SurreyCTS at BEA 2026 Shared Task 1: Semantic Funnelling and Entropy-based Multilingual Lexical Difficulty Prediction

Georgina Willoughby[1*] **Jordan Painter**[1*] **Diptesh Kanojia**[2]
Emily Wells[1] **Constantin Orăsan**[1]
[1]Centre for Translation Studies and [2]Institute for People-Centred AI
University of Surrey, UK
{gw00697, j.painter, d.kanojia, ew00880, c.orasan}@surrey.ac.uk

Abstract

This paper describes the SurreyCTS submission[1] to the BEA 2026 shared task on lexical difficulty prediction, entered in the Open Track. Our approach progressed from baseline multilingual encoders to a hybrid RemBERT architecture with extensive feature engineering, combining semantic funnelling, lexical similarity features, attention-derived signals, and language-aware representations. On our internal production validation set, the best single models achieved RMSE 0.8122 (prod-H) and Pearson correlation 0.8968 (prod-G). A weighted ensemble of the five strongest systems, with weights proportional to inverse squared validation RMSE, was submitted as our final entry. On the official shared-task test set, the ensemble achieved RMSE 1.034, 0.945, and 0.861 for Spanish, German, and Chinese respectively, outperforming the open-track baseline in all three settings and placing fifth among open-track teams.

1 Introduction

Lexical difficulty prediction aims to estimate how difficult a target word is for a language learner to understand in context. Difficulty is influenced by multiple factors, including morphological complexity, contextual informativeness, and cross-lingual similarity between the learner's first language and the target language. In multilingual settings, these factors may vary substantially across learner populations, making the task particularly challenging. Lexical complexity prediction has been explored as a shared task previously, notably at SemEval-2021 Task 1 (Shardlow et al., 2021), which focused on single- and multi-word expression complexity for English.

The BEA 2026 shared task provides a structured setting for developing and evaluating such systems, requiring participants to predict continuous difficulty scores for English vocabulary items across three typologically distinct learner populations: Spanish, German, and Chinese. These scores are derived from Generalised Linear Mixed Models (GLMMs), which estimate item difficulty as a continuous latent parameter by modelling learner responses while accounting for random variation across both items and individuals (De Boeck, 2008; Dunn, 2024). The scores are fitted to large-scale psychometric data from over 100,000 learners, making the regression target both fine-grained and linguistically motivated.

In this paper, we describe our system for lexical difficulty prediction across three learner settings: Spanish, German, and Chinese. Our experiments progressed through four stages: baseline multilingual encoders, hybrid feature engineering, split redesign, and final ensembling. Across these stages, the main pattern was that carefully designed linguistic and cross-lingual features consistently improved over encoder-only regression baselines.

The contributions of this paper are as follows:

- We propose a hybrid multilingual regression architecture for lexical difficulty prediction that combines contextual representations with linguistic and cross-lingual features.
- We provide an ablation study showing that selective feature additions consistently outperform broad feature bundles, and that the same feature strategy does not transfer equally across encoder backbones.
- We demonstrate that careful validation set construction has a meaningful effect on model performance in this setting.

2 Task and Data

The BEA 2026 shared task focuses on predicting the difficulty of English vocabulary items for learners from three L1 backgrounds: Spanish, German,

[*]Equal contribution.
[1]Code and configuration files are available at https://github.com/surrey-nlp/bea2026-surrey.

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 1016–1023

and Chinese, formulated as a regression over continuous GLMM difficulty scores. We describe the data below.

The data comes from the Extended KVL Dataset for NLP (Skidmore et al., 2025), which is derived from the British Council's Knowledge-based Vocabulary Lists (Schmitt et al., 2021, 2024). The difficulty scores were estimated from approximately 3.3 million learner responses collected from over 100,000 learners in a translation-based recall setting. These estimates were derived separately for each L1 using random-item-random-person Rasch models within a GLMM framework (De Boeck, 2008; Dunn, 2024).

Each instance in the dataset contains five core input fields: an $L1$ context sentence, an $L1$ source word, a partial-spelling clue for the English target word, a part-of-speech tag, and the English target word itself. The target variable is the corresponding GLMM score. The dataset is split into training (18,273 instances), development (2,031), and test (2,244) sets, with each instance appearing once per $L1$.

The task poses several challenges. The target scores are continuous values derived from psychometric modelling, making the regression signal fine-grained. The three $L1$s also span typologically distinct languages and different writing systems, which requires a system capable of handling varied multilingual input within a unified framework.

3 Method

3.1 Baseline Encoders

We fine-tuned and evaluated baselines which used XLM-R base and large (Conneau et al., 2020), mDeBERTa (He et al., 2021), COMET (Rei et al., 2020), and RemBERT (Chung et al., 2021) as multilingual encoder backbones. Early experiments indicated that COMET[2] was a competitive baseline, while RemBERT with decoupled embeddings became the main focus for our experiments given its performance with the hybrid feature setup.

3.2 Hybrid Regression Model

The proposed architecture in Figure 1 extends the encoder-only setup with a Deep MLP-based hybrid regression head. The regression head is a 3-layer MLP ($1024 \rightarrow 512 \rightarrow 128 \rightarrow 1$) with LayerNorm, GELU activations, and decreasing dropout

[2]We use the `Unbabel/wmt22-comet-da` checkpoint, which uses an XLM-R encoder fine-tuned for MT quality estimation (Rei et al., 2022).

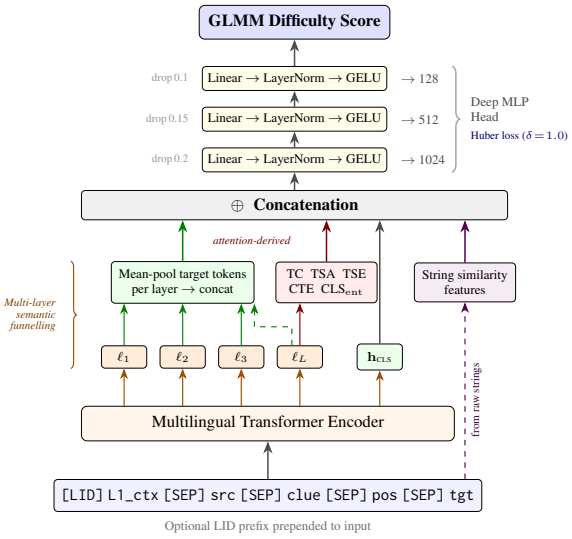

Figure 1: Architecture of the `SurreyCTS` system. The input sequence passes through a multilingual transformer encoder, producing multi-layer representations that are funnelled via mean-pooling, combined with attention-derived entropy features and string similarity features, and fed into a deep MLP regression head to predict GLMM difficulty scores.

$(0.2, 0.15, 0.1)$, trained with Huber loss ($\delta = 1.0$) (Huber, 1964). It extracts target-word representations from the encoder, combines them with engineered features, and predicts lexical difficulty through this head. This design allows the model to combine contextual representations with explicit lexical and cross-lingual signals.

We also experimented with multi-layer semantic funnelling: rather than pooling target-word representations from a single transformer layer, we mean-pool target-token hidden states at several selected layers and concatenate them before the regression head. This gives the model access to multiple levels of abstraction simultaneously, from lower-level morphosyntactic features to higher-level contextual ones. The selected layers (RemBERT: 12, 20, 26, 32; COMET: 8, 10, 12) sample across each encoder's depth, motivated by observations that different transformer layers encode distinct linguistic levels (Tenney et al., 2019; Rogers et al., 2020). Funnelling was an important component of our strongest RemBERT-based configurations.

3.3 Feature Engineering

Beyond the base encoder, we experimented with several families of additional features: **lexical similarity** (Levenshtein distance, normalized Levenshtein, Jaccard distance); **subword-based** (subword ratio); **language-aware** (language ID prefix,

language ID embeddings, pinyin-based Chinese similarity); **attention-derived** (target-to-context entropy (TC), target-to-source alignment (TSA) and entropy, context-to-target entropy, global CLS entropy, attention variance); and **representation-level** (multi-layer funnelling, target-word pooling across selected layers). Detailed descriptions of these features are provided in Appendix A.

Of these, the final ensemble systems share a base feature set of normalised Levenshtein distance, Jaccard distance, pinyin romanisation, and subword ratio. Subword ratio uses the backbone tokeniser's segmentation rather than linguistic syllabification, providing a model-internal proxy for morphological complexity that requires no external resource. Individual ensemble members augment this base with multi-layer funnelling, language-ID prepending, and selected attention-derived signals such as TSA and TC entropy (see Table 1 for per-system feature recipes). Remaining features (full entropy bundles, language-ID embeddings, attention variance, target-word pooling) were explored during model development but did not improve over the curated subsets. Target length and clue-mask ratio were examined in correlation analysis only and were not included as model features.

Table 1 reports all ten production candidates on the internal validation set, ordered by RMSE. The leaderboard supports both feature comparison (RemBERT funnelling variants vs. COMET, full vs. selective features) and ablation reading: `prod-D` (vanilla RemBERT) $\rightarrow$ `prod-A` ($+$ funnelling) $\rightarrow$ `prod-F` ($+$ TSA) $\rightarrow$ `prod-H` ($+$ LID prepend) traces the contribution of each major addition.

4 Experimental Setup

4.1 Training and Model Selection

Early experiments were conducted on the standard labelled train and development portions of the shared-task data. For the final production stage, we used an internal split with a 500-instance development set, which allowed more labelled data to be used for training while preserving a dedicated validation set for checkpoint selection. Development RMSE was used as the primary model-selection criterion, with Pearson correlation used as a secondary metric. The official development set (2,031 unstratified instances) failed to distinguish configurations that diverged clearly on our production validation split, making it unreliable for checkpoint selection.

Encoder-only experiments plateaued around RMSE 1.00 regardless of feature additions, motivating a redesign of the validation split. We divided the labelled data into 50 equal-frequency GLMM score quantile buckets and selected the 10 highest-scoring instances from each bucket, yielding a 500-instance development set biased toward harder examples. The remaining 19,804 instances were used for training. The 50-bucket choice balanced fine-grained difficulty stratification with sufficient instances per bucket to support stable selection. This design served two purposes: maximising the amount of labelled data available for training, and obtaining a validation signal sensitive to performance on difficult items, which are most discriminative for model selection. Because this split combines instances across all three $L1$s into a single training set, which is not permitted under closed-track rules, our submission was entered in the open track.

4.2 Production Models and Ensemble

Following the split redesign, a sweep of ten models spanning RemBERT and COMET backbones, alternative loss functions (Huber $\delta = 1.0$ and $\delta = 0.5$), and different feature subsets produced a best RMSE of 0.8122, a substantial improvement over the official-split plateau. These models were chosen to balance strong standalone performance with variation across backbone family, feature recipe, and loss function, while avoiding a larger search likely to add little beyond the strongest configurations.

Our final submission ensemble was built from the five best validation systems: `prod-H`, `prod-G`, `prod-J`, `prod-E`, and `prod-F`. Ensemble weights were set as $w_i \propto 1/\mathrm{RMSE}_i^2$, normalised to sum to one, so that stronger individual models contributed more while still preserving diversity across backbone and feature choices. This design was motivated by the complementary strengths of the selected systems: `prod-H` gave the best RMSE, `prod-G` gave the strongest correlation metrics, and the remaining RemBERT variants contributed alternative feature combinations and training dynamics.

5 Results and Analysis

5.1 Main Results

Our production sweep establishes a clear hierarchy among the final systems. `prod-H` achieves the lowest RMSE (0.8122), `prod-G` achieves the highest Pearson and Spearman correlations (0.8968 and 0.8953), and `prod-J` provides the best bal-

System	Backbone	Feature recipe	RMSE	Pearson	Spearman	MAE
prod-H	RemBERT	funnelling + LID prepend	0.8122	0.8920	0.8910	0.6503
prod-G	COMET	Minimal	0.8127	0.8968	0.8953	0.6339
prod-J	RemBERT	funnelling + top features	0.8129	0.8923	0.8908	0.6522
prod-E	RemBERT	funnelling + Huber $\delta = 0.5$	0.8194	0.8898	0.8838	0.6545
prod-F	RemBERT	funnelling + TSA	0.8239	0.8887	0.8862	0.6604
prod-C	RemBERT	funnelling + full entropy	0.8244	0.8885	0.8867	0.6549
prod-I	RemBERT	Full feature set	0.8261	0.8881	0.8860	0.6669
prod-A	RemBERT	funnelling only	0.8266	0.8877	0.8860	0.6607
prod-B	COMET	Full + LID + entropy	0.8278	0.8920	0.8912	0.6503
prod-D	RemBERT	Vanilla	0.8415	0.8841	0.8804	0.6668

Table 1: Production-split leaderboard on the internal validation set (500 harder-biased instances; see §4.1). All hybrid models (all except prod-D) share a base feature set of normalised Levenshtein, Jaccard distance, pinyin romanisation, and subword ratio; the "Feature recipe" column describes additions beyond this base. Lower RMSE and MAE are better; higher Pearson and Spearman are better.

Run	ES RMSE	DE RMSE	CN RMSE
Open baseline	1.198	1.166	1.034
prod-H	1.046	0.968	0.883
prod-G	1.083	1.004	0.885
ensemble	1.034	0.945	0.861

Table 2: Official shared-task test results across the three L1 populations (Spanish, German, Chinese). Open baseline is the shared-task organisers' open-track baseline. Lower RMSE is better

ance between error and rank-based metrics. The ensemble placed fifth among open-track teams, outperforming the open-track baseline across all three $L1$ groups (Table 2). Chinese yielded the strongest result (RMSE 0.861) and Spanish the weakest (RMSE 1.034).

5.2 Training Dynamics

Figure 2 shows per-epoch development RMSE for the five ensemble members. The RemBERT models (prod-H, prod-J, prod-E, prod-F) converge smoothly within 7–12 epochs, with prod-H reaching its best checkpoint at epoch 11. The COMET model (prod-G) follows a slower, more oscillatory trajectory, peaking at epoch 21, consistent with its minimal feature setup requiring more updates to stabilise. prod-E (Huber $\delta = 0.5$) reaches a competitive checkpoint early but plateaus, confirming that the robust loss accelerates initial convergence without improving the final optimum.

5.3 Ablation Findings

Starting from the vanilla RemBERT baseline (prod-D), every targeted enrichment improves performance, but the gains are not uniform. Multi-layer semantic funnelling provides the clearest first improvement, TSA is the strongest single entropy feature, and LID prepend gives the best overall

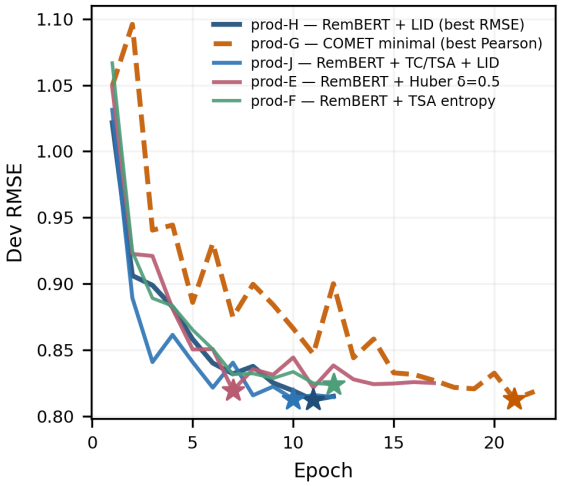

Figure 2: Per-epoch development RMSE for the five ensemble members. Stars mark best checkpoints.

RemBERT result. By contrast, the full feature set (prod-I) offers little additional benefit, suggesting diminishing returns once the strongest signals are included. Selective feature additions outperform broad feature bundles: prod-H and prod-J are both substantially better than prod-I despite sharing the same funnelling and backbone.

More broadly, the comparison between RemBERT and COMET suggests that the same feature strategy does not transfer equally across backbones: COMET performed best in minimal form, while RemBERT responded clearly to targeted linguistic and representation-level additions. COMET is pre-exposed to a large-scale quality estimation corpus but we chose to extract weights from the base model with $d = 512$. We hypothesize that its smaller embedding dimension ($d = 512$) limits its capacity to capture fine-grained cross-lingual information compared to RemBERT ($d = 1152$).

5.4 *L1*-specific Patterns

Per-*L1* correlations between external features and GLMM scores (Figure 3) reveal a clear typological split. Lexical similarity features (normalised Levenshtein, Jaccard) correlate moderately and negatively with difficulty for Spanish (-0.24, -0.27) and German (-0.34, -0.35) learners, consistent with cognate proximity easing recognition. For Chinese learners these correlations collapse to near zero (0.07, 0.06), as character-level similarity to English carries no meaningful signal even after pinyin romanisation. Subword ratio, by contrast, correlates similarly across all three L1s, suggesting that morphological-complexity proxies generalise where orthographic ones do not.

6 Conclusion

We presented a multilingual system for lexical difficulty prediction that combines multi-layer semantic funnelling, attention-derived entropy features, and cross-lingual string similarity with a hybrid transformer regression head. Our best single model achieved RMSE 0.8122 and Pearson 0.8968 on the internal production validation set.

On the official test set, the ensemble achieved RMSE 0.861–1.034 across the three $L1$ groups, outperforming the open-track baseline in all settings. A small number of well-motivated features, combined with a strong multilingual encoder and a carefully constructed validation split, proved more effective than either the encoder alone or a larger undifferentiated feature bundle. The interaction between backbone choice and feature engineering warrants further investigation, particularly given how differently RemBERT and COMET responded to the same additions. More broadly, these results suggest that explicit linguistic feature design remains a useful component of multilingual lexical difficulty prediction, even in settings where large pretrained encoders are available.

7 Limitations

The system was developed and evaluated on data covering three $L1$ backgrounds. It is unclear how well the approach would generalise to learners from other $L1$ backgrounds, particularly those with writing systems or typological properties not represented in the training data. The collapse of orthographic similarity signal for Chinese learners suggests the feature set may not transfer to other non-Latin-script $L1$s without analogous romanisation steps.

The attention-derived and representation-level features are extracted from the same encoder used for prediction, meaning their usefulness is tied to the quality of that encoder's internal representations. It is not clear whether those features would remain informative with a different or weaker backbone.

Acknowledgements

Part of this research was supported by the Leverhulme Doctoral Scholarships Network for AI-Enabled Digital Accessibility (ADA), whose support is gratefully acknowledged.

References

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.

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.

Paul De Boeck. 2008. Random item irt models. *Psychometrika*, 73(4):533–559.

Karen J. Dunn. 2024. Random-item rasch models and explanatory extensions: A worked example using l2 vocabulary test item responses. *Research Methods in Applied Linguistics*, 3(3):100143.

Namrata Bhalchandra Patil Gurav, Akashdeep Ranu, Archchana Sindhujan, and Diptesh Kanojia. 2026. Domain-specific quality estimation for machine translation in low-resource scenarios. In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 630–650, Rabat, Morocco. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Peter J. Huber. 1964. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101.

Diptesh Kanojia, Malhar Kulkarni, Pushpak Bhattacharyya, and Gholamreza Haffari. 2020. Challenge dataset of cognates and false friend pairs from Indian languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages

3096–3102, Marseille, France. European Language Resources Association.

Diptesh Kanojia, Prashant Sharma, Sayali Ghodekar, Pushpak Bhattacharyya, Gholamreza Haffari, and Malhar Kulkarni. 2021. Cognition-aware cognate detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3281–3292, Online. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 61–70.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

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*. University of Toronto Press, Toronto.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16. Association for Computational Linguistics.

Archchana Sindhujan, Shenbin Qian, Chi Chun Matthew Chan, Constantin Orăsan, and Diptesh Kanojia. 2025. Alope: Adaptive layer optimization for translation quality estimation using large language models. In *Proceedings of the Conference on Language Modeling (COLM)*.

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 (BEA 2025)*, pages 160–174, Vienna, Austria. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

A Feature Inventory

A.1 Linguistic features

Lexical similarity features. We hypothesise that cognates across English and Romance languages (German and Spanish), given the context, will help identify a lexical difficulty signal. Unlike supervised cognate-detection approaches (Kanojia et al., 2021, 2020), we aim to capture shared features at the string level unsupervised, *i.e.,* without relying on existing resources or tools specifically for cognate detection. All string distance features are standardised using training-set statistics and applied consistently to development and test data. For Chinese, raw character-to-Latin distances are not meaningful, so *pinyin romanisation* is applied first (see language-aware features below).

> **Levenshtein distance:** the raw edit distance between the $L1$ source word and the English target word, counting insertions, deletions, and substitutions. Orthographically close pairs, such as cognates, tend to receive lower scores and be easier for learners.
>
> **Normalized Levenshtein distance:** Levenshtein distance divided by the length of the longer word, making the measure length-invariant and comparable across word pairs of different sizes.
>
> **Jaccard distance:** one minus the Jaccard similarity of character bigram sets from the source and target words. This captures shared character structure independently of word order, and is more robust than edit distance for morphologically related pairs.

Subword Ratio the number of subword tokens produced by the backbone tokenizer divided by the character length of the target word. Words that fragment into many subword units tend to be morphologically complex or low-frequency, making this a lightweight proxy for lexical difficulty that requires no external resource.

> **Language ID:** optionally, we prepend a text tag (e.g. `[`$L1$`: es]`) to the input sequence before tokenisation, allowing the encoder to condition its representations on the learner's $L1$ from the first layer.

Language ID embeddings: a learned embedding lookup (Spanish → 0, German → 1, Chinese → 2) whose output is concatenated to the handcrafted feature vector before the regression head, providing an explicit $L1$ signal at the prediction stage.

Pinyin romanisation: for Chinese instances, the $L1$ source word is converted from CJK characters to pinyin before computing Levenshtein and Jaccard distances. This makes cross-lingual string similarity meaningful for a script that otherwise shares no characters with English.

A.2 Encoding-based Features

Attention-derived features. These features are computed from the encoder's attention distributions using token-level masks that identify the target word, $L1$ context, and $L1$ source word spans within the input sequence.

Target-to-context entropy: Shannon entropy of the mean attention weights from target word tokens to context tokens, averaged across heads in the final layer. High entropy indicates the target word attends broadly and diffusely to its context.

Target-to-source alignment: total attention mass flowing from target word tokens to $L1$ source word tokens in the final layer. A high value indicates strong cross-lingual correspondence between the target and its $L1$ translation.

Target-to-source entropy: entropy of the attention distribution from target tokens to source tokens. Low entropy indicates focused, peaked alignment to specific source tokens; high entropy indicates diffuse or uncertain alignment.

Context-to-target entropy: entropy of attention from context tokens to the target word. High entropy means the context does not attend strongly or focally to the target, which may indicate lower contextual salience.

Global CLS entropy: entropy of the `[CLS]` token's attention over the full input sequence. This provides a sequence-level signal about how broadly the model distributes global attention.

Attention variance: the variance of attention weights across heads rather than their entropy. This captures disagreement between attention heads, which may reflect ambiguity or heterogeneous cross-lingual signal.

Representation-level features These are derived from representations obtained from different layers of the Transformer backbones.

Multi-layer funnelling provides the regression head access to representations at different levels of abstraction, from lower-level morphological features to higher-level contextual ones. Inspired by (Sindhujan et al., 2025; Gurav et al., 2026), we propose extracting multi-layer semantic representations via *funnelling* where target-word hidden states are mean-pooled at each selected layer and concatenated.

$$\mathbf{p}^{(\ell)} = \frac{1}{|T|} \sum_{t \in T} \mathbf{h}_t^{(\ell)}, \tag{1}$$

$$\mathbf{f}_{\text{fun}} = \left[\mathbf{p}^{(\ell_1)}; \mathbf{p}^{(\ell_2)}; \ldots; \mathbf{p}^{(\ell_K)}\right]. \tag{2}$$

where T is the set of target-word token positions identified via character-level offset tracking, $\mathcal{L}$ the selected transformer layers (12, 20, 26, and 32 for RemBERT; 8, 10, and 12 for COMET), and $[\cdot\,;\,\cdot]$ denotes vertical concatenation. This gives the regression head access to representations at different levels of abstraction and was an important component of our strongest RemBERT-based configurations.

Target-word pooling: rather than using the `[CLS]` token as the sequence representation, the model mean-pools over the hidden states of tokens that correspond to the target word, identified via character-level offset tracking. This grounds the prediction in the specific word being assessed rather than a global sequence summary.

B Additional Details

Figure 3 (discussed in §5.4) additionally shows correlations for target length and clue mask ratio, which were explored during analysis but not included as model features. These two features show consistent moderate negative correlations across all three L1s (-0.33 to -0.46), comparable in magnitude to the strongest similarity features, but were

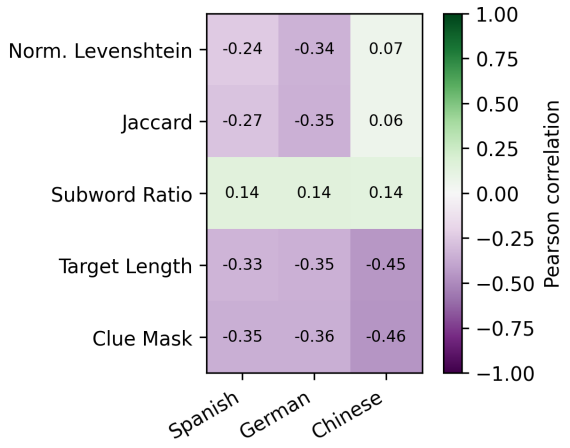

Figure 3: Pearson correlations between external lexical features and GLMM difficulty scores, computed separately per learner $L1$.

excluded from the final model because they correlate with the structure of the task input itself (clue masking and word length) rather than with linguistic properties.

B.1 Hyperparameters

All production models share the following training configuration:

Hyperparameter	Value
Learning rate	3×10^{-5}
Train batch size	32
Gradient accumulation steps	1
Weight decay	0.01
Max sequence length	256
Loss function	Huber ($\delta = 1.0$)
Seed	42

Table 3: Hyperparameters across all production models.

Model-specific settings that differ: `prod-E` uses Huber $\delta = 0.5$; RemBERT models (`prod-A`, `prod-C`, `prod-F`, `prod-H`, `prod-J`) train for 12 epochs; `prod-D` and `prod-I` for 14; `prod-E` for 17; and COMET models (`prod-B`, `prod-G`) for 22 epochs, reflecting their slower convergence.

B.2 Ensemble Weights

The submitted ensemble combines the five best validation systems using weights proportional to $1/\text{RMSE}^2$, normalised to sum to one. The top models (`prod-H`, `prod-G`, `prod-J`) receive nearly equal weight due to their similar validation scores. The resulting weights are:

Model	Dev RMSE	Weight
`prod-H`	0.8122	0.2019
`prod-G`	0.8127	0.2017
`prod-J`	0.8129	0.2016
`prod-E`	0.8194	0.1984
`prod-F`	0.8239	0.1963

Table 4: Ensemble member weights for the final submission.