Using Transfer Learning to Automatically Mark L2 Writing Texts

Tim Elks
Research Institute of Information and Language Processing, University of Wolverhampton
Oxford University Press
t.j.elks@wlv.ac.uk, tim.elks@oup.com

Abstract

The use of transfer learning in Natural Language Processing (NLP) has grown over the last few years. Large, pre-trained neural networks based on the Transformer architecture are one example of this, achieving state-of-the-art performance on several commonly used performance benchmarks, often when fine-tuned on a downstream task. Another form of transfer learning, Multitask Learning, has also been shown to improve performance in Natural Language Processing tasks and increase model robustness.

This paper outlines preliminary findings of investigations into the impact of using pre-trained language models alongside multitask fine-tuning to create an automated marking system of second language learners' written English. Using multiple transformer models and multiple datasets, this study compares different combinations of models and tasks and evaluates their impact on the performance of an automated marking system.

1 Introduction and related work

Human marking of learner productive skills is a costly part of the test development process both in terms of direct financial outlay to external markers and internal resource. Assessors are paid for marking candidates' written test responses and attending training while managing the marking process requires recruiting, training, and monitoring assessors, along with ensuring they are paid for the correct number of responses marked. This cost also increases linearly, meaning that as more candidates enrol to take the test, the number of assessors increases, as does the resource required to manage this increased pool of assessors. It therefore does not scale effectively. Further to this, there are also issues regarding reliability of assessors. Despite efforts by test developers to ensure their assessors mark consistently to the given criteria, human markers often demonstrate inconsistencies as shown by research conducted into inter-rater and intra-rater reliability (Coniam and Falvey, 1999).

One way of attempting to address this issue is to implement an automated marking system. This would make a significant difference to test developers who would be able to save significant resource while improving the reliability of their written assessments. Many test developers already use automated marking in their assessment products, such as Linguaskill (Cheung et al., 2017) from Cambridge Assessment, e-rater (Chen et al., 2017) from ETS, and PTE Academic (ins, 2019) from Pearson. It is also an area which has attracted much research (Chen et al., 2010; Briscoe et al., 2010; Phandi et al., 2015; Nguyen and Dery, 2016; Dong and Zhang, 2016; Farag, 2016; Cummins et al., 2016; Cummins and Rei, 2018; Farag and Yannakoudakis, 2019). This has meant different approaches have been used for automated essay marking and have employed a range of different algorithms.

1.1 Traditional methods

Rudner and Liang (2002) conceptualised automated marking as a classification problem, employing Naive Bayes; Chen et al. (2010) adopted an unsupervised clustering approach; and Briscoe et al. (2010) used a linear batch perceptron classifier. This approach could be well suited to classifying test takers on the CEFR (North, 2006), given the discreet categories, although they are ordinal in nature.

Conversely, regression has been used in several studies. Phandi et al. (2015) employed linear regression, as did Yannakoudakis et al. (2018), who did so with as a ranking regression task, though the authors also suggest using other regression algorithms, and Chen et al. (2016) used support vector machines and decision tree regression models.

Proceedings of the Student Research Workshop associated with RANLP-2021, pages 51–57,
held online, Sep 1–3, 2021.
https://doi.org/10.26615/issn.2603-2821.2021_008

1.2 Neural network methods and pre-trained language models

More recently, Neural Network methods have been widely applied to the problem. Nguyen and Dery (2016) and Boulanger and Kumar (2018) both applied deep learning to essay grading and found initial results to be highly competitive with state-of-the-art results using manually extracted features. Both studies used LSTMs (Hochreiter and Schmidhuber, 1997) to achieve this performance, whereas Farag (2016) and Dong and Zhang (2016) adopted CNNs (LeCun et al., 1989) to achieve high performance.

The advent of the Transformer architecture (Vaswani et al., 2017) and pre-trained language models, such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and ELECTRA (Clark et al., 2020), has opened up a new avenue of research in Automated Essay Scoring (AES). Results, however, have been mixed and the field has not seen the large performance increases that appeared in other areas of NLP. Rodriguez et al. (2019) investigated the use of pre-trained models in AES and found that they outperformed traditional machine learning techniques. They also narrowly performed better than an LSTM model, when not as an ensemble of LSTMs – but as a single model. However, the LSTM performed better when combined with others to form an ensemble. Ormerod et al. (2021) found BERT base to outperform LSTMs on an AES task, though not as well as a combined LSTM and CNN model with attention. The author, however, found that the optimum combination was when BERT was combined with manually extracted features. Mayfield and Black (2020), in contrast, found that pre-trained models did not provide any increased performance to n-gram based models but were much more computationally expensive.

1.3 Multitask Learning

Another form of transfer learning, Multitask Learning (MTL), has also been used to tackle the problem of AES, though has received less focus as a research area than pre-trained language models. MTL attempts to improve performance on a given objective by simultaneously training the model to complete a related auxiliary task – the idea being that the model will learn useful information while training on the auxiliary task which will then transfer to the principal task, improving the model's performance (Ruder, 2019). Cummins et al. (2016) took a constrained MTL approach in order to mitigate the need for large volumes of task specific training data. The authors showed that a high performance model was obtainable with very little or no task-specific training data. Further to this, Cummins and Rei (2018) found that by training an automated marker alongside an error detection objective, the automated marker's performance was improved significantly. Farag and Yannakoudakis (2019) applied an MTL approach to coherence modelling and found that by training the model to predict coherence scores for a document while training the same model to predict token level grammatical roles, the model achieved a new state-of-the-art.

One aspect of the MTL literature worth noting is that in researching this paper, only Craighead et al. (2020) used a multitask learning approach which utilized more than a single sub-task. This study took spoken word transcriptions of a speaking test to predict a learners score.

2 Research Questions

The questions that this paper sets out to investigate are:

- Does the use of pre-trained language models improve the performance of an automated marking system of L2 written English?
- Does the use of MTL with pre-trained language models improve the performance of an automated marking system of L2 written English?

3 Task definition

To address the first research question, several models were created in order to achieve suitable comparison. The first model provided a baseline against which the more sophisticated and computationally expensive techniques could be compared. This model was created from manually extracted features, and aimed to be a simplification of the model features described by Yannakoudakis et al. (2011) by using 1-3 grams of tokens, POS and dependency tags, and distances to other grammatical relations. These features were extracted using SpaCy (Honnibal et al., 2020). The task was defined as a standard regression task unlike Yannakoudakis et al. (2011) which used a rank preference approach. These features were then used to train a Random Forest (Breiman, 2001). The second model used the BERT base architecture (Devlin et al., 2018), but with all the model's weights randomly initialized. The third and fourth used BERT base (Devlin et al., 2018) and RoBERTa base (Liu et al., 2019) pretrained models. It was decided not to include an untrained version of RoBERTa because of its architectural similarity to BERT. The Transformers library implementations of the models were used (Wolf et al., 2020). All BERT models used its *uncased* variant.

Each model was trained on a dataset of human-marked written responses to test questions. See below for further information on the dataset used.

3.1 Automated marking task

The primary task in this study is predicting a mark for a written text provided by a learner of English. The data used for this task were taken from the response database of the Oxford Test of English – an English proficiency test developed by the Oxford University Press for learners of English. A subset of responses was sampled so as to achieve an even number of responses as possible across the marking scale, first language groups, gender, and assessor providing the mark. This resulted in a final dataset of 7,596 responses. From this, training, development and test sets were produced. The results reported here were those obtained from predictions on the test set.

The dependent variable, or ground truth, was a score on a scale from 0-21. This is the sum of three marking criteria used on the Oxford Test of English, *Organization*, *Grammar* and *Lexis*. The test uses four criteria in total but *Task Fulfilment*, which rates the extent to which the learner has answered the prompt, was not used because only responses were used for model training.

3.2 Auxiliary tasks

In order to address the second research question, several other tasks were required as subtasks. For this paper, three subtasks were defined:

- an error detection task which required the model to determine whether a sentence was deemed to be acceptable English or not. This was a binary classification task, similar to Cummins and Rei (2018), but at the sentence level as opposed to the token level. This task used the COLA corpus (Warstadt et al., 2018) which consists of 10,000 sentences each tagged 1, 0 depending on their linguistic acceptability.

- an error detection task which required the model to classify tokens by type of error (correct, lexical, grammatical, lexico-grammatical, form, style or missing). This task used a corpus of responses from the Oxford Test of English which had been expertly tagged for their linguistic appropriateness according to a simplified version of the Louvain tagset (Dagneaux et al., 1996).

- a lexis prediction task which required the model to predict the CEFR level of each token. This was chosen in order to train the network to become sensitive to the differences between less and more advanced lexis with regards to learners' linguistic development. This task used the responses from the automated marker task and tagged each token by its CEFR level according to the Oxford 3,000 and Oxford 5,000 word lists. The task was to predict the CEFR level of each token in the response.

Due to the lack of research in training multiple auxiliary tasks as part of an automated marking system, all combinations of the above tasks were trained and the results presented here.

4 Results

4.1 Training and evaluation approach

For each neural model, the same training scheme and training configurations were applied. Each was trained for 5 epochs and was evaluated against the test set at the end of each epoch (the development set was used for preliminary testing and whose results are not reported here). The model which obtained the best RMSE was chosen as the best performing model and whose results are presented here. The metrics presented are the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Spearman's rank correlation. Default model training hyper-parameters were used because no search had been conducted at this stage of the study due to time and processing constraints. The hyper-parameters used are shown in table 1.

4.2 Task results

Table 2 shows the results of the models trained only on the automated marking task to compare the effectiveness of using pre-trained language models for automated marking.

Table 3 shows the results of the models trained with the multitask approach. The RoBERTa pre-

Hyperparameter	**value**
Learning rate	1e-5
Epochs	5
Batch size	16
Gradient accum. steps	1
Weight decay	0
Learning sheduler	linear
Warmup steps	0

Table 1: Comparison of model performance

Model	**MAE**	**RMSE**	**Sp. r**
Manual features	2.79	3.70	.788
BERT not pre	2.82	3.64	.827
BERT pre	2.41	3.35	.892
RoBERTa pre	**2.39**	**3.28**	**.893**

Table 2: Comparison of model performance

trained model was fine-tuned on the primary automarking task alongside the sub-tasks mentioned above. All combinations of tasks were tested in order to verify the impact of each task on the automated marker's performance. *Sent. err* refers to the first auxiliary task which predicts errors at sentence level, *Token. err* refers to the task which predicts error type at the token level, and *CEFR* refers to the task which predicts the associated CEFR level of a particular token.

Tasks	**MAE**	**RMSE**	**Sp. r**
Sent. err.	2.32	3.18	.892
Token err.	2.31	3.12	.892
CEFR	**2.29**	**3.03**	**.893**
Sent. err. + CEFR	2.31	3.09	.889
Token err. + CEFR	2.56	3.49	**.893**
Sent. err. + token err.	2.31	3.09	.889
All tasks	2.38	3.16	.887

Table 3: Comparison of task performance

5 Discussion

The Transformer-based language models performed better than that which used manually extracted features across all the reported metrics. Also, the two models which used pre-trained weights reported superior metrics to those which had not. These results sit in contrast to those found by Mayfield and Black (2020) who reported no performance improvement when using pretrained transformer models. One explanation for this might be the difference in domains. Mayfield and Black (2020) was fine-tuned and tested on five datasets from the ASAP competition (Shermis, 2014) which, though similar in approach, did not mark language learners for their competence in English. This point highlights a critical difference in standard AES and marking the quality of a learners' English. Marking discursive essays requires the assessor to focus on much higher-level textual features compared to that of marking an essay written as part of and ESOL exam. For example, the scoring rubric for ASAP question 1 asks the assessor to focus on whether the *main idea(s) ... stand out*, the text *makes connections and shares insights*, or whether it is *clear, focused or interesting*. This can be contrasted with the Oxford Test of English scoring rubric (OUP, 2019) which focuses on lower-level aspects of the text, requiring responses to demonstrate *a high degree of grammatical accuracy* and *a wide range of cohesive devices*. Understanding such high-level and potentially subjective aspects are likely to be more problematic for language models than issues of grammatical accuracy.

One reason for the increased performance of RoBERTa compared to that of BERT is likely to be the most most obvious differences between the two pre-training approaches. RoBERTa (Liu et al., 2019) is a similar model to BERT but with several key differences: the authors used more training data, they removed the next sentence prediction task, the token sequence length was increased, and the masking pattern applied was dynamically changed during training. This meant that the authors reported improved performance on BERT across the same tasks, mirroring these preliminary results.

Regarding the performance of the MTL models, the vast majority of those which used MTL preformed better than those trained only on the score prediction task. However, the models trained on a single auxiliary task produced higher correlations more consistently than those trained on more than one task, though this trend was not true for the MAE and RMSE metrics, which, showed no obvious trend.

Figures 1 and 2 show scatter plots of the human mark against the automated marker. Although the model shown in figure 1 performed better across the selected evaluation metrics, the model trained on all tasks produced predictions that were much

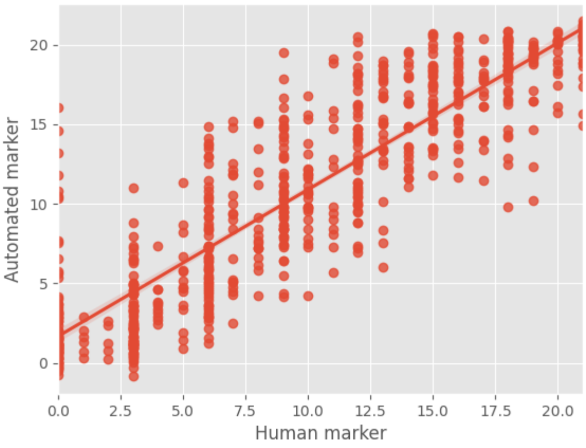

Figure 1: Predictions of MTL model trained with CEFR token level task

more centralized and less erratic. This effect can also be seen in the average standard deviation of predictions for each true score (2.48 for all tasks and 2.56 for the CEFR task alone).

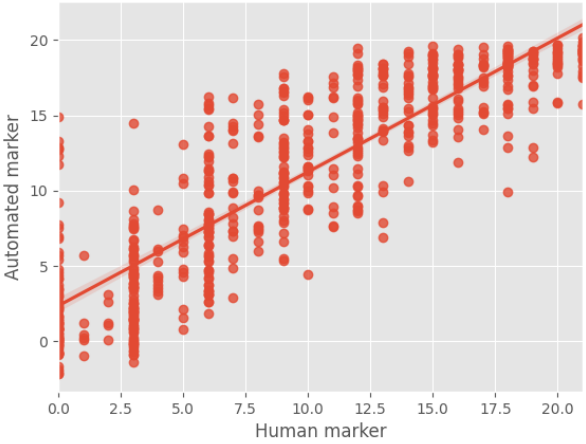

Figure 2: Predictions of MTL model trained with all tasks

Furthermore, when looking at the RMSE for the predictions split by each human mark, the model trained on all tasks was more accurate for all true scores 10-19 but the model trained with only the CEFR auxiliary task was more accurate for all true scores 0-9 and 20-21. This effect can also be seen in figures 1 and 2 where the CEFR trained model made several predictions at the top end of the scale but the model trained on all tasks never predicted the highest possible score.

One potential explanation for this effect is that the model trained on all tasks is not able to differentiate effectively between the higher scores and so clusters many predictions just below the top end of the grade scale in order to be close the true score most frequently. Another explanation might be that as the number of tasks increases, more responsibility is being placed on the model to find a single representation that can be used to predict multiple outputs with only a single transformation in the task specific layer. By increasing the depth of the task specific heads with more transformer layers, this issue might improve, though would not remove the model's dependence on a single representation.

6 Conclusion

Preliminary findings in this research project have shown that pre-trained language models can perform better at automatically grading a learner's English than both traditional methods and the same models without pre-training. The picture regarding the MTL approach is less clear, with the impact of multiple tasks included in training producing opaque results when looking beyond the reported metrics. However, The study has shown that an MTL approach can benefit model performance, but that the impact of using multiple tasks can be unpredictable.

There are several limitations with the study in its current form. Firstly, no model hyperparameter search was conducted to produce the results presented here. This means that some models might perform much better if trained with other hyperparameter values. Another limitation is that it has not explored other prediction methods. For example, as part of the multitask approach, a separate head could be trained to predict each of the three marking criteria as opposed to a total mark. Another limitation and perhaps the most important, is that because the dataset is not available to the wider community, comparison of this approach over others is not possible. One solution to this would be to apply this approach to a freely available dataset such as the Cambridge FCE dataset (Yannakoudakis et al., 2011).

7 Further research

Several issues to be investigated come directly out of the presented findings. Firstly, a more detailed analysis of the predictions made by models trained on multiple tasks compared to those trained on a single task is needed. This could be approached by looking more closely at the distributions of predictions made by the models to see if they do concentrate a much higher rate of predictions just below the top end of the scoring scale in order minimise error without being able to effectively distinguish between texts at those levels. Another way to better understand the performance of the models would be to use another metric, such as a weighted quadratic kappa, which is a more commonly used metric for AES than those presented here and would improve the comparability of results.

Another area of further research would be to introduce manually extracted features and use these alongside MTL training. This approach was demonstrated by Ormerod et al. (2021) to be greatly beneficial for AES systems and would be interesting to introduce with an MTL approach.

Although improvements to automated scoring were seen when trained alongside related tasks, there are other potential benefits from taking such an approach not mentioned in this paper. One such area is providing automated feedback to the learner. Rather than discarding the classification heads for the auxiliary tasks after training, the output predictions of these heads could be used to provide learners with information that could help them improve. For example, the output of the classification head used for the sentence-level error correction task could be used to indicate to the learner which sentences were more likely to contain an error, which they could then address and focus on to improve their writing. Although it would not necessarily be prudent to offer this kind of feedback in a high-stakes proficiency test, this could be very useful as part of a placement test or progress test.

Another benefit of this approach with regards providing learners feedback is that it would simplify the complexity of such a system as it would not require re-training a separate system specifically to perform the feedback task. It would be more simple to replace the final layer of the automarker with a classification head trained on a feedback relevant task. One problem, however, might be that because including more tasks appears to degrade the performance of the model, the more varied forms of feedback the model might give, the greater the reduction in performance of the scoring might be.

References

2019. Pearson Test of English Academic: Automated Scoring. Technical report, Pearson Education Ltd.

David Boulanger and Vivekanandan Kumar. 2018. Deep learning in automated essay scoring. In *International Conference on Intelligent Tutoring Systems*, pages 294–299. Springer.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Ted Briscoe, Ben Medlock, and Øistein Andersen. 2010. Automated assessment of ESOL free text examinations. Technical Report UCAM-CL-TR-790, University of Cambridge, Computer Laboratory.

Jing Chen, James Fife, Isaac Bejar, and André Rupp. 2016. Building e-rater® scoring models using machine learning methods. *ETS Research Report Series*.

Jing Chen, Mo Zhang, and Isaac I Bejar. 2017. An investigation of the e-rater® automated scoring engine's grammar, usage, mechanics, and style microfeatures and their aggregation model. *ETS Research Report Series*, 2017(1):1–14.

Yen-Yu Chen, Chuanjun Liu, Chia-Hoang Lee, and Tao-Hsing Chang. 2010. An unsupervised automated essay scoring system. *IEEE Intelligent Systems*, 25:61–67.

Kevin Cheung, Jing Xu, and Gad Lim. 2017. Linguaskill writing trial report.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

D. Coniam and P. Falvey. 1999. *Melbourne Papers in Language Testing*, 8(2):1–19.

Hannah Craighead, Andrew Caines, Paula Buttery, and Helen Yannakoudakis. 2020. Investigating the effect of auxiliary objectives for the automated grading of learner english speech transcriptions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2258–2269.

Ronan Cummins and Marek Rei. 2018. Neural multitask learning in automated assessment. *arXiv preprint arXiv:1801.06830*.

Ronan Cummins, Meng Zhang, and Edward Briscoe. 2016. Constrained multi-task learning for automated essay scoring. Association for Computational Linguistics.

Estelle. Dagneaux, S. Denness, Sylvain. Granger, and Fanny Meunier. 1996. Error tagging manual.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring–an empirical study. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1072–1077.

Youmna Farag. 2016. *Convolutional Neural Networks for Automated Essay Assessment*. Ph.D. thesis, Master's thesis, University of Cambridge, Computer Laboratory. https://www ....

Youmna Farag and Helen Yannakoudakis. 2019. Multitask learning for coherence modeling. *arXiv preprint arXiv:1907.02427*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Lstm can solve hard long time lag problems. *Advances in neural information processing systems*, pages 473–479.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

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

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.

Huyen Nguyen and Lucio Dery. 2016. Neural networks for automated essay grading. *CS224d Stanford Reports*, pages 1–11.

Brian North. 2006. The common european framework of reference: Development, theoretical and practical issues.

Christopher M Ormerod, Akanksha Malhotra, and Amir Jafari. 2021. Automated essay scoring using efficient transformer-based language models. *arXiv preprint arXiv:2102.13136*.

OUP. 2019. Oxford test of english test specifications.

Peter Phandi, Kian Ming Adam Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *EMNLP*.

Pedro Uria Rodriguez, Amir Jafari, and Christopher M Ormerod. 2019. Language models and automated essay scoring. *arXiv preprint arXiv:1909.09482*.

Sebastian Ruder. 2019. *Neural transfer learning for natural language processing*. Ph.D. thesis, NUI Galway.

Lawrence M Rudner and Tahung Liang. 2002. Automated essay scoring using bayes' theorem. *The Journal of Technology, Learning and Assessment*, 1(2).

Mark D Shermis. 2014. State-of-the-art automated essay scoring: Competition, results, and future directions from a united states demonstration. *Assessing Writing*, 20:53–76.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. 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*, pages 180–189.

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.