Sakura at BEA 2026 Shared Task 1: What Makes Vocabulary Difficult?

Adam Nohejl[1] **Xuanxin Wu**[2] **Yusuke Ide**[3]
Maria Angelica Riera Machin[3] **Yi-Ning Chang**[4] **Hitomi Yanaka**[1,5,6]

[1]RIKEN [2]The University of Osaka [3]Nara Institute of Science and Technology
[4]National Tsing Hua University [5]The University of Tokyo [6]Tohoku University

adam.nohejl@riken.jp xuanxin.wu@ist.osaka-u.ac.jp ide.yusuke.ja6@is.naist.jp
riera_machin.maria.rn9@naist.ac.jp changyn@gapp.nthu.edu.tw hyanaka@is.s.u-tokyo.ac.jp

Abstract

We describe two types of models for vocabulary difficulty prediction: a high-accuracy black-box model, which achieved the top shared task result in the open track, and an explainable model, which outperforms a fine-tuned encoder baseline. As the black-box model, we fine-tuned an LLM using a soft-target loss function for effective application to the rating task, achieving $r > 0.91$. The explainable model provides insights into what impacts the difficulty of each item while maintaining a strong correlation ($r > 0.77$). We further analyze the results, demonstrating that the difficulty of items in the British Council's Knowledge-based Vocabulary Lists (KVL) is often affected by spelling difficulty or the construction of the test items, in addition to the genuine production difficulty of the words. We make our code available online.[1]

1 Introduction

The goal of the BEA 2026 Shared Task on Vocabulary Difficulty Prediction for English Learners (Felice and Skidmore, 2026) is to build models of the difficulty of English words given a learner's L1. Such difficulty predictions can be used as a basis for pedagogical materials or computer-adaptive tests.

The shared task utilizes a large dataset with vocabulary difficulty scores for L1 Chinese, German, and Spanish learners, spanning thousands of vocabulary test items, the British Council's Knowledge-based Vocabulary Lists (KVL; Schmitt et al., 2021, 2024). Each test item, as shown in the example in Table 1, consists of an equivalent L1 word, L1 context, and a clue for the first letter of the English word and its length in letters.

The test format, therefore, focuses on productive knowledge, in particular, on the ability to write the English word with the correct spelling given the test prompt. The difficulty scores, which we aim to predict, are intercept values of a generalized linear mixed model (GLMM), i.e., the log-odds of a learner responding correctly.

L1		Spanish
English word		house
Part of speech		noun
Test item	L1 word	casa
	L1 context	Vivo en una casa grande que tiene tres dormitorios.
	Clue	h _ _ _ _ first letter and blanks
Difficulty score		3.07 ↑ easy, ↓ difficult

Table 1: Example of an item in the KVL data.

The shared task consisted of a closed track and an open track. While the open track was completely unrestricted, in the closed track, the use of large language models (LLMs) was limited to feature extraction, and only the L1-specific training data could be used. We present two approaches to vocabulary difficulty modeling in this setting, each leveraging LLMs in a different way:

LLMs fine-tuned using soft targets. We propose a simple yet novel technique to fine-tune LLMs for continuous value prediction using soft targets. Models based on this technique have outperformed all other shared task submissions in the open track.

Explainable model. We build a model using well-defined features, such as similarity to the L1 word and spelling difficulty, some of which are based on LLM prompting, and use SHapley Additive exPlanations (SHAP; Lundberg and Lee, 2017) to quantify the impact of each feature on the prediction. The model performed competitively in the closed track, surpassing the fine-tuned encoder baseline.

Our analysis indicates two factors contributing to difficulty scores in the KVL data beyond the productive difficulty of words: the spelling difficulty and the choice of L1 equivalents and context in some test items.

[1]https://github.com/ynklab/vocabulary-difficulty

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 1161–1178

2 Related Research

The task of vocabulary difficulty prediction, as construed by the BEA 2026 Shared Task (Felice and Skidmore, 2026), effectively combines lexical complexity prediction (LCP) or complex word identification (CWI) with test item difficulty estimation.

CWI is the task of identifying complex words in a sentence context. CWI shared tasks (Paetzold and Specia, 2016; Yimam et al., 2018) were dominated by feature-based systems focusing on word-level features.

LCP is an extension of CWI, where complexity is predicted on a continuous scale. The best performing approaches in an early shared task (Shardlow et al., 2021) were based on fine-tuning masked language models (MLMs). In the BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline (MLSP; Shardlow et al., 2024), where lexical complexity was predicted for ten languages with limited training data, the top-performing systems were based either exclusively on LLM prompting or on word-level features.

Nohejl et al. (2025a) proposed an LLM prompting-based LCP method, G-Scale, which aligns with the desired scale by applying temperature scaling to the probabilities and a linear regression to the final output. They also demonstrated that the addition of a single feature, log-frequency, can further improve the LLM-based predictions.

Smădu et al. (2024) compared feature-based models and fine-tuned Transformers (MLMs and LLMs) on CWI and LCP datasets with training data consisting of thousands of examples, i.e., in a setting similar to the present shared task. The study concluded that the LLMs rarely outperform the computationally less demanding MLMs and feature-based models. Smădu et al. (2024) fine-tuned LLMs only using the standard cross-entropy loss with discretized complexity values as hard targets. We achieve large improvements in LLM and MLM fine-tuning performance by using soft targets to sidestep discretization.

Cross-entropy loss with soft targets is often associated with knowledge distillation (Hinton et al., 2015) and has been used for distilling MLMs and LLMs (e.g., Sanh et al., 2019). To the best of our knowledge, however, it has not been used for fine-tuning to predict continuous values.

Similar to LCP and CWI, the goal of the vocabulary difficulty prediction task is to estimate the difficulty of words given a specific context that determines their sense. There are, however, prominent differences:

1. The complexity in LCP and CWI is measured by *subjective ratings*. The difficulty in the present task is based on the *success rate of test items*.

2. The input for LCP and CWI is only a word and its context, both in the same language. The input for this task consists of multiple elements in English and in L1.

3. LCP and CWI focus on *reading comprehension*. The present task focuses on *written production*.

We address the specific aspects of this task while drawing on insights from prior research on LCP and CWI.

Skidmore et al. (2025) fine-tuned encoders (i.e., MLMs) on the KVL data. Their fine-tuned XLM-RoBERTa model serves as the baseline of the shared task. They used SHAP as a tool for error analysis, attributing inaccurate predictions to specific input token positions. We use SHAP for explainability, attributing predictions to higher-level features such as spelling difficulty or similarity to L2.

3 Method

We propose two core methods: fine-tuning LLMs and MLMs with soft targets to predict the continuous difficulty values, and an explainable model with LLM-extracted features. To further improve the accuracy of both methods, we experiment with ensembling and additional features.

3.1 Fine-Tuning with Soft Targets

In the following, we assume an LLM or an MLM that has a token vocabulary V and predicts a probability distribution of tokens $i \in V$ conditioned on an input $\mathbf{x}$, denoted as $\hat{p}(i \mid \mathbf{x})$.

The standard loss function for MLM and LLM training is a cross-entropy loss with a hard target, where the entire probability mass is assigned to a single target token, i.e., the negative log likelihood of the target token. Adapting such models to predict continuous values calls for a different approach.

While encoder models (typically MLMs) are sometimes fine-tuned for the prediction of continuous values using a regression head and mean squared error (MSE) loss, this is not the case for LLMs. The prevalent supervised fine-tuning (SFT)

paradigm for LLMs is to convert such tasks to text generation by discretizing the continuous values and using the aforementioned cross-entropy with hard targets

$$\ell = -\log \hat{p}(v(d(y)) \mid \mathbf{x}), \tag{1}$$

where d is the discretization (e.g., rounding to the nearest integer), and v is the mapping of discretized values to tokens.

These common approaches to predicting continuous values using MLMs and LLMs are reflected in the LCP methods based on encoders (e.g., Ide et al., 2023), in the present shared task's encoder baseline (Skidmore et al., 2025), and in the LCP methods based on LLMs (e.g., Smădu et al., 2024). In contrast to the MSE loss used by the encoder-based regression, the standard cross-entropy loss for LLMs requires the continuous target values to be discretized into a small set of labels, losing precision in the process.

Our method sidesteps this apparent misalignment between text generation and the prediction of continuous values by fine-tuning LLMs using cross-entropy loss with soft targets

$$\ell = -\sum_{i \in V} p(i) \log \hat{p}(i \mid \mathbf{x}). \tag{2}$$

We prompt the model to predict values on a discrete scale S in the form of successive integer points, e.g., $S = \{1, 2, 3, 4, 5\}$. Our aim, however, is to predict continuous values $y \in [\min S, \max S]$. We therefore express y as a probability-weighted sum of its nearest points. Namely, we select two points, a and $a + 1$, on the scale S such that $a \leq y \leq a + 1$, and define the probability as

$$p(i) = \begin{cases} (a+1) - y & \text{if } i = v(a), \\ y - a & \text{if } i = v(a+1), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We use p as the soft target probability in the loss function defined in Equation 2.

At inference time, we predict the token probability distribution $\hat{p}$ of a single token and then compute the final output $\hat{y}$ as a token probability-weighted mean:

$$\hat{y} = \frac{\sum_{s \in S} \hat{p}(v(s) \mid \mathbf{x}) \cdot s}{\sum_{s \in S} \hat{p}(v(s) \mid \mathbf{x})}. \tag{4}$$

The same inference technique was used for LLMs by Liu et al. (2023) to infer continuous scores using few-shot learning, and its variations have been applied to LCP (Aumiller and Gertz, 2022; Enomoto et al., 2024; Smădu et al., 2024; Nohejl et al., 2025a) but without the complementary loss function for fine-tuning.

Because these two complementary techniques, soft-target cross-entropy loss and probability-weighted inference, require a single token to be predicted, they can be used not only for LLMs but also for MLMs via masked token prediction. For LLMs, the input $\mathbf{x}$ is simply the prompt; for MLMs, it is the sequence "[CLS] *prompt* [MASK] [SEP]".

3.2 Explainable Model

Directly fine-tuning of LLMs or MLMs on this task is efficient but results in a black-box model, i.e., a model whose inner decision process is difficult to interpret. As an alternative model explainable using SHAP, we train an XGBoost (Chen and Guestrin, 2016) regressor. SHAP provides an explanation of the model's output $f(\mathbf{x})$ via SHAP values ϕ_i that additively express the local importance of each feature x_i:

$$f(\mathbf{x}) = E[f(\mathbf{x})] + \sum_{i=1}^{n} \phi_i. \tag{5}$$

The SHAP values can be positive or negative, expressing how much each feature pushes a prediction higher or lower relative to the expected value. We use the additive nature of SHAP values and express the importance of groups of related features as sums of their SHAP values.

Our explainable model uses the following features:

Production frequency. We use the log-frequency of the English word in the Lang-8 learner corpus (Mizumoto et al., 2011), to estimate its written production frequency by learners. Three features represent the subcorpora of (1) all learners, (2) L1 Spanish learners, and (3) L1 Chinese learners. There was not enough data for L1 German learners.

Reception frequency. We use the log-frequency and log-range of the English word in corpora representative of spoken language to estimate its reception frequency. Namely, we use the logarithm of (1) frequency and (2) range (YouTube channels) in TUBELEX (Nohejl et al., 2025b), and (3) frequency in the spoken subcorpus of the British National Corpus (BNC; BNC Consortium, 2007).

CEFR level. We use the minimum CEFR level of the English word from the Cambridge English Vocabulary Profile (Capel, 2012)

Word length. We measure the length of the English word in letters.

L1 similarity. We compute the character-level similarity of the English word to the L1 word based on their length-normalized Levenshtein distance after removing diacritics and lowercasing. This feature is only applicable to languages written in the alphabet, Spanish and German.

Spelling difficulty. We prompt GPT-5.2 (OpenAI, 2025) to rate the spelling difficulty of an English word for each L1, given the L1 equivalent and the English word's pronunciation.

Lexical ambiguity. We prompt (1) GPT-5.2 and (2) DeepSeek-V3.2 DeepSeek-AI (2025) to determine the lexical ambiguity of the test item. A lexically ambiguous item must fulfill two conditions: (a) the English word is polysemous or one of several homonyms, and (b) the particular sense referenced by the test item is unfamiliar or challenging for learners.

L1 calque. We prompt GPT-5.2 to determine if the English word is a morpheme-for-morpheme translation of the L1 equivalent. The prompt excludes single-morpheme words and simple borrowings but does not require one word to be a calque of the other in the etymological sense. In contrast to the L1 similarity feature, L1 calque is also applicable to Chinese, e.g., 超级英雄 'superhero' (composed of morphemes 超级 'super-' and 英雄 'hero').

For each of the prompt-based features, including the binary ones, we apply G-SCALE (Nohejl et al., 2025a), i.e., temperature-scaled softmax followed by a probability-weighted mean of the LLM's predictions (Equation 4). The optimal temperature is determined for each model separately using cross-validation. This results in continuous feature values.

When reporting SHAP values, we sum the values within each group if it consists of multiple features (e.g., production frequency) without distinguishing the individual features.

We have submitted our explainable model as `explainable` and the model using only the traditional (not LLM-based) subset of features as `traditional`.

3.3 Ensembles and Additional Features

To further increase accuracy, we ensemble fine-tuned LLMs in a linear stack, combine fine-tuned LLM predictions with features, and experiment with additional features and fine-tuning encoder models using the same method we used for LLMs.

Linear stacking. When fine-tuning LLMs, we perform 5-fold cross-validation and use the out-of-fold predictions to fit a linear regression that combines the predictions of multiple LLMs in an ensemble. The linear regression is fit separately for each L1. The final ensemble uses models fine-tuned on the complete data (the union of the provided test and development subsets). As this approach is rather computationally intensive, we compare it with simple average ensembling. Corresponding submission: `finetuned_llms`.

Enhancing LLMs with features. The linear stacking approach allows us to add our explainable features to the same linear regression as the fine-tuned LLMs, building on the G-SCALE method proposed by (Nohejl et al., 2025a). Corresponding submission: `finetuned_llms_plus`.

Fine-tuned encoders and additional features. We add an encoder model fine-tuned on single-language data and several other features to the feature model. We do this to maximize performance within the rules for the closed track, although the resulting model is no longer explainable. The added features are: (1) frequency in OpenSubtitles (Lison et al., 2018), a corpus similar to the already included TUBELEX; (2) frequency in the written subcorpus of the BNC; (3) CEFR level in the Global Scale of English[2] (Jong et al., 2016), similar to the already included EVP level; (4) the Glasgow norms for concreteness, imageability, familiarity, and age of acquisition (Scott et al., 2019), which often strongly correlate with other already included features; and (5) an additional prompt for the calque feature. Corresponding submission: `closed_max`.

In the open track, we prompt two LLMs, GPT-4.1-mini and GPT-4.1-nano (OpenAI, 2024), to solve the test items, and we use their probability of a correct answer to indirectly assess the test item difficulty caused by the choice of the L1 word and context in the test item. To clearly distinguish this from the difficulty of the English vocabulary itself, we call this feature **trickiness**. We hypothesize

[2] https://www.english.com/gse/teacher-toolkit/user/vocabulary

that the LLMs have near-perfect knowledge of common vocabulary in the four relevant languages, and their performance therefore reflects the trickiness of the test items rather than the difficulty of the English vocabulary. Corresponding submission: `finetuned_llms_plus`, `open_max`.

We additionally experimented with prompting two recent LLMs, GPT-5.2 and GPT-4.1 (OpenAI, 2024), to directly rate the **difficulty** of the test items for learners of each L1 using 3-shot prompting. Corresponding submission: `open_max`.

Neither of the last two prompting approaches (trickiness and difficulty) were permitted in the closed track.

4 Experimental Settings

We experimented with multiple prompts for fine-tuning using a small LLM and 1-epoch training on a single L1. However, as we show in the ablation analysis in Section 6.1, with more training epochs, the prompt matters very little. For LLM prompting, we verified on a hand-picked sample that the model responses match our expectations, but we did not try to optimize the prompts.

In Appendix A, we provide complete listings of prompts used for fine-tuning and prompting.

As a basis for all fine-tuning experiments, we used pre-trained MLMs or LLMs, i.e., models not fine-tuned on chat or instruction data. For fine-tuning, we used recent LLMs with up to 32B parameters from the GLM-4 family (Team GLM, 2024), Qwen2.5 family (Qwen Team, 2025), and Ministral-3 family (Mistral AI, 2026), as well as MLMs mmBERT (Marone et al., 2025) and XLM-RoBERTa (Conneau et al., 2020). For prompting, we used models of the GPT-4.1 (OpenAI, 2024) and GPT-5.2 (OpenAI, 2025) families, and DeepSeek-V3.2 (DeepSeek-AI, 2025). Appendix B provides more details.

For MLMs, we performed end-to-end fine-tuning. For LLMs, we applied 4-bit quantization and fine-tuned QLoRA adapters (Dettmers et al., 2023) targeting all linear modules. All hyperparameters are described in Appendix C. We trained all models on the union of the test and development subsets provided by the shared task organizers. For the prompt-based features, we called models via the OpenAI API with zero temperature and requested log-probabilities necessary for probability weighting.

In all cases where a model is supposed to output a difficulty rating on a scale from 1 to 5, we map the difficulty scores from the KVL data linearly so that the highest score (the easiest item) in the training data maps to 5 and the lowest score to 1. The prompts are formulated accordingly. As the GLMM scores represent log-odds, we also experimented with mapping them to probabilities using the expit function (logistic curve), which decreased performance. See results in Appendix G.

System	Chinese	German	Spanish	Mean
`open_max`	0.631	**0.723**	0.743	**0.699**
`finetuned_llms_plus`	**0.630**	0.726	**0.742**	0.699
`finetuned_llms`	0.640	0.731	0.760	0.710
≤32B LLM Average	0.645	0.743	0.767	0.719
- GLM-4-32B	0.678	0.769	0.805	0.751
- Qwen2.5-32B	0.678	0.777	0.799	0.752
- Ministral-3-14B	0.681	0.781	0.799	0.753
≤14B LLM Average	0.662	0.770	0.804	0.745
≤9B LLM Average	0.683	0.791	0.835	0.769
Open-Track Baseline	1.034	1.166	1.198	1.133
Statistical Optimum	0.321	0.304	0.205	0.277

Table 2: RMSE of our open-track submissions, compared with average ensembles by model size, individual models, and the shared task's open-track baseline.

5 Results

In line with the shared task's evaluation, we report the root mean square error (RMSE) by L1 and compare it with the official shared tasks' baseline, a fine-tuned XLM-RoBERTa model (Felice and Skidmore, 2026).

Note that RMSE is in the same units as the difficulty scores, which typically range from −5 to +5. We also report the Pearson's correlation coefficients (PCC) r, the secondary evaluation metric, which may be easier to interpret, in Appendix D.

To further put the results into perspective, we report a "Statistical Optimum" result, which simulates the largest error (lowest correlation) that should be considered optimal given the precision of the gold standard data. The simulation is based on confidence intervals reported by Schmitt et al. (2024); see Appendix F for details.

5.1 Open Track

As shown in Table 2, the results of our three open-track submissions are very close to each other. In summary, the results do not justify the cost of adding features to fine-tuned LLMs. This contrasts with previous findings for in-context learning setting, where just adding frequency as a feature

System	Chinese	German	Spanish	Mean
`closed_max`	**0.816**	**0.963**	**0.983**	**0.921**
`explainable`	0.920	1.126	1.156	1.067
`traditional`	1.078	1.195	1.305	1.193
`expl.` – std. inference	0.961	1.151	1.190	1.101
`expl.` – lin. regression	0.975	1.154	1.202	1.111
Closed-Track Baseline	1.140	1.258	1.257	1.218

Table 3: RMSE of our closed-track submissions, compared with two variants of the `explainable` model and the shared task's closed-track baseline.

improves LLM predictions of lexical complexity (Nohejl et al., 2025a). The models with added features (`open_max` and `finetuned_llms_plus`) surpassed all other submissions in the open track on all three languages. The linear stack of LLMs (`finetuned_llms`) surpassed all other submissions on Chinese and German but was narrowly outperformed by submissions from the Glite team on Spanish. The RMSE of approximately 0.7 that our models achieve is relatively close to the statistical optimum of 0.277.

Linear stacking (`finetuned_llms`) results in a modest improvement over the average ensemble of the same models (≤32B LLM Average). The ensembling itself, however, improves performance more substantially, from an RMSE of 0.751–0.753 for individual models to 0.719 for the average ensemble. While increasing the model size obviously improves performance, it is worth noting that the smaller Ministral-3-14B model performs on par with the 32B Qwen2.5 and GLM-4 models.

In Appendix E, we compare all individual LLMs used in the ensembles above, as well as LLMs with 0.5B parameters and similarly sized MLMs, showing that LLMs and MLMs achieve comparable results at comparable model sizes.

5.2 Closed Track

Table 3 compares the results of our submissions to the closed-track baseline. Interestingly, our model with only traditional features performs on par with the encoder-based baseline, and our `explainable` model with extra LLM-based features outperforms the baseline by a relatively wide margin (mean RMSE of 1.218 vs. 1.067). The `closed_max` model achieves a further improvement (mean RMSE of 0.921) by adding a fine-tuned MLM as a feature.

Two variants applied to the explainable features—standard inference for prompt-based features instead of probability weighting, and linear regression instead of an XGBoost regressor—result in higher RMSE.

Method (Base Model)	Chinese	German	Spanish	Mean
Ours (Ministral-3-14B)	**0.681**	**0.781**	**0.799**	**0.753**
- single language	0.701	0.824	0.799	0.775
- out-of-language	0.874	0.901	0.999	0.925
- short prompt	0.697	0.790	0.808	0.765
- standard loss	0.762	0.841	0.892	0.832
- std. loss & inference	0.863	0.943	1.003	0.936

Table 4: RMSE of ablations of our LLM-based model.

Method (Base Model)	Chinese	German	Spanish	Mean
Ours (mmBERT-b)	0.921	**0.984**	**1.063**	**0.989**
- single language	**0.916**	1.044	1.070	1.010
- out-of-language	1.115	1.102	1.193	1.136
- short prompt	0.929	1.012	1.066	1.002
- standard loss	1.003	1.099	1.166	1.089
- std. loss & inference	1.064	1.144	1.192	1.133
Regression (XLMR-b)	1.222	1.276	1.373	1.290
Regression (XLMR-l)	1.048	1.144	1.192	1.128
Reg. (mmBERT-b)	1.000	1.105	1.153	1.086

Table 5: RMSE of ablations of our MLM-based model, compared with using a standard regression head and different base models (XLM-RoBERTa-base/large).

6 Analysis and Discussion

6.1 Ablation Analysis

We ablate the fine-tuned LLM and MLM that were used in our open-track submissions and the `closed_max` closed-track submission. For the open-track submission, we ablate only one of the three similarly performing LLMs in the ensemble, Ministral-3-14B as follows:

Single language. We fine-tune the model on single L1 data instead of all three languages.

Out-of-language. We fine-tune the model on data for the other two L1s (e.g. the model tested on Chinese is fine-tuned on German and Spanish).

Short prompt. We use a minimalistic prompt template instead of the basic verbose one. See Section A.1.

Standard loss. Instead of using the cross-entropy loss with a soft target, described in Section 3.1, we fine-tune using the standard cross-entropy loss with the discretized score following Smădu et al. (2024).

Standard loss and inference. We fine-tune using the standard cross-entropy loss with a discretized

score, and instead of using probability-weighting at inference, we decode the highest probability token.

In addition to performing the same ablations for the MLM, we also compare the results of our method to those of using a regression head with MSE loss on mmBERT-base and two sizes of XLM-RoBERTa models (listed as **Regression**). Note that in the closed track, rules required us to use the variant we report here as an ablation (mmBERT-b/single language).

As shown in Table 4 and Table 5, our method outperforms all ablations with the single exception of the MLM model fine-tuned only on Chinese L1 data, which performs better on Chinese by a small margin. In the other cases, training on a single language results in only a small performance drop.

The standard loss ablation demonstrates that our technique of fine-tuning using cross-entropy loss with soft targets is superior to common approaches for both LLMs and MLMs. Fine-tuning using the standard loss results in a tangible performance drop for both model types (a difference in mean RMSE of 0.079 and 0.900, respectively). For the MLM, the common approach of fine-tuning a regression head with MSE loss degrades performance similarly (by a difference in mean RMSE of 0.809).

6.2 The Explainable Models

As outlined in Section 3.2, we designed a feature-based model with traditional and LLM-based features that all have a clear interpretation. SHAP values explain the model's output locally in terms of the impact of each feature, which has both sign and magnitude. For a global analysis of feature importance, we report the mean absolute SHAP values computed over the test data in Figure 1.

Production frequency is a consistently high-impact feature across languages. Its importance per language, in fact, corresponds with the amount of L1-specific data available. Yet, even for German, with no L1-specific data available, the feature almost completely overshadows **reception frequency**, which ranks as the least important across L1s. This is in line with the focus of the KVL data on production and can be contrasted with findings for the comprehension-focused LCP, where reception frequency (represented by frequency in TUBELEX) and production frequency (represented by frequency in Lang-8) are equally good predictors (Nohejl et al., 2024)

Spelling difficulty is the most important feature for L1 German and the second most important

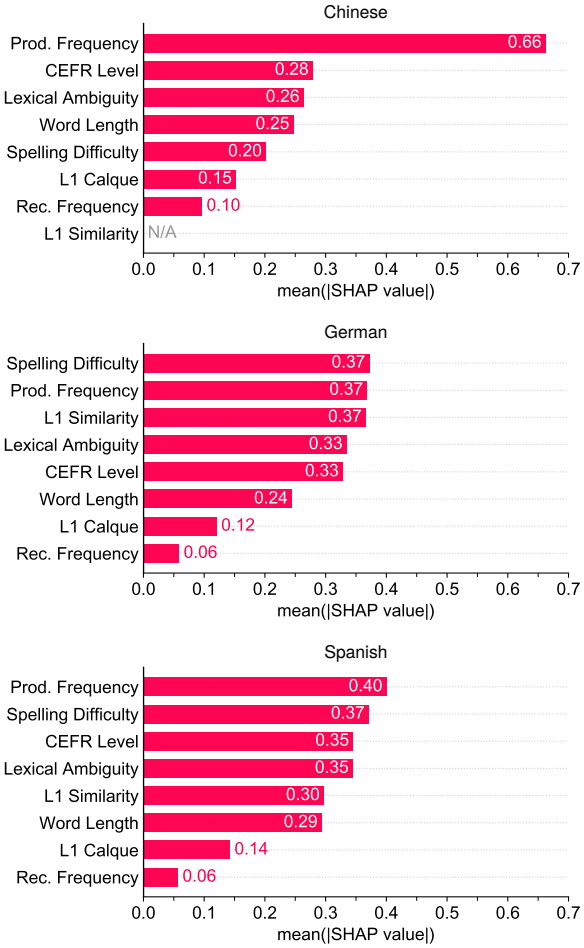

Figure 1: Global SHAP summaries by L1.

feature for Spanish. Perhaps counter-intuitively, it is much less important for L1 Chinese. We hypothesize this is caused by two factors. First, the production frequency uses learner-written texts, and therefore it partially discounts the frequency of words with frequent mistakes. As a result, the importance of the separate spelling difficulty feature is lower proportionally to the size of the L1-specific written production data. Second, for Chinese speakers, there may be less interference from the orthography of their L1, which is primarily written in Chinese characters, while Spanish and German speakers may be more prone to making errors due to the influence of their L1's spelling (e.g., English *music*, *action*; German *Musik*, *Aktion*).

CEFR level ranks particularly high for Chinese L1 learners. This corresponds to a very small proportion of cognates between English and Chinese compared to the two European languages, where **L1 Similarity** can play a comparable (Spanish) or more important role (German).

Lexical ambiguity has a similarly high impact as

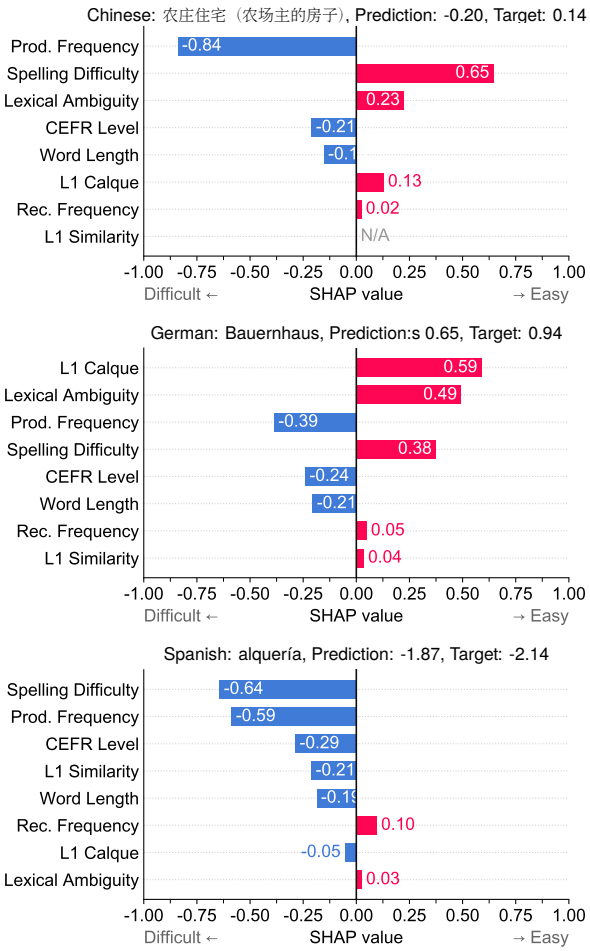

Figure 2: Example of local SHAP explanations by L1 for the English word *farmhouse*.

the CEFR level and can be seen as complementary to the CEFR level and the frequency features, all of which focus on the surface lemma (frequency of all senses or homonyms, the lowest CEFR level, regardless of sense). We recognize words as lexically ambiguous if the test item refers to a less common sense or a homonym.

Word length is surprisingly one of the consistently relevant features. While length certainly contributes to the difficulty in acquiring a word, it also simply increases the probability of typos, even in cases when the writer knows the word's spelling.

L1 calque is one of the globally least important features, reflecting both the frequency of calques in the language pairs and the fact that an L1 word being a calque (morpheme-for-morpheme translation) being less conducive to eliciting the production of the correct English word than the surface level L1 similarity.

Local explanations of all test set predictions of the explainable model can be viewed in an online application.[3] In Figure 2 we provide an example of the local explanation of the difficulty of the English word *farmhouse*, which, despite being relatively rare and not being assigned any CEFR level in the EVP, is not a particularly difficult word for L1 Chinese and German learners. Both its Chinese and German counterparts are calques of the English word *farmhouse*, which contributes to lower difficulty in these languages. On the other hand, for L1 Spanish speakers, the word is assessed as comparatively difficult to spell.

Neither the global nor the local explanations can be interpreted as the actual causes of the difficulty of a particular word. In particular, it cannot tell us anything about features not present in the model, and it underestimates the importance of a factor when the data has low quality (e.g., production frequency in the case of L1 German). On the other hand, given the relatively high accuracy of the model, we believe that the globally high-impact features provide valuable insights.

Both the high impact of production frequency and spelling difficulty is in line with the focus of the KVL. However, the combined importance of features reflecting the difficulty of spelling a word without errors (spelling difficulty, word length, and, to some extent, also the learners' written production frequency) could affect the utility of the dataset for other uses, such as creating reading materials matched to the learner's level. (Schmitt et al., 2024) suggest the KVL data could be more suitable for this purpose than frequency.

Given that the KVL are designed to focus on the most common word sense (Schmitt et al., 2024), the high importance of lexical ambiguity in our analysis could seem spurious. We have, however, observed uncommon word senses in the data. For instance, the noun *wireless* is included in the sense 'radio (receiving set)', which is dated and mainly British according to the Oxford Dictionary of English, rather than in the sense 'wireless broadcasting or networking using radio signals'. For other words, such as *log* ('chunk of wood' in KVL, not 'record') or *diet* ('nutrition' in KVL, not 'dietary regime'), two senses may be similarly common. Lexical ambiguity provides a useful signal in all these cases.

[3] https://ynklab.github.io/vocabulary-difficulty/

English word	Spanish word	Top LLM response
instantly	inmediatamente	**i**mmediately*
motivational	estimulante	**m**otivating*
(to) **r**eform	enmendar	**r**evise
(to) **a**mount	sumar	**a**dd up
everybody	todo el mundo	**e**veryone*
synonymous	sinónimo	**s**ynonym*

Table 6: Examples of tricky test items in L1 Spanish, based on responses of GPT-4.1-mini. First letter printed in bold. Asterisks mark incorrect word length.

6.3 Tricky Test Items

The shared task rules permitted us to use the trickiness feature only in the open-track submission, where its impact was relatively small. However, it provides an interesting insight into how the test item construction often affects the resulting difficulty scores. The value of our trickiness feature is the probability of an LLM answering the test item incorrectly. Table 6 shows examples of high-trickiness items from the test set. For brevity, we focus on cases that do not depend on the L1 context. As exemplified by the items *instantly*, *motivational*, and *reform*, in many cases, the creators of the KVL test items intentionally avoided cognate L1 words. The words they have chosen instead often correspond better to English words other than the intended ones. In other cases, the reason is not the avoidance of cognates, but the L1 word strongly suggests a different English word nonetheless. While in some cases the L1 context or the number of letters disambiguates the English word (e.g., *sinónimo* is used as an adjective, not a noun; *everybody* and *everyone* differ in word length), such details were likely easy to miss not only for the LLM but also for the respondents. We believe that while better choices could have been made for some items, this shows an inherent limitation of the test item format used to compile the KVL.

7 Conclusion

We proposed a novel method for fine-tuning LLMs and MLMs to predict continuous values using cross-entropy loss with soft targets. The method achieved consistent gains on the vocabulary difficulty prediction task over prior approaches, such as discretization for LLMs and a regression head with MSE loss for MLMs.

In our fine-tuning experiments, increases in base model size up to 32B parameters resulted in performance improvements. The larger-sized LLMs also consistently outperformed the smaller-sized MLMs. This result differs from the LCP study by Smădu et al. (2024), where much smaller fine-tuned MLMs performed comparably to much larger LLMs. We hypothesize that this could be due to the more complex nature of vocabulary difficulty prediction compared to LCP.

We built a separate explainable model that achieves competitive results on the task and provides insights into what affects vocabulary difficulty scores in the British Council's KVL data. The high impact of spelling difficulty and the construction of the test items call for further investigation.

Limitations

We tested the proposed method for fine-tuning LLMs and MLMs to predict continuous values using cross-entropy loss with soft-targets only on this shared task's data (KVL). Its suitability for other tasks and settings is yet to be investigated. On smaller or noisier data, for instance, its impact may be less pronounced. Various parameters, such as the number of points on the scale used, may also affect its performance.

The insights provided by our explainable model and the trickiness model can, in principle, be empirically validated, but we have not been able to do so, as the individual responses used to compile KVL are not publicly available. The frequency of spelling mistakes would be particularly easy to validate with access to the responses.

References

Dennis Aumiller and Michael Gertz. 2022. UniHD at TSAR-2022 shared task: Is compute all we need for lexical simplification? In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 251–258, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

BNC Consortium. 2007. British National Corpus, XML edition. https://llds.ling-phil.ox.ac.uk/llds/xmlui/handle/20.500.14106/2554.

Annette Capel. 2012. Completing the English Vocabulary Profile: C1 and C2 vocabulary. *English Profile Journal*, 3:e1.

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*, KDD '16, pages 785–794, New York, NY, USA. Association for Computing Machinery.

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.

DeepSeek-AI. 2025. DeepSeek-V3 Technical Report. *ArXiv preprint*, arXiv:2412.19437v2 [cs.CL].

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *Advances in Neural Information Processing Systems*, 36:10088–10115.

Taisei Enomoto, Hwichan Kim, Tosho Hirasawa, Yoshinari Nagai, Ayako Sato, Kyotaro Nakajima, and Mamoru Komachi. 2024. TMU-HIT at MLSP 2024: How well can GPT-4 tackle multilingual lexical simplification? In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 590–598, Mexico City, Mexico. Association for Computational Linguistics.

Mariano Felice and Lucy Skidmore. 2026. Findings of the BEA 2026 shared task on vocabulary difficulty prediction for English learners. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, San Diego, California. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Yusuke Ide, Masato Mita, Adam Nohejl, Hiroki Ouchi, and Taro Watanabe. 2023. Japanese lexical complexity for non-native readers: A new dataset. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 477–487, Toronto, Canada. Association for Computational Linguistics.

John H.A.L. Jong, Mike Mayor, and Catherine Hayes. 2016. Developing global scale of English learning objectives aligned to the common European framework. Technical report.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. mmBERT: A Modern Multilingual Encoder with Annealed Language Learning. *ArXiv preprint*, arXiv:2509.06888v1 [cs].

Mistral AI. 2026. Ministral 3. *ArXiv preprint*, arXiv:2601.08584v1 [cs.CL].

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Adam Nohejl, Akio Hayakawa, Yusuke Ide, and Taro Watanabe. 2024. Difficult for whom? a study of Japanese lexical complexity. In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 69–81, Miami, Florida, USA. Association for Computational Linguistics.

Adam Nohejl, Akio Hayakawa, Yusuke Ide, and Taro Watanabe. 2025a. A Japanese Dataset and Efficient Multilingual LLM-Based Methods for Lexical Simplification and Lexical Complexity Prediction. *Journal of Natural Language Processing*, 32(4):1129–1188.

Adam Nohejl, Frederikus Hudi, Eunike Andriani Kardinata, Shintaro Ozaki, Maria Angelica Riera Machin, Hongyu Sun, Justin Vasselli, and Taro Watanabe. 2025b. Beyond film subtitles: Is YouTube the best approximation of spoken vocabulary? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9566–9585, Abu Dhabi, UAE. Association for Computational Linguistics.

OpenAI. 2024. GPT-4 Technical Report. *ArXiv preprint*, arXiv:2303.08774v6 [cs.CL].

OpenAI. 2025. Update to GPT-5 System Card: GPT-5.2. Technical report.

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.

Qwen Team. 2025. Qwen2.5 Technical Report. *ArXiv preprint*, arXiv:2412.15115v2 [cs.CL].

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*, volume arXiv:1910.01108.

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

Norbert Schmitt, Karen Dunn, Barry O'Sullivan, Laurence Anthony, and Benjamin Kremmel. 2024. *Knowledge-Based Vocabulary Lists*. British Council Monographs on Modern Language Testing. University of Toronto Press.

Graham G. Scott, Anne Keitel, Marc Becirspahic, Bo Yao, and Sara C. Sereno. 2019. The Glasgow Norms: Ratings of 5,500 words on nine scales. *Behavior Research Methods*, 51(3):1258–1270.

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.

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.

Răzvan-Alexandru Smădu, David-Gabriel Ion, Dumitru-Clementin Cercel, Florin Pop, and Mihaela-Claudia Cercel. 2024. Investigating large language models for complex word identification in multilingual and multidomain setups. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16764–16800, Miami, Florida, USA. Association for Computational Linguistics.

Team GLM. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *ArXiv preprint*, arXiv:2406.12793v2 [cs.CL].

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.

A Prompt Templates

In the following, we list the full text of our prompt templates by feature for which they were used. The symbol ↵ stands for line breaks. Placeholders are printed in bold. The following common placeholders corresponding to the dataset items are used across multiple prompts:

- `{l1_name}`: an L1, i.e., `Spanish`, `Chinese`, or `German`.
- `{l1_word}`: an L1 word.
- `{l1_context}`: an L1 context.
- `{en_word}`: an English word.
- `{clue}`: a letter-pattern clue derived from the English word (first letter + space-separated underscores), e.g., `b _ _ _` for `book`.

A.1 Fine-Tuning

Basic template:

```
Rate how difficult it is for learners to guess
the English word based on the {l1_name} word,
context and clue on a scale from 1 to 5 (1=very
easy, 5=very difficult).↵
{l1_name} word: {l1_word}↵
{l1_name} context: {l1_context}↵
Clue: {clue}↵
English word: {en_word}↵
Difficulty:
```

Short prompt template:

Used in the "short prompt" ablation.

```
{l1_word} ### {l1_context} ### {clue} ### {en_word} ### Difficulty (1 to 5):
```

Regression template

Used for MLMs when fine-tuning with a regression head, following Skidmore et al. (2025).

```
[CLS] {l1_word} [SEP] {l1_context} [SEP] {clue} [SEP] {en_word} [SEP]
```

A.2 Lexical Ambiguity

Placeholders:

We use the English word "bank" as an example of lexical ambiguity ('financial institution' vs. '(river) side'). For L1 Spanish, the placeholders would take the following values:

- {ex_en_word}: "bank"
- {ex_easy_word_l1}: "banco"
- {ex_easy_context_l1}: "Deposité el dinero en el banco."
- {ex_hard_word_l1}: "orilla"
- {ex_hard_context_l1}: "Nos sentamos en la orilla del río."

Prompt template:

```
You are a language education expert.↵
TASK↵
Given:↵
- an English word form (the "English word"),↵
- an L1 gloss/translation (the "{l1_name}
item"),↵
- and the L1 usage context sentence (the
"{l1_name} context"),↵
decide whether the English word, when used to
express the meaning suggested by the L1 item +
context,↵
meets BOTH conditions:↵
A) Lexical ambiguity: the English word has
multiple established senses that share the same
form↵
   (polysemy or homonymy), such that another
common sense could plausibly be
activated/confused.↵
B) Unfamiliarity for L2 learners: in this
meaning/usage, the English word is likely to be
unfamiliar↵
   or challenging for typical second-language
learners (e.g., less frequent sense,
idiomatic/figurative,↵
   domain-specific usage, nonliteral
extension).↵
OUTPUT REQUIREMENTS↵
- Output "1" if BOTH conditions (A and B) are
met; otherwise output "0".↵
- Output MUST be exactly one character: 1 or 0.↵
- Do NOT include explanations, alternatives,
quotes, or extra text.↵
EXAMPLE 1↵
English word: {ex_en_word}↵
{l1_name} item: {ex_easy_word_l1}↵
{l1_name} context: {ex_easy_context_l1}↵
Is the English word ambiguous and unfamiliar: 0↵
EXAMPLE 2↵
English word: {ex_en_word}↵
{l1_name} item: {ex_hard_word_l1}↵
{l1_name} context: {ex_hard_context_l1}↵
Is the English word ambiguous and unfamiliar: 1↵
NOW DECIDE↵
English word: {en_word}↵
{l1_name} item: {l1_word}↵
{l1_name} context: {l1_context}↵
Is the English word ambiguous and unfamiliar:
```

A.3 Spelling Difficulty

This prompt is executed for all L1s at once.

Placeholders:

- {en_pron}: CMU-style pronunciation string of the English word of the current item. Example: K Y UW for "queue".
- {all_l1_words[l1]}: L1 words of the current item (all three in one prompt).
- {hard_pron}, {hard_cn}, {hard_es}, {hard_de}, {hard_cn_score}, {hard_es_score}, {hard_de_score}: hard demonstration item (pronunciation, three L1 translations, and example scores 5, 4, 4).
- {easy_pron}, {easy_cn}, {easy_es}, {easy_de}, {easy_cn_score}, {easy_es_score}, {easy_de_score}: easy demonstration item (pronunciation, three L1 translations, and example scores 1, 1, 1).

Prompt template:

```
TASK↵
You are required to rate English spelling
difficulty on a 1–5 scale, where 1 = very easy
and 5 = very difficult.↵
You will be given English pronunciation and the
target word's translation in Chinese, Spanish,
and German.↵
Evaluate how difficult it would be for learners
with Chinese, Spanish, and German L1 backgrounds
to spell the English word with that pronunciation
correctly when they know the translation in their
native language.↵
OUTPUT REQUIREMENTS↵
- Output exactly one digit (1, 2, 3, 4, or 5) for
each L1, separated by commas, in the order of
Chinese, Spanish, German.↵
- Do not include any other text.↵
EXAMPLE 1↵
English pronunciation: '{hard_pron}'↵
Chinese: {hard_cn}↵
Spanish: {hard_es}↵
German: {hard_de}↵
Result:
{hard_cn_score},{hard_es_score},{hard_de_score}↵
EXAMPLE 2↵
English pronunciation: '{easy_pron}'↵
Chinese: {easy_cn}↵
Spanish: {easy_es}↵
German: {easy_de}↵
Result:
{easy_cn_score},{easy_es_score},{easy_de_score}↵
NOW DECIDE↵
English pronunciation: {en_pron}↵
Chinese: {all_l1_words['cn']}↵
Spanish: {all_l1_words['es']}↵
German: {all_l1_words['de']}↵
Result:
```

A.4 L1 Calque

This prompt uses Spanish examples for all L1s.

Prompt template:

```
You are a linguist and your task is to decide
whether an English word is a
morpheme-for-morpheme translation of any of the
given {l1_name} equivalents.↵
The morpheme-for-morpheme mapping must be 1:1.
1:N or other mappings do not count.↵
Single morpheme translations or simple
borrowings/cognates do not count either.↵
Respond only with YES or NO.↵
wave/ola: NO (reason: single morpheme)↵
ecosystem/ecosistema: NO (reason: simple
cognate)↵
hotdog/perro caliente: YES (reason:
hot=caliente, dog=perro)↵
stare/mirar fijamente: NO (reason: not a 1:1
mapping)↵
{en_word}/{l1_word}:
```

A.5 L1 Calque (used only in `closed_max`)

This was our first iteration of the prompt. It did not give the expected results (monomorphemic word pairs were labeled as calques), so we did not use it for our explainable model. However, as it performed well as a feature, we included it in the `closed_max` model.

Placeholders:

- `{ex_calque_l1}`: the L1-side word, e.g., "热狗" for Chinese, composed of the morphemes 'hot' and 'dog'.
- `{ex_calque_en}`: the English-side word, e.g., "hot dog".

Prompt template:

```
You are a bilinguistics expert.↵
TASK↵
Given a {l1_name} item and an English item,
decide whether there exists a best-matching
candidate in the {l1_name} item that is a
component-by-component (morpheme-level)
translation of the English item.↵
A component-by-component mapping means that the
meaningful parts↵
(words, roots, prefixes, or suffixes) of the
English item are directly translated↵
into corresponding meaningful parts in the
{l1_name} item.↵
Procedure (internal; do NOT output these
steps):↵
1) If the {l1_name} item contains multiple
candidates, select exactly ONE candidate: the
one that aligns best component-wise with the
English form.↵
2) Judge ONLY that selected candidate for
component-by-component mapping.↵
OUTPUT REQUIREMENTS↵
- Output "1" if the selected best candidate is a
component-by-component mapping; otherwise output
"0".↵
- Output MUST be exactly one character: 1 or 0.↵
- Do NOT include explanations, alternatives,
quotes, or extra text.↵
EXAMPLE↵
{l1_name} item: {ex_calque_l1}↵
English item: {ex_calque_en}↵
Is word-for-word mapping: 1↵
NOW DECIDE↵
{l1_name} item: {l1_word}↵
English item: {en_word}↵
Is word-for-word mapping:
```

A.6 Trickiness (short prompt, used only in `open_max`)

Placeholders:

`{solve_example}`: a formatted one-shot example (L1 word, L1 context, English word) using the item English word "strawberry'. Example:

```
German word: Erdbeere↵
German context: Ich mag keine Erdbeeren.↵
English word: strawberry
```

Prompt template:

```
You are bilingual in {l1_name} and English and
your task is to find the best English translation
for a {l1_name} word given a context and
constraints. The constraints are given in the
form of a clue, e.g., "b _ _ _", meaning that the
word starts with the (upper or lower case) letter
B and has 4 letters. You must give a single
English word in dictionary form (lemma) as a
response.↵
{solve_example}↵
{l1_name} word: {l1_word}↵
{l1_name} context: {l1_context}↵
Clue: {clue}↵
English word:
```

A.7 Trickiness (long prompt, used only in `open_max`)

Placeholders:

Same as in Section A.6.

Prompt template:

```
You are bilingual in {l1_name} and English.↵
TASK↵
Given a word in {l1_name}, its usage context, and
a spelling clue, find the single best English
translation that fits BOTH the meaning and the
spelling constraint.↵
INPUTS↵
- {l1_name} word: a single word to translate↵
- {l1_name} context: a sentence showing how the
word is used↵
- Clue: a pattern such as "b _ _ _", where:↵
  * the first letter is indicated
(case-insensitive)↵
  * "_" indicates subsequent unknown letter↵
  * the total number of letters must match
exactly↵
OUTPUT REQUIREMENTS↵
- Output EXACTLY ONE English word↵
- The word must be:↵
  * a dictionary form (lemma)↵
```

```
  * a single token (no spaces, hyphens, or
punctuation)↵
  * consistent with the context↵
  * consistent with the clue↵
- Do NOT include explanations, alternatives,
quotes, or extra text.↵
EXAMPLES↵
{solve_example}↵
NOW SOLVE↵
{l1_name} word: {l1_word}↵
{l1_name} context: {l1_context}↵
Clue: {clue}↵
English word:
```

A.8 Difficulty

Placeholders:

`{examples}`: a block of examples for difficulty rating using examples from the training data. Examples with ratings converted to values close to 1, 3, and 5 are selected for each L1 separately.

Prompt template:

```
You are an English language teacher teaching
learners whose native language is {l1_name}.
Your task is to rate the difficulty of a
vocabulary test item for native {l1_name}
speakers learning English.↵
The test item consists of:↵
- a {l1_name} word,↵
- a {l1_name} context,↵
- a clue indicating the first letter and word
length of the English word,↵
- the target English word, which is the only
correct answer.↵
Letter case does not matter. The learners are
likely to respond with synonyms or misspellings
to some items, but such responses are considered
incorrect. Treat this as increasing the
difficulty.↵
Consider learners from beginner to advanced
levels, weighting the intermediate learner most
heavily. Rate how difficult the item is on a
scale from 1 to 5:↵
1 = very easy (almost everybody answers
correctly)↵
5 = very difficult (almost nobody answers
correctly)↵
Output exactly one digit (1, 2, 3, 4, or 5). Do
not include any other text.↵
{examples}↵
{l1_name} word: {l1_word}↵
{l1_name} context: {l1_context}↵
Clue: {clue}↵
English word: {en_word}↵
Difficulty:
```

B Base Models and API Models

Table 7 lists base LLMs and MLMs that we fine-tuned in our experiments. Table 8 lists LLMs that we used for prompt-based features via API.

Model Name	Hugging Face Model ID	Systems/References to Results
GLM-4-32B	`zai-org/GLM-4-32B-Base-0414`	`finetuned_llms`*
Qwen2.5-32B	`Qwen/Qwen2.5-32B`	`finetuned_llms`*
Ministral-3-14B	`mistralai/Ministral-3-14B-Base-2512`	`finetuned_llms`*; ≤14B LLM Average; ablations in Table 4
Qwen2.5-14B	`Qwen/Qwen2.5-14B`	≤14B LLM Average
GLM-4-9B	`zai-org/glm-4-9b`	≤14B LLM Average; ≤9B LLM Average
Qwen2.5-7B	`Qwen/Qwen2.5-7B`	≤9B LLM Average
Ministral-3-8B	`mistralai/Ministral-3-8B-Base-2512`	≤9B LLM Average
Qwen2.5-1.5B	`Qwen/Qwen2.5-1.5B`	model size comparison in Appendix E
Qwen2.5-0.5B	`Qwen/Qwen2.5-0.5B`	model size comparison in Appendix E
mmBERT-b (mmBERT-base)	`jhu-clsp/mmBERT-base`	`closed_max`; regression and ablations in Table 3
XLMR-b (XLM-RoBERTa-base)	`xlm-roberta-base`	baselines; regression in Table 3
XLMR-l (XLM-RoBERTa-large)	`xlm-roberta-large`	regression in Table 3

*The three models used in `finetuned_llms`, were also used in the `finetuned_llms_plus` and `open_max` submissions, and the ≤32B LLM Average ensemble.

Table 7: Open-weight base models used in fine-tuning experiments.

Model Name	Provider	Model ID (Snapshot)	Explainable Features (`explainable`; `open_max`)	Additional Features (`open_max`)
GPT-5.2	OpenAI	`gpt-5.2-2025-12-11`	Lexical ambiguity; Spelling difficulty; L1 calque	Difficulty
GPT-4.1	OpenAI	`gpt-4.1-2025-04-14`		Trickiness; Difficulty
GPT-4.1-mini	OpenAI	`gpt-4.1-mini-2025-04-14`		Trickiness
GPT-4.1-nano	OpenAI	`gpt-4.1-nano-2025-04-14`		Trickiness
DeepSeek-V3.2	DeepSeek	`deepseek-chat`	Lexical ambiguity	

Table 8: Models used for features based on LLM prompting.

C Hyperparameters

Table 9 shows the general fine-tuning hyperparameters. For XLM-RoBERTa, we followed Skidmore et al. (2025). For other models, we performed a limited search for the learning rate and epochs using cross-validation. For mmBERT, a higher learning rate would likely be optimal, but 3e-5 was the largest we evaluated. For LLMs, we found that the learning rate of 1e-4 performed better than 1.5e-4, but we could not rerun the training for Qwen2.5-32B in time for the final submission. Table 10 lists the quantization and LoRA hyperparameters used for all LLM fine-tuning.

For the XGBoost regressor (`XGBRegressor` in XGBoost's Scikit-Learn API[4]), we set `max_depth=3`, `learning_rate=0.1`, and `n_estimators=200`, while using default values for other hyperparameters.

Model	Epochs	Batch size	Grad. accum.	Learning rate	Weight decay	Warmup ratio
XLM-RoBERTa (both)	5	32	—	3e-5 (linear)	0.1	0.1
mmBERT-base	16	16	—	3e-5 (const.)	0.1	0.1
Qwen2.5-32B	4	2	8	1.5e-4 (const.)	—	—
All other LLMs	4	2	8	1e-4 (const.)	—	—

Table 9: General fine-tuning hyperparameters used for different models.

[4] https://xgboost.readthedocs.io/en/release_3.1.0/python/python_api.html#xgboost.XGBRegressor

Group	Hyperparameter	Value
Quantization	Bit-width	4-bit
	Data type	NFloat4
	Double quant.	Yes
QLoRA-related	Compute type	BFloat16
	Optimizer	`paged_adamw_8bit`
LoRA adapter	Rank r	8
	Scaling α	16
	Dropout rate	0.05
	Target modules	All linear
	Bias adaptation	No

Table 10: Quantization and LoRA hyperparameters used for all LLM fine-tuning experiments.

D Results Reported as Pearson's Correlation Coefficients

Table 11, Table 12, Table 13, and Table 14 show Pearson's correlation coefficient (PCC) evaluation corresponding to the RMSE evaluation shown in the main text in Tables 2 to 5.

E Comparison of Models Across Sizes

In Table 15 and Table 16, we compare all base models we used in experiments, including models that are only reported in ensembles in the main text. To facilitate the comparison of models of different architectures (decoder LLMs and encoder MLMs), we also add Qwen models of sizes 1.5B and 0.5B, as well as XLM-RoBERTa models fine-tuned with the same prompt and method. We can observe that the model size, rather than the architecture, determines performance in this task. At the same size, however, MLMs are more compute-efficient.

All models were fine-tuned using the same cross-entropy loss with soft targets and the same prompt, utilizing hyperparameters listed in Appendix C.

F Simulation of a Statistical Optimum

The difficulty scores provided by the KVL data are based on test item responses. The number of responses collected for each item (120 to 228 responses) determines the precision of the data, which can be expressed as a confidence interval. Schmitt et al. (2024, Sec. 6.3, App 5) provide 83% confidence intervals for selected bands of the complete KVL data (e.g., items ranked 200–299 for L1=Spanish) expressed in numbers of ranks (e.g., 44). The rationale for 83% is that it corresponds to a significance level of 5% for differences in pairwise ordering (e.g., there is no significant difference

System	Chinese	German	Spanish	Mean
`open_max`	0.927	**0.916**	**0.920**	**0.921**
`finetuned_llms_plus`	**0.928**	0.915	0.919	0.921
`finetuned_llms`	0.925	0.914	0.915	0.918
≤32B LLM Average	0.926	0.914	0.916	0.918
- GLM-4-32B	0.918	0.907	0.906	0.910
- Qwen2.5-32B	0.917	0.902	0.907	0.909
- Ministral-3-14B	0.915	0.902	0.907	0.908
≤14B LLM Average	0.921	0.905	0.908	0.911
≤9B LLM Average	0.916	0.901	0.900	0.906
Open-Track Baseline	0.804	0.786	0.783	0.791
Statistical Optimum	0.989	0.990	0.995	0.991

Table 11: PCC of our open-track submissions, compared with average ensembles by model size, individual models, and the shared task's open-track baseline.

System	Chinese	German	Spanish	Mean
`closed_max`	**0.874**	**0.844**	**0.854**	**0.857**
`explainable`	0.837	0.779	0.789	0.802
`traditional`	0.767	0.747	0.721	0.745
`exp.`: std. inference	0.820	0.768	0.776	0.788
`exp.`: lin. regression	0.815	0.766	0.769	0.783
Closed-Track Baseline	0.753	0.773	0.765	0.764

Table 12: PCC of our closed-track submissions, compared with two variants of the `explainable` model and the shared task's closed-track baseline.

Method (Base Model)	Chinese	German	Spanish	Mean
Ours (Ministral-3-14B)	**0.915**	**0.902**	0.907	**0.908**
- single language	0.909	0.890	**0.907**	0.902
- out-of-language	0.862	0.879	0.876	0.872
- short prompt	0.911	0.899	0.906	0.905
- standard loss	0.892	0.885	0.886	0.887
- std. loss & inference	0.859	0.853	0.850	0.854

Table 13: PCC of ablations of our LLM-based model.

in difficulty between items within ranks 200 and 44). The widest confidence intervals are 69, 95, and 108 ranks for Spanish, Chinese, and German, respectively. While the span of 100 ranks is given as a rule-of-thumb criterion for "strong confidence", we use the above per-L1 maximum widths w for our simulation.

We simulate our statistical optimum predictions by taking the most distant difficulty in the complete KVL Data within $\pm w$ ranks of the predicted data point and using it as a prediction. In accordance with the originally reported confidence intervals, this process is based on the complete KVL data (training, development, and test subsets), although we predict only for the test data. Such predictions could be considered to have no statistically

Method (Base Model)	Chinese	German	Spanish	Mean
Ours (mmBERT-b)	0.837	**0.837**	**0.826**	**0.833**
- single language	**0.839**	0.816	0.824	0.826
- out-of-language	1.115	1.102	1.193	1.136
- short prompt	0.834	0.827	0.825	0.828
- standard loss	0.807	0.794	0.788	0.796
- std. loss & inference	0.779	0.772	0.776	0.776
Regression (XLMR-b)	0.710	0.729	0.693	0.711
Regression (XLMR-l)	0.813	0.823	0.801	0.812
Reg. (mmBERT-b)	0.812	0.798	0.797	0.802

Table 14: PCC of ablations of our MLM-based model, compared with using a standard regression head and different base models (XLM-RoBERTa base/large).

Base Model	Chinese	German	Spanish	Mean
GLM-4-32B	**0.678**	**0.769**	0.805	**0.751**
Qwen2.5-32B	0.678	0.777	0.799	0.752
Ministral-3-14B	0.681	0.781	**0.799**	0.753
Qwen2.5-14B	0.701	0.801	0.832	0.778
GLM-4-9B	0.744	0.858	0.930	0.844
Ministral-3-8B	0.723	0.809	0.835	0.789
Qwen2.5-7B	0.722	0.842	0.882	0.816
Qwen2.5-1.5B	0.800	0.955	0.998	0.918
Qwen2.5-0.5B	0.851	1.021	1.078	0.983
XLMR-l (550B)	0.924	1.001	1.095	1.007
mmBERT-b (307M)	0.921	0.984	1.063	0.989
XLMR-b (270B)	1.042	1.136	1.220	1.133

Table 15: RMSE of all individual base LLMs and MLMs.

Base Model	Chinese	German	Spanish	Mean
GLM-4-32B	**0.918**	**0.907**	0.906	**0.910**
Qwen2.5-32B	0.917	0.902	**0.907**	0.909
Ministral-3-14B	0.915	0.902	0.907	0.908
Qwen2.5-14B	0.909	0.895	0.897	0.900
GLM-4-9B	0.900	0.880	0.874	0.885
Ministral-3-8B	0.904	0.895	0.898	0.899
Qwen2.5-7B	0.904	0.884	0.884	0.890
Qwen2.5-1.5B	0.883	0.848	0.848	0.860
Qwen2.5-0.5B	0.863	0.824	0.820	0.836
XLMR-l (550B)	0.835	0.832	0.816	0.827
mmBERT-b (307M)	0.837	0.837	0.826	0.833
XLMR-b (270B)	0.786	0.775	0.761	0.774

Table 16: PCC of all individual base LLMs and MLMs.

significant difference from the gold standard data; hence, we report their RMSE and correlation as a "statistical optimum".

G Predicting in the Probability Space

As shown in Table 17, fine-tuning an LLM to predict in the probability space instead of the logit space, in which the GLMM scores are, resulted in a decrease in performance. We therefore used the raw GLMM scores for the main experiments.

	↓ RMSE				↑ PCC			
Target Values	Chinese	German	Spanish	Mean	Chinese	German	Spanish	Mean
Logits: GLMM score	**0.681**	**0.781**	**0.799**	**0.753**	**0.915**	**0.902**	**0.907**	**0.908**
Probabilities: expit(GLMM score)	0.738	0.853	0.877	0.822	0.905	0.889	0.895	0.896

Table 17: Performance of Ministral-3-14B when fine-tuned on the raw GLMM score (logits), as we did in all experiments, compared with the same model fine-tuned on the corresponding probabilities. In both cases, the values were linearly transformed to the 1-to-5 difficulty scale we use in our prompt, and all other parameters were the same. RMSE and PCC were measured for GLMM score values in the logit space for both.