Multi-component student writing profiles for expert-aligned automated evaluation of English learner essays.

Russell Moore
ALTA Institute
Computer Laboratory
University of Cambridge
rjm49@cam.ac.uk

Andrew Caines
ALTA Institute
Computer Laboratory
University of Cambridge
apc38@cam.ac.uk

Paula Buttery
ALTA Institute
Computer Laboratory
University of Cambridge
pjb48@cam.ac.uk

Abstract

Automated Writing Evaluation (AWE) platforms have become common, but a significant gap remains between automated assessment and expert human feedback. We address this gap by introducing a supervised learning method that uses a multi-component student writing profile (comprising estimated CEFR level, grammatical error rates, and vocabulary distribution) to align AI scoring with expert human judgements. In the context of an online essay-writing platform for second language learners of English, our model achieves a 36% reduction in RMSE for holistic essay scoring and an 84% improvement in similarity to human-expert analysis of grammatical errors compared to automarker scores (26% and 57% improvement from the best-performing comparable earlier work, by Zaidi et al., 2019. Furthermore, we demonstrate that the model can predict a student's profile for a final-version essay from earlier drafts and that predictions generalise to a subsequent task, offering new possibilities for automated curriculum planning. Finally, we introduce a visualisation tool that provides educators with clear expert-aligned longitudinal views of student development.

1 Introduction

Automated Writing Evaluation (AWE) platforms typically allow learners to submit a sequence of revised essays which incorporate system-generated suggestions for improving spelling, vocabulary and grammar. However, these platforms fail to provide the depth of feedback found in human instruction.

Standard metrics for assessing essay quality (*e.g.*, TTR, MTLD, error-free-clause-ratio, grammatical error density, Flesch-Kincaid grade level) offer coarse-grained statistics, but they sacrifice nuance for summarisation. In this work we move beyond such metrics by representing texts as a richer *writing profile* object. This profile comprises subcomponents for holistic proficiency, grammatical error rates and vocabulary use; this provides a more granular view of student development and captures elements that the student *can do* (vocabulary usage) as well as elements that the student is struggling with (grammatical error rates). The concept is illustrated in Figure 1.

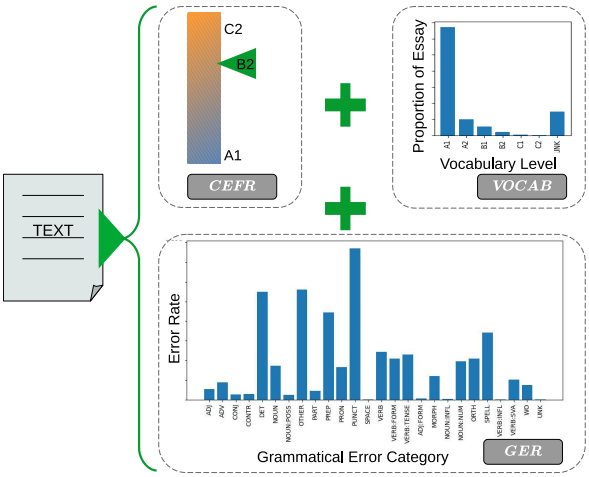

Figure 1: The *writing profile* concept; rather than a single score, a student's writing is represented by a profile with an estimated CEFR level, grammatical error rates (GER) and a profile of vocabulary usage (VOCAB).

Writing profiles provide a dual-purpose representation that is simultaneously human-interpretable (offering pedagogical reference for educators) and machine-readable (as required for automated scoring). In this work, the profiles integrate a holistic essay score mapped to the CEFR[1] (North, 2005), with granular diagnostic actionable data. Specifically, the grammatical error rates are mapped to 26 error categories that are displayed by the platform, and are actionable by a language teacher; while the vocabulary usage data can be visualised as a histogram depicting vocabulary range and distribution, which aligns with standardised exam-marking

[1] https://www.coe.int/en/web/common-european-framework-reference-languages/level-descriptions

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 562–573
July 3-4, 2026

rubric requirements.[2,3] Together the components constitute a multi-dimensional linguistic signature of a learner's essay, expressing holistic proficiency, specific aptitude and the areas for improvement.

We source our essays from the *Write & Improve Corpus 2024* (WI_CORPUS) (Nicholls et al., 2024), a public dataset sourced from a major online essay practice platform, *Write & Improve*[4] (Andersen et al., 2013; Yannakoudakis et al., 2018). This dataset comprises sequences of revisions of learner essays. The final version of each essay is scored by a human expert and mapped to a CEFR-level; the essay is also manually corrected for grammatical (morpho-syntactic) errors. The holistic score, the distribution of the errors, and the vocabulary-usage in the corrected version provides a high-fidelity *expert profile* which can be used as the gold-standard for computer-generated *auto profiles*.

Auto profiles derived using popular NLP tools (*Language Tool*, *ERRANT*, *spaCy*) do not align closely with these expert profiles. To calibrate the auto-profiles, and significantly enhance their alignment with the human experts, we train a neural model using the expert profiles as target data. Beyond this calibration task, we demonstrate that the same model can be used to accurately forecast a student's final submission profile from an initial draft. Furthermore, we show that the model can also generalise, accurately predicting the writing profile for a subsequent task (which opens possibilities for long-term curriculum planning).

In summary, the contributions of this work are as follows: (1) High-fidelity calibration: a model for accurately aligning automated CEFR-level assignments and grammatical error rates with expert judgments; (2) Vocabulary forecasting: predictions of a learner's exhibition of vocabulary-usage on a task; (3) Predictive modeling: a model to predict a student's performance in the final version of the current task from an early draft; and (4) Longitudinal predictions: a model to predict a student's performance in a subsequent task.

[2] https://takeielts.britishcouncil.org/sites/default/files/ielts_task_2_writing_band_descriptors.pdf

[3] https://www.cambridgeenglish.org/images/210434-converting-practice-test-scores-to-cambridge-english-scale-scores.pdf

[4] https://writeandimprove.com/

2 Related Work

Accurate modeling of longitudinal skill acquisition is key for adaptive, personalised tutoring systems (Corbett and Anderson, 1994; Anderson et al., 1995). These systems typically rely on a well-defined taxonomy of skills (or *knowledge components* – Koedinger et al. 2012) with a particular acquisition model such as that in Cen et al. (2006) and Pavlik Jr et al. (2009), although implementations vary widely.

In language learning, knowledge components tend to be less clearly defined than in STEM subjects. Resources such as the *English Vocabulary Profile* [5] or the *English Grammar Profile* (O'Keeffe and Mark, 2017) describe components in a qualitative way, but not as directly machine-readable rules. Consequently, many AWE platforms prioritise identification of errors over an analysis of correct usage.

The tasks of Grammatical error detection (GED) and correction (GEC) have been tackled using a range of methods including feature-based classification and machine-translation (for a survey see Bryant et al., 2023). There are several widely-used, annotated corpora for this task but the error typing within them is not consistent. We opt to use the ERRANT tool that has an independent error typology that can be used with any such annotated corpora (Bryant et al., 2017).

In our work, Automated Essay Scoring (AES) becomes a sub-task of AWE. AES typically refers to the task of predicting a holistic grade or analytic grades for an essay without interpretable diagnostics. Recent preferred methods for AES function as "black boxes" (e.g. Mayfield and Black, 2020 and Mansour et al., 2024). By training a model on writing profiles, which are interpretable (as required by legislation[6]) we remove the black-box limitations but retain the accuracy of neural models.

To our knowledge only Zaidi et al. (2019) has attempted a similar prediction task on comparable data from the *Write & Improve* platform, and our work draws comparisons with that study as part of our evaluation.

[5] https://englishprofile.org/?menu=english-vocabulary-profile

[6] https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence

3 Write & Improve Corpus 2024

The *Write & Improve* platform has a publicly available dataset (WI_CORPUS) used in this work (Nicholls et al., 2024). This logs the essay-writing activities of English language learners, against prompts covering typical topics such as pets, holidays, *etc.* For each essay submitted, the *Write & Improve* automarker assigns a level on the CEFR scale (see A.3) and gives feedback on errors in the text, inviting the student to correct and resubmit. Hence there is a sequence of submissions for each student-prompt pair. The student chooses when a task is complete and it is time to move on to another prompt.

As well as the platform scores, the WI_CORPUS dataset includes annotations from a human expert for the final version of each essay: this consists of the human-judged CEFR level, and a human-corrected version of the text. This expert information is used to create *expert profiles*, which form the prediction targets for our neural model.

We define our own validation and test folds, removing a subset of samples where the expert human-corrected versions are not currently publicly available. There are 679 users responding to 50 prompts in the filtered dataset, and 23k submissions. 95% of drafting series are between 2 and 16 submissions long, median=3, mean=4.48. The mean Levenshtein ratio (Levenshtein, 1966) for final draft to human-corrected text is 0.96. For all drafts to human-corrected text it is 0.91. Between pairs of drafts it is 0.97. Prompts in the dataset are assigned to one of three difficulty levels: *beginner*, *intermediate*, or *advanced*.

4 Expert Profile Prediction

The main objective (Task 1) for this work is to improve the auto profiles so that they are more in line with the expert profiles. The secondary objective (Task 2) is forward prediction, where we attempt to predict the final draft's human profile from an earlier version. The tertiary objective (Task 3) is a combination of these – we take the user automarker features from their final attempt at prompt i and try to forward predict the final profile for prompt $i+1$, to see how well the final user features from prompt i generalise to other prompts on the platform.

4.1 Automatic Writing Profiles

In order to work with writing profiles, we must first extract them. In WI_CORPUS, only the final drafts

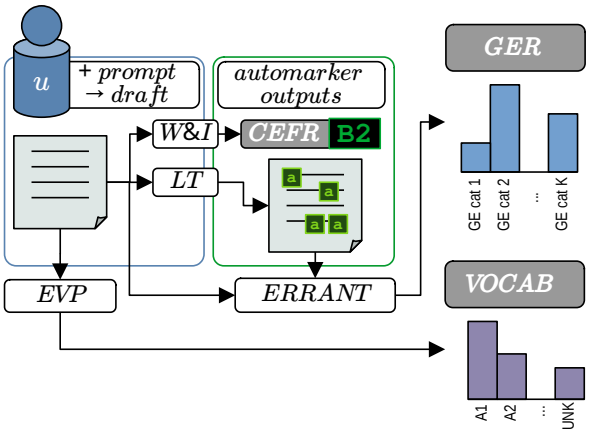

Figure 2: *Auto profile* generation. *W&I* is the *Write & Improve* automarker, *LT* is Language Tool. *EVP* is English Vocabulary Profile. *ERRANT* is the ERRANT grammar annotator. The gray text boxes denote components of the resultant profile. (See section 4.4)

are paired with human CEFR scores and human-corrected texts – these can be used directly in expert profile extraction. For earlier drafts we must approximate some aspects. The dataset already contains an auto-marked CEFR level for every draft. To get an estimated distribution of grammatical errors we must auto-correct the text and compare back to the original using ERRANT (the process is described in section 4.4.3). We also get the distribution of vocabulary used in the draft, by looking up the CEFR level of each content word (see section 4.4.4). These three components together form the *auto profile* of an essay attempt. See Figure 2.

4.2 Neural Network Architecture

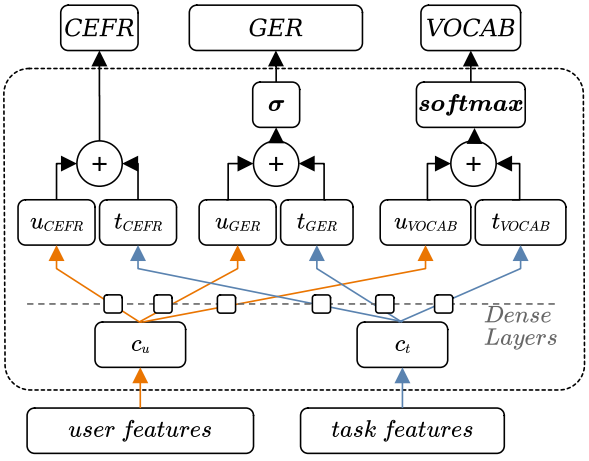

Figure 3: Neural network architecture. A feedforward network with three branches for each component of the predicted output.

The architecture of our neural system can be seen

in Figure 3. The system is feature-based and takes user features $\mathbf{u} = (u_0, \cdots, u_n)$ and task features $\mathbf{t} = (t_0, \cdots, t_m)$. These features are concatenated for each sample to get vectors $\mathbf{c}_u$, $\mathbf{c}_t$. For the Grammatical Error Rate predictions, the concatenated inputs are converted into a pair of 26-dimensional vectors by a matrix product, which are then combined as follows:

$$u_{GER} = W_{gu} \cdot D(\mathbf{c}_u)$$
$$t_{GER} = W_{gt} \cdot D(\mathbf{c}_t)$$
$$\widehat{GER} = \sigma(u_{GER} \oplus t_{GER})$$

where $D(\cdot)$ is a dropout layer with possible $rate \in [10^{-10}, 1]$ and σ is the *logistic sigmoid* activation function, $\sigma(z) = 1/(1 + e^{-z})$. The elementwise addition $\oplus$ combines the propensity for user error (u_{GER}) with the intrinsic opportunities for error in the exercise (t_{GER}). The sigmoid ensures that these are always in the $[0, 1]$ range – that is, rates are expressed as a per-category *Rasch model* (Rasch, 1980).

For CEFR level, the outputs of the matrix products are a single number, and there is no sigmoid since the target range is $[0, 12]$:

$$u_{CEFR} = W_{cu} \cdot D(\mathbf{c}_u)$$
$$t_{CEFR} = W_{ct} \cdot D(\mathbf{c}_t)$$
$$\widehat{CEFR} = u_{CEFR} \oplus t_{CEFR}$$

For the Vocabulary Profile predictions, the histogram estimate is constructed in a similar manner to other outputs, but it is normalised using a *softmax* activation function (Bridle, 1990):

$$u_{VOCAB} = W_{vu} \cdot D(\mathbf{c}_u)$$
$$t_{VOCAB} = W_{vt} \cdot D(\mathbf{c}_t)$$
$$\widehat{VOCAB} = softmax(u_{VOCAB} \oplus t_{VOCAB})$$

The outputs from the neural network are then returned as a triple with element dimensions (26,1,7):

$$OUT = (\widehat{GER}, \widehat{CEFR}, \widehat{VOCAB})$$

Implementation was in *Keras* (Chollet et al., 2015) on *Tensorflow* backend (Abadi et al., 2015) using utilities from *scikit-learn* (Pedregosa et al., 2011).

4.3 Loss functions

System loss is the sum of three loss functions:

4.3.1 Grammar Error Rates

For grammatical error rate vectors we use *Euclidean distance* as a loss to minimise the overall distance between prediction and target:

$$L_{GER} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\sum_{k} (x_k - y_k)^2}$$

where the index k spans across the components of the vector, and i spans across the N samples in the dataset. *Mean squared error* loss can be used, but Euclidean distance gives slightly better results.

4.3.2 CEFR Level

CEFR level is modelled as a scalar and for this we use *mean squared error* as a loss:

$$L_{CEFR} = \text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2$$

where x_i is the predicted value, y_i is the actual value, and N is the number of samples.

4.3.3 Vocabulary

Vocabulary use is represented as an empirical probability distribution, so we can use *cross-entropy loss* as an appropriate loss function:

$$L_{VOCAB} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c \in C} y_c \log(x_c)$$

where C is the number of classes (*i.e.* CEFR levels), y_c is the true likelihood of class c, and x_c is the predicted likelihood of class c. Both $x_c, y_c \in [0, 1]$ and $\sum_c x_c = \sum_c y_c = 1$.

4.4 Input Features

The WI_CORPUS dataset contains textual and metadata features for student essays. In this section we outline the input features and the pre-processing required to obtain them. For more information on dimensions and types, see A.4.

4.4.1 Auto CEFR Level

This is the CEFR level reported by the *Write & Improve* auto-marker for each essay submission, expressed as an integer in range [0,12].

4.4.2 Best-essay flags

This *best-essay* feature consists of two binary flags: one indicates the best draft submitted by the user on this task (*task-best*), and one shows the best draft submitted by the user in the whole dataset

```
S Everyhtings seem quite meaningless to me .
C Everything seems quite meaningless to me .
A 0 1||R:SPELL||Everything||REQUIRED||-NONE-||0
A 1 2||R:VERB:SVA||seems||REQUIRED||-NONE-||0
```

Figure 4: ERRANT-style annotations. S is the original sentence, C is the corrected sentence, Rows marked A are annotations suggested by ERRANT - here a replacement spelling (R:SPELL) and a replacement on the verb to change the subject-verb-agreement (R:VERB:SVA).

(*user-best*). The motivation behind this feature is to provide a signal about the level of this draft relative to previous drafts, on the basis that the task-best and user-best drafts are more reflective of the upper bound of the student's ability. We use the student's automarked CEFR history to set these values.

4.4.3 Grammatical Error Rates

A learner's grammatical ability is expressed as a *grammatical error rates* (GER) object, a vector where each element corresponds to a type of grammatical error. Elements are real-valued and show the per-token rate of occurrence of that error type within a draft. For the main GER objects, the text written by a student is analyzed using *Language Tool* (LT), a widely-used grammatical error detection and correction system (Miłkowski, 2010). This system returns a list of probable corrections that are then applied to the student text, to get the LT-corrected text. This is then compared to the original in ERRANT to get the corrections categorised as ERRANT edit codes.

Figure 4 gives an example of ERRANT-style annotation. The red highlights in the first S(entence) line show the words that ERRANT has tagged as edits. The second line is the C(orrection) by a human expert. The third and fourth A(nnotation) lines show token spans for corrections (in blue). The strings highlighted in green are the error types (*e.g.*, R:SPELL is a spelling error; R:VERB:SVA is a subject-verb agreement error).

ERRANTised GER vectors are simplified by collapsing the R/M/U prefixes (for *replace*, *missing*, *unnecessary*) to get 26 categories (see appendix A.2). Note that in the target data, GERs are only available as ERRANT-type vectors, since they are derived from human corrections. ERRANT GER vectors are called **uGER** in the results.

As well as creating ERRANT-style GER vectors, it is possible to create a similar object for LT output directly. Although LT has hundreds of rules to identify errors, it groups them into categories of which 15 appear in WI_CORPUSS. Counts are summed and normalised as before. LT GER vectors are denoted **uGER.LT** in the results.

4.4.4 Vocabulary Histogram (uVOCAB)

This feature is a histogram of the proportion of words in a draft that fall into each of the CEFR vocabulary difficulty levels, as defined by the *English Vocabulary Profile* (EVP) (Capel, 2015). These levels cover the same range as the CEFR – for vocabulary each token is categorised to a level and counted in the histogram. As well as A1–C2 there is also an unknown (UNK) category. The EVP level for a word is dependent on the word-class and word-sense of the usage. To ascertain word-class (noun, verb *etc.*) we use the *spaCy part-of-speech* tagger (Montani et al., 2023) and we select from the EVP the CEFR level for that class. If no such word-class is recorded, we assume the lowest CEFR level for that word as in Tyen et al. (2022). The EVP look-up uses both US and UK English gazetteers. We do not attempt to disambiguate word senses: a word-sense disambiguation model could be trialed in future work.

4.4.5 Essay statistics

Included in the feature set is the number of tokens in the essay (known to correlate with proficiency level) and the revision number (as a raw count).

4.4.6 Profile averages (tCEFR, tGER, tVOCAB)

To help provide a central value against which to measure the offset of grammatical and proficiency scores, we optionally include the mean values for CEFR level, GER and VOCAB for each task.

4.4.7 Avg. Error Deltas (dGER.LT, dCEFR)

These values measure average change in uGER.LT and uCEFR level per submission for each task *t*.

4.5 Feature and hyperparameter search

The features given in Section 4.4 are filtered to select a subset with high performance. We use *Bayesian optimisation* (Pelikan, 2005) with five-fold cross validation. Specifically, features are selected, and applied uniformly across all five folds. The mean of the validation losses is used in the selection optimisation step. This prevents coincidental outlier scores on particular folds from dominating the tuning process. Hyperparameter search was carried out as part of the same process. Hyperparameters include dropout rate $\in [10^{-10}, 1]$

for each dense layer, number of dense layers on each branch $\in [0, 5]$, width of dense hidden layers $\in [5, 200]$, and choice of activation functions $\in [sigmoid, linear, ReLU]$. In practice, no hidden layers are needed to get good results, and dropout rates are best when small ($< 10^{-3}$).

4.6 Evaluation

We simultaneously predict three targets (the expert assigned values of CEFR level, GER and VOCAB). This section outlines the evaluation for each. For more information on target dimensions and types see A.5.

4.6.1 Evaluation of CEFR level predictions

For the evaluation of CEFR level prediction, we use *Root Mean Squared Error* (RMSE) to provide a measure that can be compared directly to the CEFR. This is calculated globally with equal weighting across all data points (Pelánek, 2018).

4.6.2 Evaluation of GER predictions

We can compare true and predicted GER vector values $\mathbf{g}$ and $\hat{\mathbf{g}}$ using *cosine proximity*:

$$CosProx(\mathbf{g}, \hat{\mathbf{g}}) = \frac{\sum_k g_k^2 \hat{g}_k^2}{\sqrt{\sum_k g_k^2}\sqrt{\sum_k \hat{g}_k^2}}$$

For positive valued vectors like GER profiles, cosine proximity ranges from [0,1] with 1 being a perfect match. The use of cosine proximity is based on Zaidi et al. (2019) and it measures the directional similarity of the vectors (*i.e.*, the relative distribution of the magnitudes of components), but it does not take into account vector magnitude. We use *Root Mean Squared Error* (RMSE) as a secondary measure, to check that rates are at the correct levels as well as being distributed in the same way.

4.6.3 Evaluation of vocabulary distributions

Vocabulary distributions are empirical frequency distributions, so our interest is in how similar the histogram is as a whole. We use *Earth Mover Distance* (EMD). For histogram vectors with true value $\mathbf{h}$ and prediction $\hat{\mathbf{h}}$ we have:

$$EMD_0 = 0$$
$$EMD_{i+1} = h_i + EMD_i - \hat{h}_i$$
$$EMD(\mathbf{h}, \hat{\mathbf{h}}) = \sum_{c=1}^{|C|} |EMD_c|$$

where there are C classes in each histogram. EMD $\in [0, 1]$ where 0 is a perfect match.

4.7 Testing

Testing is carried out with a separate held-out test set for each of the five validation folds. In each case, data is partitioned by student, so any student in the test set will not have appeared in either the training or validation data. This ensures that we are measuring generalisable performance and not just re-using samples that have been encountered during training.

5 Results

Table 1 summarises the results of this work, subdivided into the three tasks; for Task 1 we also provide single-feature scoring and best-performing combinations. We also include baseline scores for Zaidi et al. (2019) and the results of our attempts to reproduce this approach.

The first line for Task 1 shows that the *Write & Improve* automarker predicts the expert CEFR level with mean RMSE of 1.403 (where a unit is half a CEFR level). The LT automarker creates GER vectors that are 0.363 cosine proximal to the expert versions. Vocabulary extractions are good predictors, with a mean EMD of 0.012 (out of 1.0) from the expert-corrected case.

We consider three categories (user-essay features comprising Document Basics and Auto Profiles; and Prompt Features) on their own and in combinations. The listing in Table 1 represents a breakdown of the best combinations found by feature filtering.

Some features are clearly more promising than others. In the *Document Basics*, the best.try flag achieves the best CEFR prediction with 1.742 RMSE and the deltas achieve the best cosine proximity (0.644). For vocabulary distribution the EMD scores are good (0.034-0.035) but vocabulary features are needed to improve this further.

In the *Auto Profile Features* category, uCEFR is the strongest single feature for the CEFR target with 1.050 RMSE, but it is otherwise unremarkable. uVOCAB is the only good predictor of the final vocabulary distribution, dropping the EMD from at best 0.034 to 0.012.

In the *Prompt Features*, we see that tCEFR and tGER have the greatest influence on their respective targets: if we include tCEFR, CEFR RMSE drops to 1.084, so the mean CEFR is a strong indicator as we might expect. Similarly, for tGER alone, GER cos. prox. is good at 0.650 but for other targets it is less strong. The *deltas* alone look promising but are less effective in combination with other features.

	CEFR level	GER		VOCAB
Task 1: Improving profiles	RMSE↓	Cos.Prox.↑	RMSE↓	EMD↓
Automarker performance	1.403	0.363	0.007	0.012
(Document Basics)				
best.try	1.742	0.643	0.007	0.035
n.tokens	1.784	0.640	0.007	0.035
version	1.910	0.640	0.007	0.035
deltas	1.852	0.644	0.007	0.035
best.try+n.tokens+version	1.103	0.623	0.007	0.034
best.try+n.tokens+version+deltas	1.106	0.585	0.008	0.034
(Auto Profile Features)				
uCEFR	1.050	0.622	0.007	0.035
uGER	1.707	0.661	0.007	0.034
uGER.LT	1.708	0.657	0.007	0.034
uVOCAB	1.297	0.648	0.007	0.012
All auto profile features (uALL)	0.932	0.663	0.007	0.012
(Prompt Features)				
tCEFR	1.084	0.626	0.007	0.034
tGER	1.595	0.650	0.007	0.031
tVOCAB	1.267	0.649	0.007	0.030
All prompt features (tALL)	1.076	0.650	0.007	0.030
(Replicating Zaidi et al., 2019)				
Prev. Baseline	1.383	-	-	-
Prev. Best	1.093	0.426	-	-
Closest (uGER+best)	1.045	0.639	0.007	-
uALL+tALL (excl. VOCAB)	0.848	0.664	0.007	-
uALL+tALL+best.try (excl. VOCAB)	0.838	0.665	0.007	-
(Best Overall Combinations)				
uALL+tALL	0.817	0.670	0.007	0.012
uALL+tALL+n.tokens	0.816	0.664	0.007	0.012
uALL+tALL+deltas	0.817	0.664	0.007	0.012
uALL+tALL+best.try	**0.809**	**0.670**	**0.007**	**0.011**
Task 2: Forward prediction				
Automarker performance	1.426	0.339	0.006	0.016
Best from Task 1	**0.874**	**0.659**	**0.006**	**0.014**
Task 3: Generalisation				
Automarker performance	1.913	0.283	0.007	0.040
Best from Task 1	**0.963**	**0.649**	**0.007**	**0.028**

Table 1: Results for predicting human expert profile scores from auto profiles. Best features or combinations for each subclass are underlined, best combinations for each Task are in **bold**. ↑=higher is better; ↓=lower is better.

The most performant combination is all *Auto Profile* and *Prompt* features, with the best.try metric. This results in CEFR RMSE of 0.809, GER cos. prox. 0.670 and VOCAB EMD of 0.011.

For comparison with previous work we include a baseline and best score from Zaidi et al. (2019), which is the closest precedent to our Task 1. No baseline grammar scores were given and vocabulary was not included in that paper. In *Zaidi et al.* the highest performing configuration used just GER profile and best.try metric – with these features we achieve CEFR RMSE of 1.045 and GER cos. prox. of 0.639, both better than the earlier work. With our best configurations, but excluding VOCAB entirely (as per *Zaidi et al.*) we do better still, with CEFR RMSE of 0.838 and GER cos. prox. of 0.665.

In Task 2, we attempt to predict the final expert profile, using auto profiles from earlier drafts. Here,

using the best configuration from Task 1, we see CEFR RMSE of 0.874 and GER cos. prox. of 0.659. VOCAB EMD is 0.014. These scores are slightly worse than for Task 1, although still beating the Task 1 automarkers.

In Task 3, we attempt to predict the expert profile for the final draft of the *next* series that the student will undertake. In this task we see CEFR RMSE of 0.963 and GER cos. prox. of 0.649. Vocabulary scores, although low in the full range of EMD, are half as good as in other tasks, with EMD 0.028.

6 Discussion

The multi-task learning configuration used in this work is currently the best recorded in this class of task and data. In Zaidi et al. (2019) a larger, private data set is used but the best performance (using an ERRANTised GER vector and the best-essay flags to target CEFR level) is weaker in terms of both CEFR level prediction and GER cosine proximity (1.093 to 0.809 RMSE and 0.426 to 0.669 cos. prox. respectively).

The implication is that a richer target is just as important as having richer input features (which are tested without leading to any improvement in *Zaidi et al*). The 84% increase in GER cosine proximity is a key result, but consistent GER RMSE of 0.006-0.007 is also better than we might expect, thanks to the choice of loss function. Retrained with negative cosine proximity as the loss, our best system returns similar cosine proximity but GER RMSE 0.332, much worse than with Euclidean loss.

Outside of prediction, we can inspect the internal representations output by the interior branches of the network. For instance we can extract and concatenate the CEFR, GER and VOCAB activations just for the *user* branches. This is shown for 200 randomly selected students (at submission of final essay) projected to 2D using MDS (Kruskal, 1964) in Figure 5. Colours are (a) expert CEFR and (b) the user's choice of prompt difficulty. Neither of these is used in training, but banded clustering is evident for both, which suggests these representations could be used to apply adaptive pedagogical policies, such as changing a student's prompt group when their characteristics do not fit their current grouping.

7 Visualiser

We make the data from this work available in a visualisation tool[7] (See figure 6). For each student the tool presents personalised profile data: (1) Student learning statistics, including average human and automarker CEFR level, ranks in cohort, average number of drafts, and rate of change of CEFR and average GER per draft. (2) CEFR level time series. (3) Average grammatical error rate (GER) time series. (4) Grammatical error rate (GER) summary. (5) Vocabulary distribution (VOCAB) summary. In the time series and grammar charts, the gold line shows our predicted score, teal shows the human expert scores and cyan shows the automarker scores.

8 Limitations and Future Work

The technique described in this work is simple and fast to train, but it is *Markovian* and assumes that the effect of history has been incorporated into the features seen at each timestep. While it may be reasonable to model skill as cumulative in this way, it may be better to include history explicitly using a method such as Knowledge Tracing (Corbett and Anderson, 1994; Piech et al., 2015). This is scheduled for future work. Another limitation is that this study has been restricted to a single data-set and platform but this is hard to rectify since there is a scarcity of such corpora.

9 Conclusion

We introduce interpretable multi-component vectors, *writing profiles*, that describe a learner's writing in terms of CEFR level, and vocabulary usage alongside their grammatical errors. Using a neural model, we effectively align automatically generated writing profiles with high-quality human-expert profiles, enhancing AES performance. Further, the neural model is shown to generalise across essay drafts and future tasks; paving the way for future work in longitudinal curriculum planning. Lastly, we make writing profile data explorable within our longitudinal visualisation tool.

Acknowledgements

This paper reports on research supported by *Cambridge University Press & Assessment.*

[7] https://github.com/rjm49/bea26_visualiser

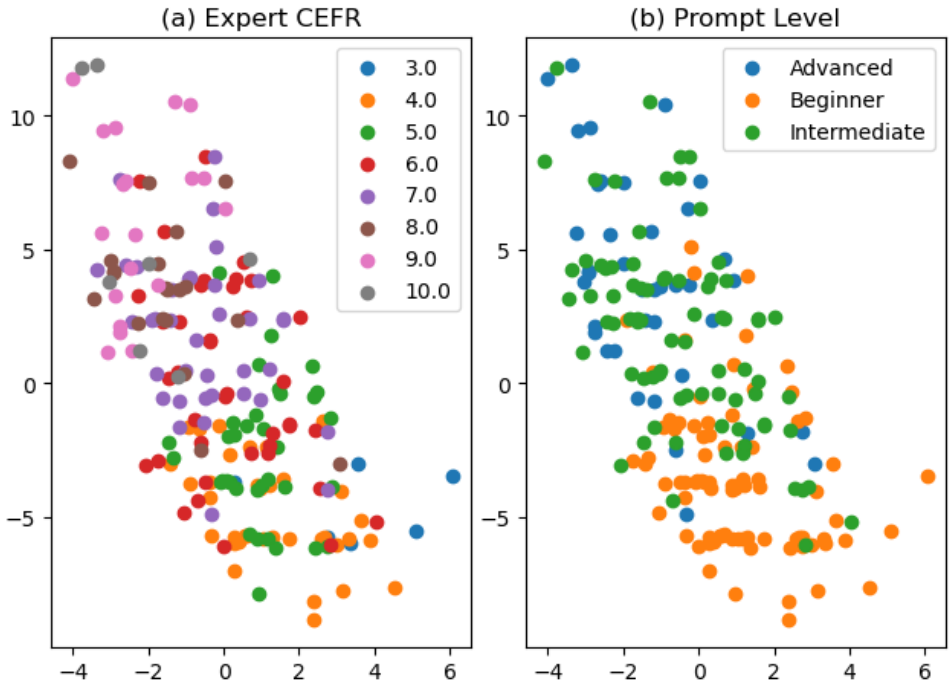

Figure 5: Representations of students from the test set, concatenated from the branches of the neural network associated with the user only. These are projected into 2D by MDS, and coloured by (a) expert CEFR and (b) prompt (difficulty) levels. These colour groupings are not known to the system, so the scatters reveal that this structure has been learned independently. This may be helpful in supporting pedagogical adaptivity such as deciding which prompt group a student should actually belong to.

Figure 6: Visualiser Screenshot - student detail view. For each student the tool presents: (1) Student learning statistics. (2) Student CEFR level time series. (3) Student average grammatical error rates time series. (4) Grammatical error summary. (5) Vocabulary usage summary. In the time series and grammar summary charts, the gold line represents our predicted score, dark teal is expert score, cyan is automarker score.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, and 21 others. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Øistein E Andersen, Helen Yannakoudakis, Fiona Barker, and Tim Parish. 2013. Developing and testing a self-assessment and tutoring system. *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 32–41.

John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. 1995. Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2):167–207.

John S. Bridle. 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neural Networks for Signal Processing*, pages 227–236. MIT Press.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, 49(3):643–701.

Annette Capel. 2015. The English Vocabulary Profile. In J. Harrison and F. Barker, editors, *English Profile in Practice*. Cambridge: Cambridge University Press.

Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*, pages 164–175. Springer.

François Chollet and 1 others. 2015. Keras. `https://keras.io`.

Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4:253–278.

Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. 2012. The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798.

Joseph B Kruskal. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Watheq Ahmad Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. Can large language models automatically score proficiency of written essays? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2777–2786, Torino, Italia. ELRA and ICCL.

Elijah Mayfield and Alan W Black. 2020. Should you fine-tune BERT for automated essay scoring? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 151–162, Seattle, WA, USA → Online. Association for Computational Linguistics.

Marcin Miłkowski. 2010. Developing an open-source, rule-based proofreading tool. *Software: Practice and Experience*, 40(7):543–566.

Ines Montani, Matthew Honnibal, Matthew Honnibal, Adriane Boyd, Sofie Van Landeghem, and Henning Peters. 2023. explosion/spacy: v3.7.2: Fixes for apis and requirements.

Diane Nicholls, Andrew Caines, and Paula Buttery. 2024. The Write & Improve Corpus 2024: Error-annotated and CEFR-labelled essays by learners of English. Cambridge University Press & Assessment.

Brian North. 2005. The CEFR levels and descriptor scales. In Lynda Taylor and Cyril Weir, editors, *Multilingualism and assessment: Achieving transparency, assuring quality, sustaining diversity. Proceedings of the ALTE Berlin Conference*. Cambridge: Cambridge University Press.

Anne O'Keeffe and Geraldine Mark. 2017. The English Grammar Profile of learner competence: Methodology and key findings. *International Journal of Corpus Linguistics*, 22(4):457–489.

Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. 2009. Performance factors analysis–a new alternative to knowledge tracing. In *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*.

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

Radek Pelánek. 2018. The details matter: methodological nuances in the evaluation of student models. *User Modeling and User-Adapted Interaction*, 28(3):207–235.

Martin Pelikan. 2005. Bayesian optimization algorithm. In *Hierarchical Bayesian optimization algorithm: toward a new generation of evolutionary algorithms*, pages 31–48. Berlin, Heidelberg: Springer.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing.

Georg Rasch. 1980. *Probabilistic models for some intelligence and attainment tests*. Chicago: University of Chicago Press.

Gladys Tyen, Mark Brenchley, Andrew Caines, and Paula Buttery. 2022. Towards an open-domain chatbot for language practice. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 234–249, Seattle, Washington. Association for Computational Linguistics.

Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for ESL learners. *Applied Measurement in Education*, 31(3):251–267.

Ahmed H Zaidi, Andrew Caines, Christopher Davis, Russell Moore, Paula Buttery, and Andrew Rice. 2019. Accurate modelling of language learning tasks and students using representations of grammatical proficiency. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM)*.

A Appendix

A.1 CEFR Level Mapping

Level	Numeric
pre-A1	0
A1	1-2
A2	3-4
B1	5-6
B2	7-8
C1	9-10
C2	11-12

Table 2: CEFR Level Mappings: note that each CEFR level is divided into two sublevels, *e.g.* A1 and A1+

A.2 ERRANT Error Categories

Code	Description
ADJ	Adjective
ADJ:FORM	Adjective form
ADV	Adverb
CONJ	Conjunction
CONTR	Contraction
DET	Determiner
NOUN	Noun
NOUN:POSS	Possessive noun
MORPH	Morphology
NOUN:INFL	Noun inflection
NOUN:NUM	Noun number
ORTH	Orthography
OTHER	Other
PART	Particle
PREP	Preposition
PRON	Pronoun
PUNCT	Punctuation
SPACE	Whitespace
SPELL	Spelling
VERB	Verb
VERB:FORM	Verb form
VERB:INFL	Verb inflection
VERB:SVA	Verb subject-verb-agreement
VERB:TENSE	Verb tense
WO	Word order
UNK	Unknown

Table 3: ERRANT codes. ERRANT marks correction as belonging to one of these categories with an additional (R)eplaced (M)issing or (U)nnecessary marker to describe whether edit has changed, added to, or deleted from, the original text. We drop the R/M/U markers and just keep the categories listed.

A.3 Language Tool Error Categories

Language Tool Category	Description
CASING	Incorrect upper and lower casing.
COLLOCATIONS	Incorrect word combinations.
COMPOUNDING	Hyphenation and compound word errors.
CONFUSED WORDS	Commonly confused words.
GRAMMAR	Syntactic errors.
MISC	Other errors (unspecified).
MULTITOKEN SPELLING	Words incorrectly split or merged.
NONSTANDARD PHRASES	Slang and inappropriate colloquial forms.
PUNCTUATION	Missing or incorrect punctuation marks.
REDUNDANCY	Unnecessary repetition or wordiness.
REPETITIONS STYLE	Immediate or local duplication.
SEMANTICS	Problems with meaning.
STYLE	Problems with register, readability and verbosity.
TYPOGRAPHY	Errors in spacing, quotes and other symbology.
TYPOS	Misspellings.

Table 4: *Language Tool* error categories, with descriptors.

A.4 Input Features

Input Name	Width, Type
uGER	26, float
uGER.LT	14, float
uCEFR	1, integer
uVOCAB	7, float
best.try	2, bit
n.tokens	1, integer
version	1, integer
dGER.LT	14, float
dCEFR	1, float
tGER	26, float
tCEFR	1, float
tVOCAB	7, float

Table 5: Dimensions and types for the input features used in this work.

A.5 Prediction Targets

Target Name	Width, Type
GER	26, float
CEFR	1, integer
VOCAB	7, float

Table 6: Dimensions and types for the prediction targets used in this work.