uogal at BEA 2026 Shared Task 1: Ensemble of Multilingual Encoders with NMT Augmentation for L1-Aware Vocabulary Difficulty Prediction

Bernardo Stearns[1] **Thomas Gaillat**[2]
John P. McCrae[1] **Jefkine Kafunah**[1]
[1] Research Ireland Insight Centre for Data Analytics,
Data Science Institute, University of Galway, Ireland
[2] LIDILE / Université Rennes 2, 35000 Rennes, France
Contact: bstearnsreisendepinho@universityofgalway.ie

Abstract

This paper describes a system submission to the closed track of the BEA 2026 shared task on L1-aware vocabulary difficulty prediction (Spanish, German, Mandarin Chinese). We explored three approaches: hand-crafted tabular features with tree-based regressors, fine-tuned multilingual encoders, and fine-tuned decoders (artificial learner simulation with LoRA-tuned Pythia models). Each was examined with and without an English context produced by neural machine translation (NMT), and we also combined all three families in a single ensemble. In the end, ensembling *only* the fine-tuned multilingual encoders (four base architectures together with four NMT-augmented variants, combined through per-language stacking) gave the best results while remaining the simplest configuration to train.

1 Introduction

Vocabulary knowledge is fundamental to language proficiency, directly influencing what learners can comprehend and produce (Laufer, 1997). Establishing word difficulty is essential for creating level-appropriate educational content and valid assessment instruments, yet traditional calibration methods rely on costly expert judgment and pretesting, limiting scalability.

The BEA 2026 shared task (British Council, 2026) addresses this challenge by framing vocabulary difficulty prediction as a regression problem: given an English word and its associated context, predict its difficulty for learners with a specific first language (L1) background. The task uses the British Council's Knowledge-based Vocabulary Lists (KVL; Schmitt et al., 2021), a dataset of 6,768 items per L1 with psychometrically calibrated GLMM difficulty scores across three L1 groups: Spanish (Es), German (De), and Mandarin Chinese (Cn). Two tracks are offered: *closed* (separate per-language models) and *open* (a single multilingual model).

While previous shared tasks have addressed related problems, such as complex word identification (Paetzold and Specia, 2016; Yimam et al., 2018) and lexical complexity prediction (Shardlow et al., 2021, 2024), they did not consider the learner's L1 background, which is known to strongly influence L2 vocabulary acquisition through crosslinguistic transfer (Jarvis and Pavlenko, 2008). The last BEA language learning shared task was on grammatical error correction in 2019 (Bryant et al., 2019).

Our submission makes the following contributions:

1. A systematic comparison of 11 multilingual encoder architectures for L1-aware difficulty prediction;
2. Encoder training augmented with neural machine translation (NMT) using three translation systems with two decoding modes, providing complementary crosslinguistic signal;
3. An ensemble of eight encoders with per-language optimized stacking weights;
4. A tabular feature study combining multilingual embeddings, typological, phonetic, CEFR/WordNet, and cognate features with XGBoost and LightGBM regressors;
5. A scaling study of decoder-based artificial learner simulation (ALS) using Pythia models from 70M to 1.4B parameters.

2 Related Work

Lexical Complexity and Difficulty. The SemEval-2016 CWI shared task (Paetzold and Specia, 2016) introduced binary complex word identification, extended to multilingual settings by

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 1065–1076

Yimam et al. (2018). Shardlow et al. (2021) moved to continuous complexity prediction with the LCP shared task, and Shardlow et al. (2024) addressed multilingual lexical simplification. However, none of these tasks incorporated learner L1 as a predictive factor, even though there is longstanding evidence that L1 background shapes L2 vocabulary processing and difficulty.

L1 Influence on L2 Vocabulary. Crosslinguistic influence research has established that a learner's L1 systematically affects L2 vocabulary acquisition (Jarvis and Pavlenko, 2008). Cognates are easier to learn, while false friends create difficulty. This motivates L1-aware prediction models that can capture these transfer effects. This L1 influence has also been examined within language model-based approaches that simulate L2 learners, offering a complementary computational lens on the same phenomenon.

Artificial Learner Simulation. Several lines of work adapt or probe language models to simulate L2 processing and production.

Aoyama and Schneider (2024) pretrain GPT-2 sequentially on six different L1s and then on English as L2. The resulting "L2LMs" produce word surprisal that predicts nonnative reading times, with effects most pronounced for L1 backgrounds typologically distant from English.

In a related direction, Stearns et al. (2024) pretrain BERT on EFCAMDAT – the Education First-Cambridge Open Language Database (Geertzen et al., 2013), a publicly released corpus of ∼83M words of English-as-a-foreign-language essays from ∼550k EF Englishtown learners with self-declared L1 and CEFR-aligned proficiency labels – and C4200M – a 200M-sentence synthetic learner-error corpus built by injecting grammatical errors into clean C4 web text via a tagged corruption model, approximating the distribution of L2 mistakes at web scale (Stahlberg and Kumar, 2021) – and probe the resulting "artificial learners" via masked-token prediction, finding systematic divergence from a generic native model that is largest at lower CEFR levels.

Hu and Cong (2025) apply LLM-derived surprisal as a proficiency signal for Chinese L2 writing. They compare a multilingual, a Chinese-general, and a Traditional-Chinese-specific LLM on TOCFL essays across CEFR levels A2–C1, and show that surprisal differentiates proficiency and correlates with classic lexical and syntactic complexity indices – evidence that LLM-based probes apply cross-linguistically beyond English.

We extend this artificial-learner line to vocabulary difficulty regression by using Pythia decoder models (Biderman et al., 2023) with LoRA adaptation (Hu et al., 2022) as a next-token-prediction proxy for item difficulty.

3 System Description

3.1 Task and Data

The KVL dataset provides 6,768 parallel vocabulary items across three L1 groups (Spanish, German, Mandarin Chinese), split into training (∼6,091 items/language) and development (677 items/language) sets. Each item consists of:

- `en_word`: the target English word
- `en_clue`: a letter clue for the word
- `L1_word`: the L1 translation
- `L1_context`: an L1 contextual prompt
- `GLMM_score`: the target difficulty (lower = harder)

Items with the same `item_id` across L1 files refer to the same English word, enabling cross-language analysis.

3.2 Encoder Models

We fine-tune 11 multilingual encoder architectures for regression, using the full model parameters (no adapter layers). The input is constructed by concatenating all available text fields separated by `[SEP]` tokens in the order: `L1_word`, `L1_context`, `en_word`, `en_clue`. A single regression head maps the `[CLS]` representation to the predicted difficulty score.

Table 1 summarizes the training configuration. All models use identical hyperparameters: learning rate 1×10^{-5}, batch size 8 (effective 16 with gradient accumulation), 10 epochs with early stopping (patience 3), AdamW optimizer with weight decay 0.01 and 10% warmup, and FP16 mixed precision. The maximum sequence length is 256 tokens. We select the best checkpoint by development-set Pearson correlation. The regression target is the GLMM lexical-difficulty score provided by the organizers; the head is trained with mean-squared-error loss.

The 11 architectures tested are: Multilingual E5-Large (Wang et al., 2022), XLM-R Large and Base (Conneau et al., 2020), RemBERT (Chung

Hyperparameter	Value
Learning rate	1×10^{-5}
Batch size (effective)	16
Max epochs	10
Early stopping patience	3
Max sequence length	256
Classifier dropout	0.1
Optimizer	AdamW
Weight decay	0.01
Warmup ratio	0.10
Precision	FP16

Table 1: Encoder training hyperparameters.

et al., 2021), InfoXLM-Large (Chi et al., 2021a), Paraphrase Multilingual MPNet (Reimers and Gurevych, 2019, 2020), Multilingual E5-Base, mBERT (Devlin et al., 2019), CANINE-S (Clark et al., 2022), XLM-Align Base (Chi et al., 2021b), and Chinese RoBERTa Large (Liu et al., 2019). All checkpoints were obtained from the HuggingFace Hub. Of these, the top four (E5-Large, XLM-R Large, RemBERT, InfoXLM-Large) were selected for the ensemble and NMT augmentation experiments.

3.3 NMT Augmentation

We hypothesize that augmenting the L1 context with English translations from NMT systems provides complementary crosslinguistic signal, effectively giving the encoder both the L1 perspective and multiple L2 (English) interpretations of the same content. We generate translations using three NMT systems of varying capacity:

- **OPUS-MT** (Tiedemann and Thottingal, 2020): ∼77M parameters
- **NLLB-600M**: distilled 600M model (NLLB Team et al., 2022)
- **NLLB-1.3B**: full 1.3B model (NLLB Team et al., 2022)

For each system, we produce translations in two decoding modes: (1) *free beam search*, where the model generates its own English translation from the L1 input, and (2) *forced decoding*, where the model is constrained to produce the target English word. This yields six additional translated text strings per item.

The NMT-augmented encoders use the same four top architectures but with an extended input sequence that interleaves the original L1 context with these translated L2 contexts: `L1_word`, `L1_context`, followed by the six NMT translations, then `en_word`, `en_clue`, all concatenated as text with `[SEP]` separators. The maximum sequence length is increased to 384 tokens to accommodate the longer input. Batch size is increased to 12; all other hyperparameters remain identical.

Spanish item, target "flock"	
`L1_word`	`rebaño`
`L1_context`	`Un ___ de ovejas pasta en el campo.`
OPUS free	`A herd of sheep grazes in the field.`
OPUS forced	`A flock of sheep grazes in the field.`
NLLB-600M free	`A flock of sheep grazes in the field.`
NLLB-600M forced	`A flock of sheep grazes in the field.`
NLLB-1.3B free	`A flock of sheep is grazing in the field.`
NLLB-1.3B forced	`A flock of sheep grazes in the field.`

Table 2: Worked example of NMT-augmented input for a Spanish item. Three NMT systems each emit a free and a forced decode; the six resulting English strings are concatenated to the L1 fields. Disagreements between free and forced decodes (OPUS's "herd" vs. the gold "flock") expose the encoder to lexical alternatives consistent with the L1 context.

3.4 Artificial Learner Simulation

We hypothesize that diversifying the model's embedding space away from native-speaker distributional norms and towards representations conditioned on learner production (which we refer to as *learner embeddings*) provides complementary predictive signal for vocabulary difficulty: words that are hard for L2 learners may be unremarkable under a native prior but surface as atypical once the model's lexical distribution has been shifted towards L2 usage. Following this intuition, our *artificial learner simulation* (ALS) pipeline shifts a Pythia (Biderman et al., 2023) decoder's next-token distribution towards L2 learner production and uses the resulting learner-conditioned representation as the input to the regression head. We chose decoder-based rather than encoder-based artificial learners to maximise architectural diversity relative to the multilingual encoders of Section 3.2; encoder artificial learners remain a separate ablation we did not run.

Concretely, the ALS pipeline has two stages: (i) a learner-style language-modelling pretraining on EFCAMDAT that yields an *artificial learner*

checkpoint, and (ii) a regression adaptation of that checkpoint to the shared-task data.

Stage 1: EFCAMDAT pretraining. We continue the causal-LM pretraining of Pythia (Biderman et al., 2023) at five scales (70M, 160M, 410M, 1B, 1.4B) on the full EFCAMDAT corpus (Geertzen et al., 2013) of L2 English learner essays, using the standard next-token prediction objective and no modification to the model architecture. After this stage, each checkpoint (`pythia-{size}-all-data`) is a decoder whose distribution over continuations reflects the lexical and grammatical choices of L2 learners rather than those of a native reference corpus.

Stage 2: Regression adaptation. To transfer an artificial-learner checkpoint to the shared task, we discard the pretrained LM head and attach a randomly initialised 2-layer MLP regression head of the form `Dropout` $\rightarrow$ `Linear`$(h, h/2)$ $\rightarrow$ `GELU` $\rightarrow$ `Dropout` $\rightarrow$ `Linear`$(h/2, 1)$, where h is the backbone hidden size. Items are serialised as `en_word | L1_word | en_clue | L1_context` followed by the suffix "`Difficulty:`", left-padded, and the sequence representation is obtained by taking the hidden state of the last non-padding token, the decoder-appropriate analogue of the `[CLS]` pooling used by the encoder models of Section 3.2. The backbone itself is frozen except for LoRA adapters (Hu et al., 2022) with rank $r = 16$, $\alpha = 32$, dropout 0.05, applied to the `query_key_value` and `dense` projections of every GPT-NeoX block; the regression head is trained at full precision. Training minimises MSE with AdamW, learning rate 2×10^{-5}, effective batch size 16, up to 10 epochs with early stopping on dev Pearson r.

3.5 Tabular Features

We additionally extract a set of hand-crafted features organized into nine modules. For each module below we report the implementation actually used:

- **Embeddings** (1,154): Multilingual E5-Large sentence embeddings for the L1 context, target word, and source, plus pairwise cosine similarities; computed with `sentence-transformers` (Reimers and Gurevych, 2019).
- **Language pair typology** (226): morphological and typological features comparing each L1 to English, looked up from a hypergranular morphological-features table derived from WALS (Dryer and Haspelmath, 2013) and expressed as both raw L1 values and L1−English difference vectors.
- **IPA phonetic** (95): articulatory, sonority, and syllable-structure features over the G2P-converted target word, computed with `panphon`'s `FeatureTable` and `Distance` modules (Mortensen et al., 2016).
- **CEFR + WordNet** (12): per-word CEFR statistics (presence, sense count, min/max/mean/range level, and binary indicators for A1–C2), obtained from a sensekey-to-CEFR lookup table built over WordNet (Fellbaum, 1998) senses.
- **Cognate similarity**: Levenshtein distance, longest-common-subsequence length, character-bigram Jaccard, and shared-prefix length between the English target and its L1 translation; all implemented as plain Python, no external distance library.
- **Clue features**: length, revealed-positions length, vowel presence, and blank-to-letter ratio over the `en_clue` field.
- **Counts / POS / L1**: string-length and word-count statistics; one-hot encoding of a pre-annotated `en_target_pos` column; one-hot L1 language indicators (es/de/cn).
- **NMT-derived**: five features per NMT system capturing whether the target word appears in the free translation, the normalised Levenshtein distance between forced and free decodes, the Pythia perplexity of the forced translation, the forced/free perplexity ratio, and the NMT-score drop.
- **Frequency**: Zipf-scaled word-frequency bands for the English target, retrieved via `wordfreq` (Speer, 2022).

We fit regressors from the `scikit-learn` toolkit (Pedregosa et al., 2011), covering linear regression, Ridge, Lasso, decision trees, and tree ensembles (random forest, extra trees, gradient boosting, AdaBoost), together with XGBoost (Chen and Guestrin, 2016) and LightGBM (Ke et al., 2017). The best-performing tabular configuration combines all feature groups with ALS-derived

embeddings (`all_concat_als_xgb`, $r = 0.797$), using `XGBRegressor` with $n_{\text{estimators}} = 1000$, `max_depth` $= 6$, learning rate 0.03, and L_1/L_2 regularisation coefficients 0.1. Feature extraction relied on `sentence-transformers` (Reimers and Gurevych, 2019) for the 1,154 Multilingual E5-Large embeddings and `panphon` (Mortensen et al., 2016) for the 95 articulatory, sonority, and syllable-structure features. The full per-configuration breakdown across ∼15 feature/regressor combinations is reported in Appendix A.

3.6 Ensemble Strategy

Our final system ensembles the predictions of eight encoder models: four base and four NMT-augmented (Section 3.3). We experimented with two stacking strategies:

Nelder-Mead. We compute a weighted average of predictions: $\hat{y} = \sum_{i=1}^{8} w_i \cdot \hat{y}_i$, where weights $w_i \geq 0$ and $\sum w_i = 1$. The weights are fit *separately per language* using the Nelder-Mead simplex method (Nelder and Mead, 1965), as implemented in `scipy.optimize.minimize` (Virtanen et al., 2020) with `xatol` $= 10^{-6}$ and `fatol` $= 10^{-8}$, to maximize Pearson correlation on the 677 development items. This is a non-learning approach: no meta-model is trained, and the only free parameters are the 8 weights per language.

ElasticNet Meta-Learner. We train a `scikit-learn` (Pedregosa et al., 2011) ElasticNet regressor (Zou and Hastie, 2005) ($\alpha = 0.1$, l_1_ratio $= 0.5$) on the training-set predictions of the eight encoders.

4 Results

4.1 Individual Encoder Performance

Table 3 presents the development-set results for all 11 encoder architectures. Multilingual E5-Large achieves the best individual performance ($r = 0.845$), followed closely by XLM-R Large ($r = 0.843$) and RemBERT ($r = 0.843$). There is a clear divide between the top-four large models and the rest: the gap between 4th (InfoXLM-Large, $r = 0.829$) and 5th (Paraphrase MPNet, $r = 0.810$) is substantial.

4.2 NMT Augmentation Effect

Table 3 shows the impact of NMT augmentation on the top-four encoders. For each architecture, we compare the base model with its NMT-augmented variant (using all six NMT contexts: OPUS-MT, NLLB-600M, NLLB-1.3B, each with free and forced decoding). NMT augmentation consistently improves Pearson r across all four architectures, with the largest gain for InfoXLM-Large (+0.021). RMSE improves for XLM-R Large and RemBERT but slightly worsens for E5-Large and InfoXLM-Large.

4.3 Artificial Learner Simulation

Table 3 presents the ALS scaling study. Performance scales monotonically with model size, from $r = 0.494$ (70M) to $r = 0.812$ (1.4B). However, even the largest ALS model underperforms the best individual encoder ($r = 0.845$), suggesting that for this task, direct fine-tuning of encoder representations is more effective than the decoder-based ALS approach. Several follow-up experiments remain open, including the NMT-augmented variant of the 1.4B checkpoint, larger Pythia scales, encoder-based artificial learners, and adding the Pythia ALS models to the final ensemble alongside the multilingual encoders; we could not run them within the shared-task timeline and leave them to future work.

4.4 Tabular Feature Models

We trained approximately 40 scikit-learn regressors over the hand-crafted feature modules described in Section 3.5, formed by combining feature subsets with five regressor families (linear regression, ridge, decision tree, XGBoost, LightGBM). The strongest configuration, `all_concat_als_xgb`, concatenates all features with ALS-derived embeddings and is fit with XGBoost, reaching $r = 0.797$ on the development set. This result surpasses three of the base encoders (XLM-R Base, $r = 0.791$; mBERT, $r = 0.761$; CANINE-S, $r = 0.668$) and is comparable to Pythia 1B ALS ($r = 0.793$), but remains below the seven stronger base encoders and all NMT-augmented encoder variants. We retain the top two tree-based configurations for the 10-model ensemble experiments, but observe no improvement in results. The full per-configuration results appear in Appendix A.

4.5 Ensemble Results (Development Set)

Table 3 presents our ensemble results on the development set. The 8-encoder XGBoost blend (a learned non-linear meta-model) achieves $r = 0.882$, a +3.7% absolute improvement over the best single encoder. No-learning strategies (simple average, Nelder-Mead, hill climbing) cluster tightly between $r = 0.870$ and $r = 0.871$, and the ElasticNet meta-learner achieves $r = 0.871$. On hidden

System	Es	De	Cn	Avg r	RMSE↓ (primary)
Individual encoders — base (dev)					
E5-Large	.843	.843	.851	.845	1.025
XLM-R Large (v1)	.840	.845	.845	.843	1.085
RemBERT	.840	.840	.848	.843	0.980
XLM-R Large (v2)	.833	.843	.843	.839	1.105
InfoXLM-Large	.827	.826	.833	.829	1.166
Para. MPNet	.808	.806	.816	.810	1.094
E5-Base	.804	.803	.812	.806	1.166
XLM-R Base	.789	.787	.796	.791	1.195
mBERT	.758	.755	.768	.761	1.202
CANINE-S	.665	.668	.672	.668	1.364
Individual encoders — NMT-augmented (dev)					
E5-Large +NMT	.852	.857	.841	.850	1.046
XLM-R Large +NMT	.855	.854	.835	.849	0.994
RemBERT +NMT	.854	.845	.842	.848	0.963
InfoXLM-Large +NMT	.849	.858	.842	.850	1.043
Individual decoders — artificial learner simulation, no NMT (dev, seed 42)					
Pythia 70M	.481	.530	.465	.494	1.512
Pythia 160M	.666	.601	.590	.622	1.362
Pythia 1B	.801	.790	.785	.793	1.068
Pythia 1.4B	.824	.822	.790	.812	1.030
Individual decoders — artificial learner simulation, NMT features (dev, seed 42)					
Pythia 70M +NMT (single, opus-mt)	.580	.561	.539	.561	1.440
Pythia 70M +NMT (all 3 systems)	.563	.557	.549	.557	1.444
Tabular / scikit-learn models (dev; best of ∼15, full breakdown in Appendix A)					
all_concat_als_xgb (best)	.813	.784	.793	.797	1.099
Submitted 8-encoder ensembles (dev; full ablation in Appendix C)					
Nelder-Mead	**.875**	**.870**	**.868**	**.871**	0.902
ElasticNet meta	**.875**	**.870**	.867	**.871**	**0.891**
Baselines (shared task)					
Closed	.748	.753	.736	.745	1.287
Open	.787	.800	.804	.797	1.125

Table 3: Train-dev split results across all model families: base and NMT-augmented multilingual encoders, decoder-based artificial learner simulation (Pythia, with and without NMT features), tabular/scikit-learn regressors, the two submitted 8-encoder ensemble strategies (Nelder-Mead and ElasticNet meta-learner), and the shared-task baselines. Per-language Pearson r on the 677-item dev set is shown where reported per language; otherwise overall (pooled) r/RMSE is given. Baseline dev-set RMSE is the average across the three L1s, computed from the organizer-released per-L1 baseline predictions: Closed 1.287 (Es 1.357, De 1.328, Cn 1.175); Open 1.125 (Es 1.206, De 1.149, Cn 1.021). Test-set RMSE for both baselines appears in Table 4. Tabular section shows only the best configuration; the full breakdown is in Appendix A. GPT-2 decoder baselines are reported in Appendix B. Full ensemble-strategy ablation (alternative weighting schemes and 10-model variants) is in Appendix C.

test data, however, the Blend XGB advantage over these simpler strategies disappears (see §5): we therefore selected Nelder-Mead and ElasticNet for official submission, and treat the +0.011 dev-set gap as meta-model overfitting rather than generalisable signal.

Notably, including tabular models (XGBoost, LightGBM) in a 10-model ensemble slightly degraded performance ($r = 0.872$) compared to the 8-encoder-only ensemble, suggesting the tabular models introduced noise rather than complementary signal.

4.6 Official Test Results

Table 4 presents the official test set results for the closed track. Ranked by RMSE (the official metric), our best submission per language reaches 0.975 (Es, Nelder-Mead), 0.903 (De, ElasticNet) and 0.820 (Cn, ElasticNet), placing uogal **2nd** on Spanish and German behind Glite, and **3rd** on Chinese behind Glite and Sakura (which submitted 0.816). All four uogal submissions improve over the closed-track baseline by 0.28–0.36 RMSE points, and even the worse of the two beats the open-track baseline on every language. The same picture in Pearson r – 0.859 (Es), 0.869 (De), 0.880 (Cn) – shows ElasticNet ahead on Spanish/German and Nelder-Mead nominally ahead on Chinese by 0.001, illus-

Lang.	System	r	RMSE↓ (primary)
Es	Glite (1st)	.877	0.903
	uogal Nelder-Mead	.858	**0.975**
	uogal ElasticNet	.859	0.977
	Closed Baseline	.765	1.257
	Open Baseline	.783	1.198
De	Glite (1st)	.871	0.885
	uogal ElasticNet	**.869**	**0.903**
	uogal Nelder-Mead	.865	0.944
	Closed Baseline	.773	1.258
	Open Baseline	.786	1.166
Cn	Glite (1st)	.889	0.776
	Sakura (2nd)	.874	0.816
	uogal ElasticNet	.879	**0.820**
	uogal Nelder-Mead	.880	0.841
	Closed Baseline	.753	1.140
	Open Baseline	.804	1.034

Table 4: Official test set results (closed track). uogal ranks 2nd across all three languages, behind Glite.

trating that the two metrics disagree slightly on the Chinese ranking; we treat RMSE as authoritative since it is the official ranking metric.

5 Analysis

Primary Ranking Metric (RMSE). RMSE is the official ranking metric. Our best per-language submission reaches RMSE 0.975 (Es), 0.903 (De) and 0.820 (Cn), beating the closed-track baseline by 0.28–0.36 points (Table 4). On Chinese the two metrics order our submissions differently (r marginally prefers Nelder-Mead, RMSE clearly prefers ElasticNet); we follow the RMSE verdict throughout.

Superiority of Ensembling. Ensembling provides a substantial gain over the best individual model on the official metric. The best base encoder reaches RMSE 0.980 (RemBERT) and the best NMT-augmented encoder reaches 0.963 (RemBERT +NMT). On the dev set the 8-encoder Blend XGB meta-model reaches RMSE 0.851, a 0.13 absolute improvement over the best base encoder and 0.11 over the best NMT-augmented one (Table 3); the no-learning and linear-meta strategies (simple averaging, weighted-RMSE, weighted-Pearson, hill climbing, Nelder-Mead, ElasticNet) reach dev RMSE between 0.891 and 0.973. The dev-set advantage of the learned XGBoost blend *does not transfer to test*: Blend XGB underperforms the simpler Nelder-Mead and ElasticNet strategies on the hidden test set, and was therefore excluded from our two submissions. We read this as meta-model overfitting to the 677 dev items used to train the blend — a learned non-linear stacker has enough capacity to absorb dev-set idiosyncrasies that do not generalise, whereas the constrained weighting strategies cannot. Accordingly, the robust portion of the ensemble gain is the ≈ 0.08 RMSE drop shared by every no-learning and linear-meta strategy, which we attribute to *model diversity* rather than to any particular weighting scheme; the two submitted ensembles exceed the closed-track baseline by 0.28 (Es), 0.36 (De) and 0.32 (Cn) RMSE points on test (Table 4).

Variance of Encoder Results. Across the 11 encoders of Table 3 the cross-architecture spread in RMSE is large – from 0.980 (RemBERT) to 1.364 (CANINE-S), a range of 0.38 points – and similar in shape to the cross-architecture Pearson spread. Per-L1 Pearson columns (where the table reports them) further show that base encoders are remarkably stable across L1s (0.005–0.013 Pearson spread, e.g. XLM-R Large .840/.845/.845), so cross-architecture variance dominates cross-L1 variance by roughly an order of magnitude. NMT augmentation roughly doubles cross-L1 Pearson variance (up to 0.020 for XLM-R Large +NMT: .855/.854/.835) and consistently pulls Chinese down relative to Spanish and German – the English-centric NMT systems contribute complementary signal for the two European L1s but interfere with Chinese. This pattern justifies the ensemble design choice of combining multiple *architectures* rather than multiple seeds of a single model.

Language-Specific Weight Preferences. The Nelder-Mead per-language weights reveal interesting patterns. For Spanish, the NMT-augmented XLM-R Large and RemBERT receive the highest weights; for German, NMT-augmented E5-Large and InfoXLM-Large dominate; for Chinese, the base E5-Large and XLM-R Large are preferred. This suggests that different encoder-L1 combinations capture complementary aspects of difficulty, motivating per-language weight fitting over a global weighting scheme.

Encoder vs. Decoder for Difficulty Prediction. The best encoder reaches RMSE 0.980 (RemBERT) while the best decoder-based ALS checkpoint reaches 1.030 (Pythia 1.4B), a gap of 0.05 RMSE points in favour of encoders. While the ALS models show clean monotonic RMSE scaling – $1.512 \rightarrow 1.362 \rightarrow 1.068 \rightarrow 1.030$ from 70M to

1.4B parameters – they do not close the gap, and the encoder approach is both more effective and more computationally efficient for this regression task.

NMT Augmentation Helps Both Individually and in Ensembles. NMT augmentation improves RMSE for three of the four top architectures (Table 3): XLM-R Large (1.085 → 0.994), RemBERT (0.980 → 0.963) and InfoXLM-Large (1.166 → 1.043), with E5-Large the lone exception (1.025 → 1.046); Pearson rises monotonically across all four. The 8-encoder ensemble (4 base + 4 NMT) reaches dev RMSE 0.891 (ElasticNet) and 0.902 (Nelder-Mead), improving over the best individual NMT model by 0.07–0.08 RMSE points. Exposing the encoder to multiple English renderings of the same L1 input, produced by NMT systems of varying capacity, likely surfaces complementary aspects of crosslinguistic transfer. We did not separately evaluate a base-only (4 encoder) ensemble, so we cannot directly attribute the ensemble gain to the NMT variants alone vs. to model diversity in general.

6 Conclusion

We presented the *uogal* system, which placed 2nd in the closed track of the BEA 2026 shared task on L1-aware vocabulary difficulty prediction. Our approach combines eight fine-tuned multilingual encoders—four base architectures and four NMT-augmented variants—in an ensemble with two stacking strategies, achieving test set Pearson correlations of $r = 0.859$ (Es), $r = 0.869$ (De), and $r = 0.880$ (Cn). We additionally demonstrated monotonic scaling behavior in decoder-based artificial learner simulation, though this approach did not match encoder ensemble performance. Our results indicate that large multilingual encoders can effectively capture L1-specific vocabulary difficulty patterns without explicit L1 conditioning, and that NMT augmentation provides valuable ensemble diversity.

7 Future Work

There are numerous avenues to follow from this work. One direction concerns metadata-aware models, where the L1 identifier is provided explicitly rather than left implicit in the input: our early L1-concat experiments suggest that naive injection interferes with the encoder's multilingual representations, and open questions remain about how and where to introduce the L1 signal, whether through token placement, embedding initialization, prefix tuning, or adapter-style conditioning. A second direction is scaling, both in encoder backbones and in the artificial learner simulation side, where the observed trends have not yet plateaued and larger checkpoints or alternative decoder families may narrow the gap to encoder-based difficulty prediction. A third direction is improving the NMT augmentation pipeline by asking a more targeted question: given an L1 context in which a learner encountered a word, what is the best equivalent L2 context to generate? Better answers to this question, whether through context-aware translation, retrieval of comparable L2 passages, or learner-conditioned generation, could make augmentation a stronger signal for L1-specific vocabulary difficulty.

Limitations

Our system has several limitations. First, it was evaluated on only three L1 groups (Spanish, German, Mandarin Chinese); performance on typologically more distant or lower-resource L1 backgrounds remains unknown. Second, the ensemble of eight large models (∼560M parameters each) has substantial computational requirements for both training and inference, limiting practical deployment. Third, our ensemble weights were optimized on the development set (677 items per language), which, while a favorable parameter-to-sample ratio (∼85:1), carries some risk of overfitting. Finally, we report development-set performance for ablation studies (e.g., individual encoder comparison, ALS scaling); only the final ensemble submissions were evaluated on the official test set.

Acknowledgments

This work has been supported by Research Ireland under Grant Number SFI12RC2289_P2 Insight_2, Insight Research Ireland Centre for Data Analytics.

References

Tatsuya Aoyama and Nathan Schneider. 2024. Modeling nonnative sentence processing with L2 language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4927–4940, Miami, Florida, USA. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

British Council. 2026. BEA 2026 shared task: Vocabulary difficulty prediction for English learners. https://www.britishcouncil.org/data-science-and-insights/bea2026st. Data Science and Insights, British Council. Accessed 2026-04-22.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021a. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021b. Improving pretrained cross-lingual language models via self-labeled word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3418–3430.

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*.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

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, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online (v2020.4)*. Zenodo.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Jeroen Geertzen, Theodora Alexopoulou, and Anna Korhonen. 2013. Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database (EFCAMDAT). In *Proceedings of the 31st Second Language Research Forum (SLRF)*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jingying Hu and Yan Cong. 2025. Modeling Chinese L2 writing development: The LLM-surprisal perspective. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 172–183, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Scott Jarvis and Aneta Pavlenko. 2008. *Crosslinguistic influence in language and cognition*, 1 edition. Routledge, New York.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30.

Batia Laufer. 1997. The lexical plight in second language reading: Words you don't know, words you think you know, and words you can't guess. In James Coady and Thomas Huckin, editors, *Second Language Vocabulary Acquisition*, pages 20–34. Cambridge University Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. PanPhon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484.

John A. Nelder and Roger Mead. 1965. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Celebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Baez, Gabriel Birber, Johan Boscolo, and 51 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Norbert Schmitt, Karen Dunn, Barry O'Sullivan, Laurence Anthony, and Benjamin Kremmel. 2021. Introducing knowledge-based vocabulary lists (KVL). *TESOL Journal*, 12(4):e622.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, and 3 others. 2024. The BEA 2024 shared task on the multilingual lexical simplification pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Robyn Speer. 2022. wordfreq: v3.0.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th workshop on innovative use of NLP for building educational applications*, pages 37–47.

Bernardo Stearns, Nicolas Ballier, Thomas Gaillat, Andrew Simpkin, and John P. McCrae. 2024. Evaluating the generalisation of an artificial learner. In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 199–208, Rennes, France. LiU Electronic Press.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 16 others. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Configuration	r	RMSE
all_concat_als_xgb	**.797**	**1.099**
all_features_lgbm	.791	1.110
all_features_xgb	.790	1.112
all_concat_ipa_xgb	.782	1.133
all_features_full_xgb	.778	1.140
all_ipa_phonetic_xgb	.777	1.141
cvo_allconcat_xgb	.756	1.188
all_features_lr	.750	1.201
cvo_allconcat_ridge	.738	1.224
cvo_embedding_ridge	.689	1.316
als_nonmt_pythia410m_xgb	.618	1.429
cefr_wordnet_only_xgb	.508	1.563
ipa_phonetic_only_xgb	.507	1.566
cognate_xgb	.404	1.660
nmt_xgb	.385	1.673

Table 5: Detailed dev-set Pearson r and RMSE for the top 15 tabular / scikit-learn configurations over hand-crafted feature modules. Naming convention: `{feature_subset}_{regressor}`. The best configuration combines all feature modules with ALS-derived embeddings and an XGBoost regressor.

A Detailed Tabular Feature Results

Table 5 reports the dev-set performance of the strongest 15 hand-crafted-feature scikit-learn configurations. All runs use the feature modules described in Section 3.5 (Embeddings, Language-pair typology, IPA phonetic, CEFR + WordNet, Other) in various subsets, paired with five regressor families (linear regression, ridge, decision tree, XGBoost, LightGBM). Approximately 25 additional configurations with $r < 0.40$ (e.g., word-count-only, language-pair-only, single-feature ablations) are omitted for brevity; complete metrics are available in the project repository.

B GPT-2 Decoder Baselines

As a pre-ALS sanity check we fine-tuned native-English GPT-2 decoders (without the EFCAMDAT artificial-learner adaptation) on the same regression head and serialisation described in Section 3.4. These numbers serve purely as a reference point: a generic next-token model with no learner adaptation. They fall well below every multilingual encoder in the main table and below all Pythia-based ALS checkpoints from 410M upwards (Table 3), supporting the design choice to adapt decoders on learner data rather than use them off-the-shelf.

C Full Ensemble-Strategy Ablation

Table 7 reports the full set of ensemble weighting strategies we evaluated on the development set. In the main paper (Table 3) we show only the two submitted configurations (Nelder-Mead and ElasticNet meta-learner over the 8 encoders). The remaining rows below include alternative no-learning schemes (simple average, weighted-RMSE, weighted-Pearson, hill climbing, rank average), the learned non-linear Blend XGB meta-model, and a 10-model variant that adds the two strongest tabular configurations (`all_features_xgb`, `all_features_lgbm`) to the 8 encoders.

System	Es	De	Cn	Avg r	RMSE
GPT-2	.578	.609	.652	.610	1.476
GPT-2 Medium	.577	.601	.646	.606	1.483

Table 6: GPT-2 decoder baselines on the 677-item dev set. Neither model was adapted on learner data; they serve as a reference for the gain provided by EFCAMDAT-based artificial learner simulation (Table 3).

Strategy	Es	De	Cn	Avg r	RMSE
8-encoder ensembles (dev)					
Simple avg	.872	.870	.867	.870	0.973
Weighted RMSE	.872	.871	.867	.870	0.969
Weighted Pearson	.872	.871	.867	.870	0.970
Hill climb	.874	.869	.867	.870	0.910
Rank avg	.868	.871	.860	.866	1.520
Nelder-Mead (submitted)	.875	.870	.868	.871	0.902
ElasticNet meta (submitted)	.875	.870	.867	.871	0.891
Blend XGB	.885	.887	.875	**.882**	**0.851**
10-model ensembles — 8 encoders + 2 tabular (dev)					
Nelder-Mead	.878	.870	.868	.872	0.899
ElasticNet meta (dmeta)	.874	.872	.867	.871	0.906
Blend ElasticNet	.873	.873	.868	.871	0.891

Table 7: Full ensemble-strategy ablation on the 677-item dev set. The two submitted configurations (Nelder-Mead and ElasticNet meta-learner over 8 encoders) also appear in Table 3 of the main text. No-learning strategies cluster tightly at $r \in [.866, .871]$; the learned non-linear Blend XGB meta-model reaches the highest average Pearson ($r = .882$) on dev but was not selected for submission.