Accurate modelling of language learning tasks and students using representations of grammatical proficiency

Ahmed H. Zaidi
Computer Laboratory
University of Cambridge
ahz22@cl.cam.ac.uk

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

Christopher Davis
Computer Laboratory
University of Cambridge
ccd38@cam.ac.uk

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

Paula Buttery
Computer Laboratory
University of Cambridge
paula.buttery@cl.cam.ac.uk

Andrew Rice
Computer Laboratory
University of Cambridge
acr31@cam.ac.uk

ABSTRACT

Adaptive learning systems aim to learn the relationship between curriculum content and students in order to optimise a student's learning process. One form of such a system is content recommendation in which the system attempts to predict the most suitable content to next present to the student. In order to develop such a system, we must learn reliable representations of the curriculum content and the student. We consider this in the context of foreign language learning and present a novel neural network architecture to learn such representations. We also show that by incorporating grammatical error distributions as a feature in our neural architecture, we can substantially improve the quality of our representations. Different types of grammatical error are automatically detected in essays submitted by students to an online learning platform. We evaluate our model and representations by predicting student scores and grammatical error distributions on unseen language tasks. We also discuss further uses for our model beyond content recommendation such as inferring student knowledge components for a given domain and optimising spacing and repetition of content for efficient long term retention.

Keywords

language learning, task representations, student representations, grammatical errors, deep learning, student modelling

1. INTRODUCTION

Adaptive learning is a computational procedure for the automatic selection and presentation of teaching materials which are deemed most suitable for the user of an educational platform. In this framework, the platform user – a student – is guided through online courseware – a curriculum – in an optimal and personalised fashion. In order to select items (tasks) for students appropriately it is necessary to relate accurate machine-readable representations of each individual task to machine-readable representations of each student[1]. Such representations can be used to predict future performance on parts of the curriculum that a student is yet to reach (as in [20, 13, 44]). These predictions can in turn be used to select the set of appropriate next items for this individual – those which are not too easy and not too hard (as in [1, 11]).

In general the adaptive learning approach has been shown to lead to improved learning outcomes for student users of educational platforms [17, 25, 30]. However, there remains a question of what is the best methodology to construct representations for students and tasks. Previous approaches manually engineer features to construct representations [22]. These features are usually tuples of a knowledge component (e.g. differentiation, fractions in the case of maths) and student outcome (i.e. whether or not the student demonstrated understanding for that knowledge component through completing the task). A task may contain multiple knowledge components. Whilst this approach is highly interpretable, in the domain of language learning, it is difficult to clearly divide the tasks into knowledge components. Furthermore, in the recently popular paradigm of deep learning, we have seen that training representations through neural networks have yielded state-of-the-art results in the space of image recognition, and various natural language tasks.

Motivated by this, we propose a methodology of automatically developing high quality representations of students and tasks in a language learning context. Having reliable student and task representations in place facilitates work on downstream tasks such as curriculum learning and recommender systems for language learning.

Representations are derived from a novel neural architecture (described in Section 4.2) and real student data collected through the Write & Improve[2] (W&I) assessment and feedback platform for learners of the English language. [41]. Our representations take the form of *embeddings* – numeric vec-

[1] Note the terms 'student' and 'user' are used synonymously; as are 'task' and 'item'.

[2] `https://writeandimprove.com`

tors of a certain dimensionality, densely representing complex datasets. Using such a methods enables us to avoid explicitly defining the knowledge components that make up students and tasks (a task that presents many challenges in the language learning domain).

Additionally, such representations enable us to draw upon established methods from *representation learning*[3] including concatenating embeddings from different sources of information, learning representations of different targets (in our case, users and tasks) and passing the resultant vectors to multi-layered neural networks to train prediction models for unseen data.

To develop our student representations we incorporate information about a student's essay submissions to W&I, score history, and the grammatical errors made on every task – all together an approximation of the student's knowledge state for language learning at any given point. Our task representations, on other hand, incorporate the aforementioned information for all the students who have attempted a particular task. The reason for this design choice is motivated by the view that the appropriateness or difficulty of a task is defined by the way students interact with the task. We further constrain our task embeddings by training it to predict it's respective difficulty level (beginner, intermediate and advanced).

We evaluate the quality of our student and task representations extrinsically: 1) we use a combination of student and task representations to predict a student's overall score on a given task; 2) we use the student and task representations to predict the grammatical errors a student will make on a given task. The first task is a conventional one in educational data mining [16]; the second tests the generalisability of the student representations by evaluating their aptitude for *transfer learning* – the application of machine models trained on one problem onto a different but related problem [27].

Our best-performing neural network model incorporates grammatical error distributions detected by ERRANT [6] as a feature and achieves mean squared error (MSE) of 1.195 on score prediction, an absolute value of 1.093 on a scoring scale of 0-13. On the second task of predicting grammatical errors on an unseen task, we achieve a cosine proximity score of -0.426 (-1 being perfect alignment). These results support the signal that grammatical error distributions provide in determine student ability.

Our main contributions are as follows:

- The introduction of a novel neural framework that can be used to automatically learn student and task representations for language learning without explicitly modelling knowledge components.
- The incorporation and evaluation of automatically detected grammatical error representations as a key feature in our neural network classification model to learn user and task representations. When tested on an unseen task, our set-up yields reliable prediction of both user-task score as well as grammar errors made by students on tasks.

2. RELATED WORK

Our general objective is modelling the acquisition of procedural knowledge [8], and we can usefully envisage this as the successful learning of 'knowledge components' (KCs) for any given educational domain [15]. Models which take knowledge components into account have been shown to trace learning more successfully than otherwise [7, 12].

Personalisation in educational technology is of wide interest, since learners are known to progress at different rates and in different styles [33, 2, 5]. Without an ontology or other knowledge base to guide personalisation [37], we can only represent users through their interaction with learning items (tasks). Whereas well-known recommendation systems may have access to user ratings, reviews, click-throughs and sales figures, our measure of success is user performance – the score assigned to a given essay submission on the proposed item – and representation quality (predicting score and grammar errors on a task using the same representations).

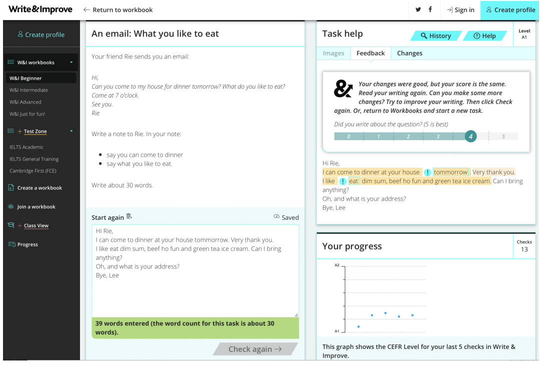
Tracking users as they acquire knowledge in a learning system is a type of knowledge tracing, and previous approaches to knowledge tracing have ranged from item-response theory [40], to Bayesian knowledge tracing [8], to deep learning [22], factorisation [39] and dynamic time warping [35]. We adopt a deep learning approach but, whereas for Montero *et al* there were defined KCs in the mathematics domain (e.g. fractions, differentiation) which could each be assigned binary values representing whether the student got that KC right for a question, in language learning it is not so clear how KCs should be defined and delimited. Therefore we rely on learning representations through interaction and back-propagating from the score assigned to each text, and grammatical error distributions.

To improve our student and task representations we incorporate automatically detected grammatical errors made on a task by a given student as a feature in our neural network model. Grammatical error detection is a well-established research field, with most focus having been placed so far on learners of English. Error detection techniques range from feature-based classification to neural machine translation [31, 43], and widely-used annotated corpora include the First Certificate in English corpus [42], the National University of Singapore Corpus of Learner English [9], and the JHU FLuency-Extended GUG corpus [26]. These corpora all involve different error typologies and one advantage of using ERRANT is that it defined a new error typology independent of but compatible with existing annotated data.

3. WRITE & IMPROVE

On W&I, students are prompted to input a short text of at least 25 words in response to a given question. Once they have completed the task, the system automatically provides

[3] An area of research that focuses on developing representations of data for machine learning tasks.



An email: What you like to eat

Your friend Rie sends you an email:

Hi,
Can you come to my house for dinner tomorrow? What do you like to eat?
Come at 7 o'clock.
See you.
Rie

Write a note to Rie. In your note:

- say you can come to dinner
- say what you like to eat.

Write about 30 words.

Hi Rie,
I can come to dinner at your house tommorrow. Very thank you.
I like eat dim sum, beef ho fun and green tea ice cream. Can I bring anything?
Oh, and what is your address?
Bye, Lee

39 words entered (the word count for this task is about 30 words).

Your changes were good, but your score is the same. Read your writing again. Can you make some more changes? Try to improve your writing. Then click Check again. Or, return to Workbooks and start a new task.

Did you write about the question? (5 is best)

Hi Rie,
I can come to dinner at your house ! tommorrow . Very thank you.
I like ! eat dim sum, beef ho fun and green tea ice cream. Can I bring anything?
Oh, and what is your address?
Bye, Lee

This graph shows the CEFR Level for your last 5 checks in Write & Improve.

Figure 1: Write & Improve example screenshot

a grade on the CEFR scale[4] along with feedback on grammatical errors detected in the text. The W&I automarker assigns each text an integer score between 0 and 13. Table 1 outlines how essay scores are mapped to the CEFR scale.

Table 1: Student scores mapped to CEFR levels

CEFR	Score
A1	1-2
A2	3-4
B1	5-6
B2	7-8
C1	9-10
C2	11-13

For instance, a student may submit a text such as that in (1), for which they receive a score of 1.5, which equates to a grade of A1 (beginner) and indications that *tommorrow* and *I like eat* are ungrammatical. A screenshot of this example is provided in Figure 1.

(1) Hi Rie,
I can come to dinner at your house tommorrow. Very thank you.
I like eat dim sum, beef ho fun and green tea ice cream.
Can I bring anything?
Oh, and what is your address?
Bye, Lee

The student is encouraged to update and resubmit their text for further scoring and error feedback, and there is, in principle, no upper limit on the number of submissions they can make for a given task. It is their choice when to deem the task 'complete' and move on to a new question.

It is our long-term aim to develop an adaptive tutoring system (ATS) for language learners. There are 122 unique question items, or TASKS, in the W&I curriculum. Currently all users of W&I move through the curriculum in an unguided and independent fashion. An ATS would instead guide students from task to task in order to personalise their learning experience and improve their level of performance.

In order to provide this type of guidance, we need accurate representations of task difficulty and student ability as an essential prerequisite. W&I currently has tasks grouped by three broad difficulty levels: beginner, intermediate, advanced. However, our task representations need to be more fine-grained than this, so that we can guide students within and across the broader levels, and identify parts of the broad tripartite curriculum which have been separated *a priori* but are in fact of overlapping difficulty levels. Therefore we attempt to jointly train student and task representations based on past performance of real W&I users to capture the relative difficultly of tasks such that they can be reliably used to predict a particular student's score on a given task.

4. LEARNING STUDENT AND TASK REPRESENTATIONS

Our primary goal was to predict student scores on a given language learning task based on our representations of students and tasks in Write & Improve . Secondary to that, we check the quality of our student representations by predicting the grammar error distribution of a given student-task tuple. In what follows we describe the data, evaluation metrics and models used in this work.

4.1 Write & Improve data

Our training and test data come from the W&I language learning platform. W&I users submit responses that are at least 25 words in length for automated scoring and error feedback, and may opt to answer any number of prompts tagged with one of three difficulty levels – beginner, intermediate, advanced. We obtained application logs of user activity from the past two years – a total of 3+ million essay submissions by 300,000+ account holders.

We filtered the data for users who had submitted at least 10 submissions. This resulted in a dataset of 1.3 million submissions by 100,140 users. We also had a record of the questions ('prompts') users responded to and the scores assigned to their texts by W&I's auto-marker.

In addition, we obtained counts of grammatical errors in

[4] The Common European Framework of Reference for Languages

each submitted text using the ERRANT annotation toolkit [6]. This gives us a distribution over 55 possible error types, of which 47 were observed in the data we work with.

4.2 Model architecture

The architecture of our neural system can be seen in Figure 2. The neural network takes as an input a user id u and task id t which are taken as indices in the user embedding layer U and task embedding layer T respectively. $u \in N_u$ where N_u is the number of unique users in the W&I dataset. $t \in N_t$ where N_t is the number of unique tasks in the W&I dataset.

U is an $N_u \times d_u$ matrix where d_u is the size of the user representation. T is an $N_t \times d_t$ where d_t is the size of the task representation.

The output of U and T are vectors of dimensions d_u and d_t respectively, and will be henceforth referenced as $\vec{u}$ and $\vec{t}$. The description of the score prediction model can be seen in Equation 1:

$$\begin{aligned} c &= (\vec{u}, \vec{t}) \\ h_1 &= \mathbf{D}(\sigma(c \cdot W^1)) \\ h_2 &= \mathbf{D}(\sigma(h_1 \cdot W^2)) \\ s &= h_2 \cdot W^s \end{aligned} \tag{1}$$

– where c is the concatenated vector of features, h_1 and h_2 are the first and second hidden layers, $\mathbf{D}$ is the dropout function [36] and σ is the ReLU activation [24]. W^1, W^2, W^s are the weight parameters of the model. Finally, s is the predicted score of user u on task t.

We optimise our system and learn a user embedding matrix U and task embedding matrix T by minimising the mean squared error (MSE) of our predicted score s and the target score $\hat{s}$:

$$L = \frac{1}{K} \sum_k (s - \hat{s})^2 \tag{2}$$

– where k is a given submission by the user for a particular task.

We introduce an auxiliary objective to predict the difficulty β of each task t, referenced as t_β. The ground-truth labels for task difficulty (beginner, intermediate, advanced) are obtained from the meta-data of each task in the dataset:

$$\begin{aligned} h_3 &= \mathbf{D}(\sigma(\vec{t} \cdot W^3)) \\ t_\beta &= softmax(h_3 \cdot W^\beta) \end{aligned} \tag{3}$$

– where h_3 is the hidden layer between the task embedding matrix T and the output and W^3 and W^β are the weight parameters. We optimise the prediction of task difficulty t_β using a categorical cross-entropy loss function:

$$\mathcal{L} = -\frac{1}{N_t} \sum_i \sum_\beta \cdot 1_{t_\beta \in \beta} \log p(t_\beta \in \beta) \tag{4}$$

4.3 Feature set

In addition to the score s, the W&I dataset contains prompts and answers in natural language as well as metrics on whether submission k is the highest scoring submission by user u. We incorporate these additional features into the architecture of the model in order to evaluate their impact on the quality of user and task embeddings.

4.3.1 Answer embedding

We obtain a vectorised form of each student response using 300-dimension word2vec embeddings[5] pre-trained on the Google News corpus [19]. This means that we have information about the way words tend to be used by knowing which other words they are found to co-occur with, learned from a large dataset of news articles. In our case, the answer embedding for a student's essay is an additive compositional model where the final embedding is a sum of every word in the essay. Whilst this model is not state-of-the-art for distributional semantics, Mitchell & Lapata [21] show that the additive model can yield results comparable to significantly more sophisticated models.

4.3.2 Question embedding

Similar to the answer embeddings, we construct a vectorised form of each prompt represented in natural language, again summing word2vec representations of every word. We were motivated to incorporate question embeddings because we assume that the lexical distribution of words in the prompt is directly correlated to the complexity of the question. We propose that linguistically complex questions are indicative of difficult tasks.

4.3.3 Metric embedding

The motivation behind using the metric embedding is to provide a signal to the model regarding the relative score of the submission in comparison to the user's previous submissions. This signal may facilitate the model to down-weight submissions that are not task-best or user-best, as one could argue that task-best and user-best are a more accurate reflection of the student's holistic capabilities.

The metric embedding is a 2-dimensional vector that stores benchmark information about submission k for user u in comparison to the user's previous W&I submissions. The first dimension is a binary value for whether the score for the submission was the highest score on task t for user u. The second dimension is a binary value for whether the score for the submission was the highest score across all W&I tasks for user u. We have access to each user's score history and

[5] A word2vec embedding is a $1 \times x$ dimensional dense vector that represents a word semantically. Words that are similar in meaning have vectors that are close together in vector space.

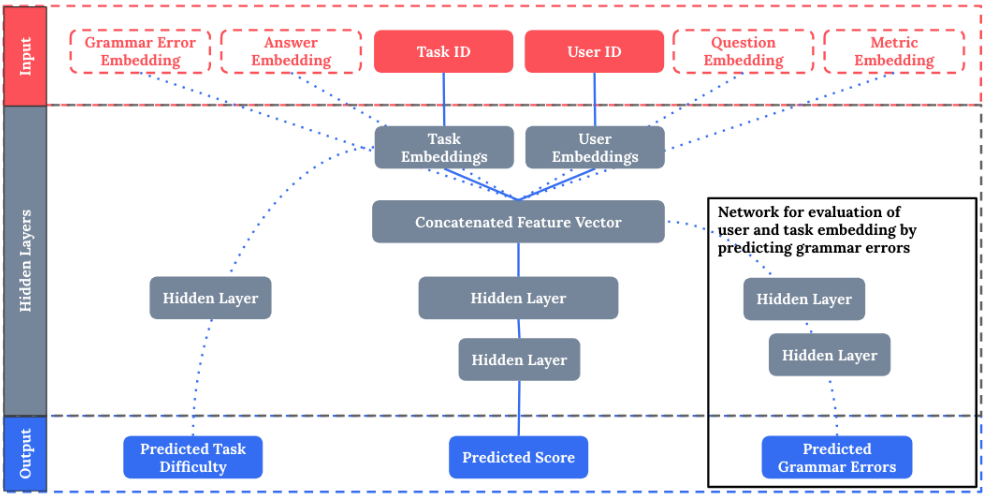


Figure 2: Task score prediction system architecture. Dotted lines and boxes are optional features and network connections.

infer metric embedding values by inspecting this history for each task and across all tasks.

4.3.4 *Grammar error embedding*

A student's grammatical proficiency plays a vital role in determining how well they perform on a particular task. As we do not know of any system that identifies appropriate use of grammar, we focused on understanding what grammatical structures the student struggles with. This was done by running ERRANT [6], an automated error detection and correction system, in order to identify grammatical errors in the student's essay. The text below illustrates an example output from ERRANT.

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

The words highlighted in red are candidates for grammatical errors as detected by the system. The second and third lines are correction suggestions where the first two numerical digits (highlighted in blue) are the token spans for corrections (i.e. where in the sentence the corrections should apply). The strings highlighted in green are the error types (e.g. `R:SPELL`, a spelling error; `R:VERB:SVA`, a subject-verb agreement error on the verb). ERRANT provides error detection and correction outputs on a sentence level.

For each submission k, we constructed a 47-dimensional vector, one dimension for each of the error types observed in the W&I dataset. Each dimension stored the number of times that error type appeared in the student's essay submission.

$$< e_k >=< f_k^1, f_k^2, \ldots, f_k^{47} > \tag{5}$$

– where e_k is the grammar error embedding e for submission k, and f_k^n is the frequency of errors for error type n in submission k.

4.4 Mean score baseline

Our baseline system for predicting s for user u on task t is to calculate the mean of observed scores by all users for that task. We refer to this baseline as MEAN_SCORE.

$$s_u^t = \frac{1}{N_k^t} \sum_k \hat{s}_k^t \tag{6}$$

– where s_u^t is the predicted score for user u on task t, N_k^t is the number of submissions for t, and $\hat{s}_k^t$ is the observed score for submission k on t.

Settles & Meeder showed that predicting the average is a strong baseline in modelling language learning [34] – only 2 out of 4 models outperformed the average. Whilst the authors' work focuses on predicting successful recall and understanding of words, we apply the same principal to the predicting student scores on unseen tasks.

4.5 Evaluation

We identify two approaches to evaluating our system and the quality of our learned user and task representations: 1) score prediction; and 2) grammar error prediction.

4.5.1 *Evaluation of score predictions*

To evaluate the performance of score prediction we use mean squared error (MSE) in common with other works in this field, using global computation where all data points are treated equally [28].

To form our test set, we remove the last score observed by every student from our dataset. The last observed score, instead of a random observed score, was used due to the fact that as the student progresses through the learning material, both the student's knowledge representation and the task representations evolve. Therefore, in order to ensure we are modelling the score that is based on the student's *current* knowledge state, we predict the last observed score for a student on a given task.

4.5.2 Evaluation of grammar embedding predictions

In order to further evaluate the quality of the learned user and task representations, we also introduce an additional evaluation task of predicting the distribution of grammar errors for a user u on a task t.

This was done by building a network that takes as an input the user $\vec{u}$ and task $\vec{t}$ from the pre-trained embedding U and T and predicts the grammar embedding $\vec{g}$. Our dataset for grammar error prediction was created by extracting the last submission k of every user u. This was to ensure that the system is predicting the distribution of errors for the users at their most recent knowledge state. The grammar error embedding prediction model can be defined as follows:

$$\begin{aligned} c &= (\vec{u}, \vec{t}) \\ h_1 &= \mathbf{D}(\sigma(c \cdot W^1)) \\ h_2 &= \mathbf{D}(\sigma(h_1 \cdot W^2)) \\ \vec{g} &= h_2 \cdot W^g \end{aligned} \tag{7}$$

– where c is the concatenated vector of $\vec{u}$ and $\vec{t}$, h_1 and h_2 are the first and second hidden layers, $\mathbf{D}$ is the dropout function and σ is the ReLU activation function. W^1, W^2, W^g are the weight parameters of the model. $\vec{g}$ is the predicted grammar error embedding for user u on task t.

We optimise our system by minimising the cosine proximity of the predicted grammar vector $\vec{g}$ and the target grammar vector $\hat{\vec{g}}$, as in (8).

$$\mathcal{L} = -\frac{\sum_k \vec{g}_k \cdot \hat{\vec{g}}_k}{\sqrt{\sum_k (\vec{g}_k)^2} \cdot \sqrt{\sum_k (\hat{\vec{g}}_k)^2}} \tag{8}$$

– where k is a given submission by the user for a particular task. The more negative the cosine proximity the closer the prediction and target vectors. A value of -1 is a perfect match.

4.6 Implementation

We run our score prediction models for 30 epochs and use an Adam optimiser [14] with a learning rate of 0.001. Both user embedding matrix U and task embedding matrix T were initialised with zero values. In order to identify the right combination of features, we experiment with a variety of feature combinations and identify the ones that provide the greatest reduction in MSE. When evaluating our model at test time, we pass in null vectors for the metric, answer, and grammar error features as the student has, in theory, never attempted the task. Instead, we rely exclusively on the pre-trained user and task representations to make a reliable prediction of the user's score s on task t.

Table 2: Feature dimension sizes. N_h^{min} is the minimum size of the feature or only size where there is no value for N_h^{max}.

Features	N_h^{min}	N_h^{max}
	Score prediction model	
user U	100,140 × 3	100,140 × 32
task T	122 × 3	122 × 32
answer	1 × 300	-
question	1 × 300	-
metric	1 × 2	-
error	1 × 47	-
h_1	1 × 8	-
h_2	1 × 4	-
h_3	1 × 3	-
	Grammar error prediction model	
h_1	1 × 16	-
h_2	1 × 16	-

For our grammar error prediction model we ran 50 epochs with an Adagrad optimiser [10] and learning rate of 0.01. We used a dropout rate of 0.2 for both score prediction and grammar error prediction models.

Table 2 outlines the dimensions used for the various layers of the model. The user and task embedding were tested across a range of dimensions ranging from 3 to 32 dimensions. The justification behind using $n \times 3$ dimension embeddings was to align the size of the embedding with the number of task difficulty levels (beginner, intermediate and advanced). Furthermore, we created a bottleneck[6] in our system in order to learn more meaningful student and task representations [4]. Therefore, we ensured that the upper-bound for the size of our user and task representations was less than 47 – that is, the number of dimensions in the smallest feature vector, the grammar error embedding[7].

5. RESULTS

Table 3 summarises the results of our system. We compare the effectiveness of various features in the prediction of a user's score s on a task t which is evaluated by MSE. We include the top 8 MSE values on the score prediction system and their corresponding cosine value from the grammar error prediction model. Our baseline model MEAN_SCORE achieves an MSE of 1.913.

We find that incorporating question and answer embeddings do not provide any performance improvement in terms of MSE beyond the baseline model. The metric embedding provides marginally better results than the baseline with an MSE of 1.907. The grammatical error embedding provides substantial improvements beyond both the baseline and the metric embedding with an error of 1.761. The best perform-

[6] A bottleneck is where the size of the representation layer is less than the size of the input.

[7] We excluded the metric embedding size as we assumed that an upper bound of 2 would not capture the inherent complexity of language learning.

Table 3: Score prediction (MSE) and grammar embedding prediction (cosine) results for the top 8 best performing feature combinations (error: grammar error embedding; ques: question embedding; ans: answer embedding; metric: metric embedding).

Model	MSE	Cosine
MEAN_SCORE (baseline)	1.913	-
error+ques+ans+metric	2.254	-0.385
ques+metric	1.942	-0.402
ans+metric	1.951	-0.414
error+metric	**1.350**	**-0.426**
ques	2.028	-0.403
ans	2.014	-0.412
error	1.761	-0.410
metric	1.907	-0.393

Table 4: Performance across various student and task representations sizes (N_h)

Model	N_h	MSE	Cosine
error+metric	3	1.350	-0.426
error+metric	5	1.297	-0.431
error+metric	16	1.245	-0.415
error+metric	32	**1.195**	**-0.433**

ing system incorporates both grammatical error embedding and metric embedding, reducing the MSE to 1.350.

The model that provides the lowest cosine proximity to the target grammatical error vector (i.e. best system) was error+metric, which is consistent with the lowest MSE for the score prediction system. We also observe that the system trained on just the answer feature resulted in a cosine proximity of -0.412, an improvement over the system trained on just the grammar error embedding which achieves -0.410. This outcome was unexpected: the system trained on the grammar error embedding resulted in a lower MSE than the system trained on the answer embedding, a representation which by definition contains the grammatical errors but not encoded in the same way. Intuitively the grammar error embedding is a better representation of student knowledge at a given point, which in turn gives us better predictions of task scores.

An important aspect of learning well-formed representations is identifying the correct number of dimensions [4]. Table 4 summarises the various student and task representation sizes we used as part of our system. We set our upper bound at 32 in order to ensure a sufficient bottleneck. The results show that larger representation size improves both score prediction (MSE) and grammar error prediction (cosine).

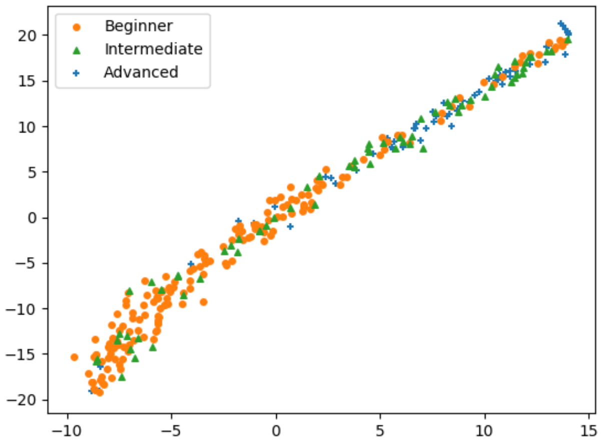


Figure 3: t-SNE of 300 randomly sampled student representations classified by different levels of proficiency

Table 5: Correlation between score prediction MSE and grammar embedding prediction cosine.

Pearson's coefficient	p-value
0.7883	0.0201

In order to interpret the relevance of cosine proximity we conducted a Pearson's correlation test between the MSE values from the score prediction system and the cosine proximity scores from the grammar error prediction system. Table 5 shows the correlation between the score predictions (MSE) and the grammar error prediction (cosine). The results show a 0.7883 Pearson's correlation with a p-value of 0.0201 which is statistically significant at $\alpha < 0.05$.

Figure 3 shows a t-SNE [38] of 300 randomly sampled student representations learned by our best performing score prediction system. The students are classified by their proficiency which has been determined by observing the most frequent task level attempted in their five most recent submissions. Qualitatively, the results from the plot are promising as the advanced and intermediate users, whilst present throughout the plot, are more concentrated towards the top right (higher level of language proficiency). Beginner students, on the other hand, are more concentrated in the bottom left. This suggests that the embeddings constructed from our model provide context on the language abilities of the student.

6. DISCUSSION

The results in Table 3 show that incorporating grammar error embeddings provides a reliable signal to learn well-formed student and task representations. Furthermore, Table 4 identifies the optimal size for student and task representations by training the system using various configurations and evaluating both the MSE and cosine. Larger embedding size performed better than the smaller embedding sizes up to our experimental maximum of 32 dimensions. However, making the embedding size too large would result in what is known as 'overcomplete'[8] which in turn causes

[8] When $N_h > N_x$ (input).

the model to simply memorise the correct response instead of learning discriminative features [4].

In real terms, an MSE of 1.195 represents a root mean squared error of 1.093 on a scale of 0 to 13. This means that on average we stay within the bounds of a CEFR level when predicting student proficiency (since the 0-13 values are mapped to the 6-point CEFR scale), which seems sufficiently robust for real world application. The MSE might mask some more severe errors at the edges, and therefore any downstream use of our user and task representations for ATS would have to be implemented conservatively with reference to model confidence scores.

Grammar errors highlight the weaknesses of the student as opposed to their strengths. Therefore, instead of learning the upper-bound of a student's ability, we are learning the features for the lower-bound. The results of the model also suggest that there is a correlation between the types of errors students make on task t and the score they achieve on said task. This enables the model to learn latent features within the student and task representations which in turn can be used to reliably predict the student's score on a future unseen task.

The importance and value of the signal provided by grammar errors in determining student ability and thus creating quality representations can be further highlighted by Figure 4. The bar-chart shows a comparison between beginner and intermediate students, where the values in x-axis are the various error types in ERRANT and the values for the y-axis are the normalised difference of the frequency for each error type between the two groups of students (positive bars indicate greater frequency of that error type for intermediate students). We can observe that certain errors such as `M:VERB:TENSE` (highlighted in orange) are more frequent with intermediate students. This is not surprising as beginner students tend not to experiment with verb tenses but rather focus on using verb tenses that they are comfortable with. Intermediate students are more likely to learn verb conjugation rules and over-regularise to introduce variation in sentence structure. However, over-regularisation usually results in increased number of verb tense errors [32, 3]. This is then corrected once students reach an advanced level of proficiency where they can account for the irregular verb tenses. We can observe this correction in Figure 5 where advanced students make less verb tense errors than intermediate students.

We also show that a score prediction objective function with a task difficulty prediction auxiliary objective are effective in training well-formed student representations, as evidenced by Table 3, Table 4 and Figure 3. Whilst the plot in Figure 3 generally behaves as expected, we observe some students that are classified as *advanced* but reside towards the bottom-left. We believe this is due to students having *beginner* profiles but attempting *advanced* tasks.

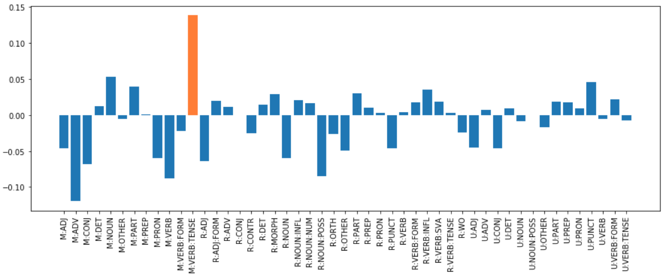


Figure 4: A bar-chart showing a comparison of errors between beginner and intermediate students, where the bars are the normalised difference between the two groups of students. Positive bars indicate that intermediate students made more errors than beginner students for that error type.

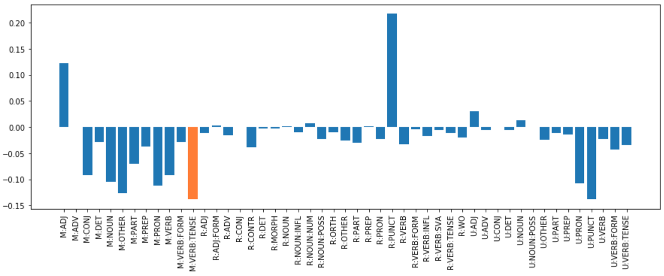


Figure 5: A bar-chart showing a comparison of errors between intermediate and advanced students, where the bars are the normalised difference between the two groups of students. Positive bars indicate that advanced students made more errors for that error type than beginner students.

Whilst grammar error features improved the performance of our system, answer embeddings seemed to reduce the accuracy, counter-intuitive to our understanding of the response as the fundamental indicator of student proficiency. We believe this reflects the fact that the answer text requires several levels of abstraction before it can be transformed into interpretable evidence of language proficiency. We therefore view grammar error embeddings as answer embeddings which have been passed through various levels of processing (in our case, by ERRANT [6]). Without processing, answer embeddings provide only a noisy signal of student ability and negatively impact performance of predictive systems. The answers are relatively small samples of text, perhaps insufficiently so to properly trace language knowledge for the given student. Grammatical errors, on the other hand, appear to be sufficiently robust to short text lengths to provide representative signals of student knowledge.

Question embeddings are faced with a similar limitation. We expressed the hypothesis that the wording of questions would directly indicate task difficulty. However, instead they proved to be the weakest standalone feature (Table 3). We interpret these findings together to mean either that it is how student's perceive and respond to the question that determines the difficulty of the task, or that W&I scores are determined more by grammar errors than answer and question content. Additionally, the content of the question itself does not yield any signal that is discriminative of student ability or task difficulty.

In isolation, the metric embedding also failed to provide a strong signal for student proficiency. However, combined with the grammar error embedding, we noticed significant improvements in performance.

The grammatical error distribution prediction system was introduced to further evaluate the quality of our student and task representations. The purpose of creating that system was to measure the generalisability of the student embeddings and demonstrate their ability to do transfer learning.

Although we do not know of an established gold standard for cosine proximity in our grammar error prediction task, we are able to interpret it in order to compare the performance between the different configurations of our user and task representation learning system. The positive correlation between the score prediction loss and the grammatical error prediction loss further supports our claim that our model architecture and the use of grammatical errors as features are reliable for training student and task representations of language learning. That is, the performance of the model is strong on two tasks, such that we view our representations of students and tasks as sufficiently accurate for further use in downstream educational applications.

7. CONCLUSION

We introduced a novel neural network model to automatically learn student and task representations for language learning by incorporating various features extracted from the W&I dataset and evaluating on score and grammar error prediction. We demonstrated through the results on the score prediction task that the use of grammar error embeddings and metric embeddings in our framework provide a reliable signal for user proficiency in language. These findings were further supported by the cosine proximity score achieved when evaluating the grammar error prediction task.

Learning user and task representations is a central component to enable a truly adaptive learning system. Future work in incorporating aspects such as memory decay and attention can play an important role in further improving the quality of user and task representations. Additionally, this framework may also enable downstream tasks such as curriculum learning in the language learning domain, item similarity [29], and task scheduling through spaced repetition learning [23, 18] .

Alongside the in-principle evaluation metrics we present here, we would then be able to obtain real world evaluation of learning gains for trial groups presented with adaptively selected tasks, compared with control groups who continue to select tasks independently. We propose that the dense representations of users and tasks presented here could underpin an ATS which selects tasks at an appropriate difficulty level for each user with a known submission history on the platform.

ACKNOWLEDGEMENTS

This paper reports on ALTA Institute research supported by Cambridge Assessment, University of Cambridge. We would like to thank Chris Bryant for help with grammar embeddings, Paul Butcher and Henry Garner of English Language iTutoring Limited for access to W&I data, NVIDIA Corporation for the donation of the Titan GPU, Kanako Komiya, Amandla Mabona, Ahmed Fetit for their valuable feedback, and the anonymous reviewers for helpful comments.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] H. Ba-Omar, I. Petrounias, and F. Anwar. A framework for using web usage mining to personalise e-learning. In *Proceedings of the 7th IEEE International Conference on Advanced Learning Technologies*, 2007.

[3] K. Bardovi-Harlig. Tense and aspect in second language acquisition: form, meaning, and use. *Language Learning: A Journal of Research in Language Studies*, 50:1, 2000.

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[5] C. Brinton, R. Rill, S. Ha, M. Chiang, R. Smith, and W. Ju. Individualization for education at scale: MIIC design and preliminary evaluation. *IEEE Transactions on Learning Technologies*, 8:136–148, 2015.

[6] C. Bryant, M. Felice, and T. Briscoe. 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)*, 2017.

[7] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah. Wide and deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS)*, 2016.

[8] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278, 1994.

[9] D. Dahlmeier and H. T. Ng. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012.

[10] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[11] D. Fudholi and H. Suominen. The importance of recommender and feedback features in a pronunciation learning aid. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 83–87. Association for Computational Linguistics, 2018.

[12] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*,

2017.

[13] M. Khajah, R. Wing, R. Lindsey, and M. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Proceedings of Educational Data Mining (EDM)*, 2014.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[15] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.

[16] K. R. Koedinger, S. D'Mello, E. A. McLaughlin, Z. A. Pardos, and C. P. RosÃľ. Data mining and education. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(4):333–353, 2015.

[17] R. V. Lindsey, J. D. Shroyer, H. Pashler, and M. C. Mozer. Improving studentsâĂŹ long-term knowledge retention through personalized review. *Psychological Science*, 25(3):639–647, 2014.

[18] L. Ling and C. W. Tan. Pilot study on optimal task scheduling in learning. In *Proceedings of Learning @ Scale*, 2018.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[20] B. Minaei-Bidgoli, D. A. Kashy, G. Kortemeyer, and W. F. Punch. Predicting student performance: an application of data mining methods with an educational web-based system. In *33rd Annual Frontiers in Education (FIE)*, 2003.

[21] J. Mitchell and M. Lapata. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429, 2010.

[22] S. Montero, A. Arora, S. Kelly, B. Milne, and M. Mozer. Does deep knowledge tracing model interactions among skills? In *Proceedings of the 11th International Conference on Educational Data Mining (EDM)*, 2018.

[23] M. C. Mozer, H. Pashler, N. Cepeda, R. Lindsey, and E. Vul. Predicting the optimal spacing of study:a multiscale context model of memory. In *Advances in Neural Information Processing Systems 22 (NIPS)*, 2009.

[24] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010.

[25] A. S. Najar, A. Mitrovic, and B. M. McLaren. Learning with intelligent tutors and worked examples: selecting learning activities adaptively leads to better learning outcomes than a fixed curriculum. *User Modeling and User-Adapted Interaction*, 26:459–491, 2016.

[26] C. Napoles, K. Sakaguchi, and J. Tetreault. Jfleg: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017.

[27] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[28] R. Pelánek. The details matter: methodological nuances in the evaluation of student models. *User Modeling and User-Adapted Interaction*, 28:207–235, 2018.

[29] R. Pelánek, T. Effenberger, M. Vaněk, V. Sassmann, and D. Gmiterko. Measuring item similarity in introductory programming. In *Proceedings of Learning @ Scale*, 2018.

[30] Y. Rosen, I. Rushkin, R. Rubin, L. Munson, A. Ang, G. Weber, G. Lopez, and D. Tingley. The effects of adaptive learning in a massive open online course on learners' skill development. In *Proceedings of Learning @ Scale*, 2018.

[31] A. Rozovskaya, D. Roth, and M. Sammons. Adapting to learner errors with minimal supervision. *Computational Linguistics*, 43:723–760, 2017.

[32] D. E. Rumelhart and J. L. McClelland. On learning the past tenses of English verbs. In D. E. Rumelhart, J. L. McClelland, and PDP Research Group, editors, *Parallel distributed processing: explorations in the microstructure of cognition, vol. 2.* Cambridge, MA: MIT Press, 1986.

[33] D. Sampson and C. Karagiannidis. Personalised learning: Educational, technological and standardisation perspective. *Interactive Educational Multimedia*, 4:24–39, 2002.

[34] B. Settles and B. Meeder. A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1848–1858, 2016.

[35] S. Shen and M. Chi. Clustering student sequential trajectories using dynamic time warping. In *Proceedings of Educational Data Mining*, 2018.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[37] J. Tarus, Z. Niu, and G. Mustafa. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, 50:21–48, 2018.

[38] L. J. P. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:579–2605, 2008.

[39] J.-J. Vie and H. Kashima. Knowledge tracing machines. In *Proceedings of AAAI*, 2019.

[40] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham. Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation. In *Proceedings of the 9th International Conference on Educational Data Mining (EDM)*, 2016.

[41] H. Yannakoudakis, Ø. E. Andersen, A. Geranpayeh, T. Briscoe, and D. Nicholls. Developing an automated writing placement system for esl learners. *Applied Measurement in Education*, 31:251–267, 2018.

[42] H. Yannakoudakis, T. Briscoe, and B. Medlock. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of*

the Association for Computational Linguistics: Human Language Technologies, 2011.

[43] H. Yannakoudakis, M. Rei, Ø. E. Andersen, and Z. Yuan. Neural sequence-labelling models for grammatical error correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[44] A. H. Zaidi, R. Moore, and T. Briscoe. Curriculum q-learning for visual vocabulary acquisition. In *Proceedings of Visually-Grounded Interaction and Language (ViGIL), NeurIPS*, 2017.