximate reproducibility; cuDNN benchmark mode is enabled for throughput, so results may vary slightly across hardware.

Memory-efficient implementation. To operate within the T4 memory budget, we use dynamic per-batch padding so sequences are padded only to the longest item in each batch. Additional choices include `pin_memory=True` in all DataLoaders, `set_to_none=True` in gradient zeroing, explicit GPU cache clearing and Python garbage collection between folds and Optuna trials, and `gc_after_trial=True` in the Optuna study to prevent memory accumulation across trials.

Checkpoint selection and ensemble inference. Within each fold, the checkpoint achieving the lowest validation RMSE across all epochs is saved, rather than the final-epoch state. For each L1, all 5 fold checkpoints are loaded sequentially and their predictions are averaged. Because each checkpoint stores its own feature and target normalization statistics, inference is fully self-contained: no external normalization state is needed beyond what is saved in the checkpoint file.

Fixed settings. Training batch size 32, validation batch size 64, maximum sequence length 128, dropout 0.1, seed 42. All non-tuned settings are identical across Spanish, German, and Mandarin Chinese.

7 Results and Discussion

7.1 Main Results

Table 1 presents our official closed-track test results alongside the official baseline.

Our system improves over the official baseline in RMSE for all three L1 conditions. The same pipeline is applied across all three languages without any language-specific adaptation. The largest absolute improvement is observed for German (0.140 RMSE, 11.1%), followed by Chinese (0.050, 4.4%) and Spanish (0.038, 3.0%). These results show that a shared architecture and training recipe can produce consistent gains across all three L1 settings without any language-specific engineering.

L1	Ours	Baseline	ΔRMSE	Δ%
Spanish	1.219	1.257	−0.038	−3.0%
German	1.118	1.258	−0.140	−11.1%
Chinese	1.090	1.140	−0.050	−4.4%

Table 1: Official closed-track RMSE on the test set. Lower is better.

L1	F1	F2	F3	F4	F5	Mean	Std
Spanish	1.270	1.231	1.269	1.335	1.279	1.277	0.036
German	1.161	1.189	1.168	1.198	1.231	1.189	0.025
Chinese	1.105	1.134	1.106	1.139	1.109	1.118	0.015

Table 2: Per-fold validation RMSE across 5 folds for each L1 condition.

7.2 Validation Stability

Table 2 reports per-fold validation RMSE across all five folds for each L1 condition. Standard deviation remains low for all three languages (0.015–0.036), indicating stable training and suggesting that no single fold dominates the final ensemble prediction. Chinese is the most stable condition, while Spanish shows the largest fold-to-fold variation. Fold 4 yields the highest validation RMSE for both Spanish and German, but the overall spread remains limited, supporting the robustness of the cross-validation procedure.

7.3 Error Analysis

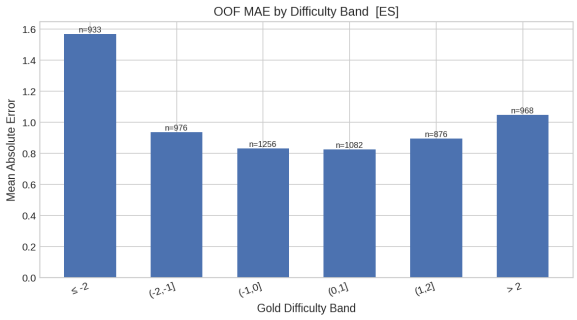
Figure 2 shows out-of-fold mean absolute error (MAE) by gold difficulty band for all three L1 conditions.

Across all three languages, error is highest in the easiest band (≤ -2), at approximately 1.57 for Spanish, 1.59 for German, and 1.52 for Chinese. In the middle bands (-1 to $+1$), MAE is substantially lower. Error increases again at the hardest end (> 2), especially for Spanish.

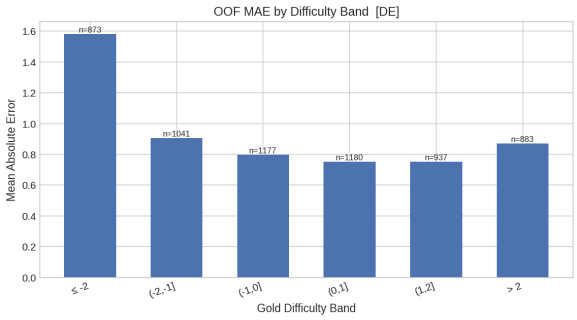
The same pattern appears across all three L1 conditions, suggesting that the model is less accurate at the extremes of the difficulty range and tends to pull very easy items toward the center of the score distribution.

7.4 Discussion

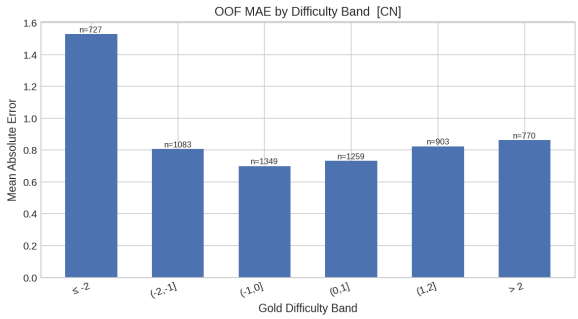
Language-agnostic generalization. Our system improves on the official baseline for all three L1



(a) Spanish



(b) German



(c) Chinese

Figure 2: Out-of-fold mean absolute error by gold difficulty band for the three L1 conditions.

conditions without any language-specific tuning. This suggests that the task contains enough shared structure for a single pipeline design to work reliably across Spanish, German, and Chinese. The combination of the target word, spelling clue, L1 translation, and L1 context appears to provide sufficient signal for a multilingual encoder to learn useful patterns of vocabulary difficulty.

Why German benefits most. The largest improvement is observed for German. One possible explanation is that the German subset aligns especially well with the multilingual encoder and the added scalar features. However, this remains a hypothesis rather than a firm conclusion, since verifying it would require a more detailed per-language analysis.

Difficulty extremes are hardest. Figure 2 shows that prediction error is highest for the easiest items and rises again, though less sharply, for the hardest ones. Because the same pattern appears across all three L1 conditions, it is more likely to reflect a general regression-to-the-mean effect than a language-specific issue. This suggests that future work should focus more directly on improving calibration at the tails of the score distribution.

Gap to top-ranked systems. Although our system outperforms the official baseline, it still remains below the top closed-track submissions. The most likely reason is limited model scale: we rely on a single base-sized encoder and a modest 5-fold ensemble under a restricted Kaggle compute budget. Larger multilingual encoders and stronger ensembles would likely improve performance further while still remaining within the closed-track rules.

Reproducibility. A practical strength of the system is that the same notebook is used for all three languages, with only the L1 identifier changed between runs. In addition, each checkpoint stores its own normalization statistics together with the model weights, which makes inference reproducible directly from the saved checkpoints. This keeps the pipeline simple to rerun, inspect, and extend.

8 Conclusion

We presented a unified, language-agnostic closed-track system for the BEA 2026 Shared Task on vocabulary difficulty prediction. Using one notebook, one model, one feature set, and one training recipe across Spanish, German, and Mandarin Chinese,

our system consistently outperforms the official baseline on all three L1 conditions. Error analysis reveals a consistent pattern across languages: prediction error is highest for the easiest vocabulary items, suggesting a systematic regression-to-the-mean tendency that future work could address. We hope this pipeline serves as a practical and accessible starting point for future work on L1-aware vocabulary modeling.

Limitations

Our system does not achieve top-ranked performance. A main reason is compute: under a budget of two NVIDIA T4 GPUs on the Kaggle free tier, we used `xlm-roberta-base` and a 5-fold ensemble of the same architecture rather than a larger encoder or a multi-encoder ensemble. Such extensions remain compatible with the closed-track rules but were beyond our available resources.

Our feature set is also intentionally minimal. We use only four character-count features, which capture simple surface-level signals but omit potentially useful information such as word frequency, morphological complexity, subword count, and cognate similarity. These additions would still be closed-track compliant and are a natural direction for future work.

Although the pipeline is language-agnostic in design, it has been evaluated only on the three L1 conditions provided by the shared task. Generalization to other L1s therefore remains untested. Finally, our test predictions are based on fold models trained on the training split only; retraining on the combined training and development data might yield further gains. Because cuDNN benchmark mode is enabled for speed, small run-to-run variation across GPU setups is also possible.

Ethical Considerations

Vocabulary difficulty prediction has direct applications in adaptive testing and personalized content selection. However, prediction errors can negatively affect learner experience, particularly for learners at the easiest end of the difficulty distribution, where our analysis shows error is consistently highest across all three L1 conditions (Figure 2). We used only the officially released shared-task data and comply with all closed-track constraints. No personally identifiable learner information is present in the data. As with any automated scoring system, outputs should inform rather than replace human pedagogical judgment, especially in high-stakes assessment contexts.

Acknowledgements

We used ChatGPT (OpenAI) for code assistance and Claude (Anthropic) for manuscript preparation. No generative AI model was used as part of the submitted closed-track system, data, or prediction pipeline. All results are from the official shared-task leaderboard and were verified by the authors, who take full responsibility for this submission.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Mariano Felice and Lucy Skidmore. 2026. BEA 2026 shared task: Vocabulary difficulty prediction for English learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Norbert Schmitt, Karen Dunn, Barry O'Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. Introducing knowledge-based vocabulary lists (KVL). *TESOL Journal*, 12(4).

Norbert Schmitt, Karen Dunn, Barry O'Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. *Knowledge-based Vocabulary Lists*. British Council.

Lucy Skidmore, Mariano Felice, and Karen Dunn. 2025. Transformer architectures for vocabulary test item difficulty prediction. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 160–174. Association for Computational Linguistics.