How to account for *mispellings*: Quantifying the benefit of character representations in neural content scoring models

Brian Riordan
ETS
briordan@ets.org

Michael Flor
ETS
mflor@ets.org

Robert Pugh[*]
Course Hero
robert.pugh@coursehero.com

Abstract

Character-based representations in neural models have been claimed to be a tool to overcome spelling variation in word token-based input. We examine this claim in neural models for content scoring. We formulate precise hypotheses about the possible effects of adding character representations to word-based models and test these hypotheses on large-scale real-world content scoring datasets. We find that, while character representations may provide small performance gains in general, their effectiveness in accounting for spelling variation may be limited. We show that spelling correction can provide larger gains than character representations, and that spelling correction improves the performance of models with character representations. With these insights, we report a new state of the art on the ASAP-SAS short content scoring dataset.

1 Introduction

Character-based representations have recently been explored in a variety of models in natural language processing, including sequence labeling (Peters et al., 2017) and machine translation (Chen et al., 2018). In educational applications such as content and essay scoring, character-based representations have been claimed to hold promise as a way to account for variation in spelling without resorting to spelling correction (Madnani et al., 2017; Horbach et al., 2017) – particularly in assessments of K-12 populations or English language learners – in part because spelling correction can introduce mistakes from bad corrections. To the extent that character-based representations can in fact help overcome noise from spelling and other errors, they could be a useful component of robust scoring models. For content scoring applications in particular, where scoring rubrics specifically exclude spelling variation from consideration in scoring, it is important that credit is given for the intended words and ideas regardless of spelling.

However, the contributions of character-based representations to automated scoring performance have rarely been systematically studied. To date, no large-scale study of the effect of character representations in real-world scoring scenarios has been carried out. In particular, given the success and proliferation of neural network-based character-based representations in related tasks, there is a need to assess the potential of neural character representations for educational scoring applications.

The rationale for adoption and use of character representations, especially to augment a backbone of word representations in neural models, is typically based on enriching the input representations with morphological information (Peters et al., 2017; Chen et al., 2018), accounting for noise, out-of-vocabulary inputs (Luong and Manning, 2016), or both (Madnani et al., 2017).

We distinguish two main claims that are made for employing character representations in order to account for noise in inputs, sometimes implicitly. One claim is that including character representations in a model accounts for spelling errors in the input. The idea is that models sensitive to characters can implicitly learn the correspondence between incorrect and correct spellings of words from the character-sequence-to-score associations (as opposed to word-to-score associations) across the training data (Horbach et al., 2017). If this is the case, then models without access to character representations should perform more poorly on responses with more misspelled words, since standard word-only neural models ignore these tokens (because the tokens are unlikely to appear in sets of word embeddings and hence are typically

[*] Work carried out while at ETS.

Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 116–126
Florence, Italy, August 2, 2019. ©2019 Association for Computational Linguistics

treated as an unknown token). Therefore, one way to operationalize this claim is the following hypothesis:

- Hypothesis 1: On responses with *more* spelling errors, models with additional character representations should improve model performance relative to models with only word representations. This result should be manifested in a statistical interaction between the addition of character representations to a model and number of misspellings in the input.

A second claim, based on the first claim, is that the addition of character representations to a model's representational repertoire should be sufficient to match the use of spelling correction on the input (without adding character representations). This claim leads to two hypotheses:

- Hypothesis 2.1: Models with additional character representations should achieve performance similar to models without character representations trained on spell-corrected input.

- Hypothesis 2.2: The performance of models with additional character representations should be similar whether or not they are trained on spell-corrected input.

In this paper, we test these hypotheses on a large and diverse collection of content-based questions spanning formative and summative assessments.

We focus on *neural* models for *content* scoring. Content scoring scenarios offer a good testbed for exploring the potential contributions of character-based models because the rubrics of questions focus solely on the content of responses and ignore writing quality metrics such as spelling and mechanics errors. Neural models have seen the most active research on character-based representation and may make possible more flexible and expressive character representations compared with non-neural models. We leave a more general exploration of the contribution of character-based models across both neural and non-neural contexts to future work.

Our work makes the following contributions:

- We demonstrate that, while neural models with additional character representations show a small but durable edge over word-only models in representative real-world contexts, this improvement does not increase significantly as the number of spelling errors increases.

- We show that spell-corrected input improves model performance more than the addition of character representations, and that models with additional character representations can be improved further by using spell correction.

- We achieve a new state of the art on the ASAP-SAS dataset.

2 Related Work

Several recent works provide background on automated content scoring in educational applications (Horbach and Zesch, 2019; Burrows et al., 2015; Riordan et al., 2017). The effect of spelling errors on content scoring was investigated by Horbach et al. (2017). They generated artificial errors on the ASAP-SAS dataset and explored how the scoring performance of a non-neural model of word and character n-grams was affected by increasing amounts of artificial misspellings. They found that models with additional character representations were relatively resilient at higher rates of misspellings. Our work is complementary in that (1) we investigate neural models and (2) we analyze trends in performance on two additional large collections of real-world data.

Spelling correction has been employed in several published systems for the ASAP-SAS dataset. Tandalla (2012), the best-performing system on the ASAP-SAS shared task, employed spelling correction. Kumar et al. (2019) demonstrate strong performance on ASAP-SAS in part due to spelling correction, but use a different train and test set along with data augmentation. Recent work on neural methods for short content scoring uses word- and sentence-level representations (Kumar et al., 2017; Saha et al., 2018; Marvaniya et al., 2018); the current work examines character representations in neural content scoring and explores both short and long content scenarios.

Among neural approaches for essay scoring, Dong et al. (2017) explore a family of combinations of hierarchical CNNs and LSTMs with character-based, word-based, and combined word- and character-based representations. They find that the concatenation of word and character representations does not improve on a word-based representation. Cozma et al. (2018) describe a model that incorporates character information via string kernels. We leave to future work an exploration of the strengths of this and other non-neural character representations for capturing character-sequence-to-score correspondences that account for spelling variation in content scoring.

While spelling correction is often mentioned as a preprocessing step for content scoring, text classification, and other content-focused NLP tasks, to our knowledge, little work exists that attempts to quantify the relative contribution of spelling correction to task performance (although there are indications that general NLP tools such as morphological analyzers can have strong positive effects (Zalmout and Habash, 2017)).

	Mean	SD
Per response	0.920	1.327
Per word	0.040	0.113

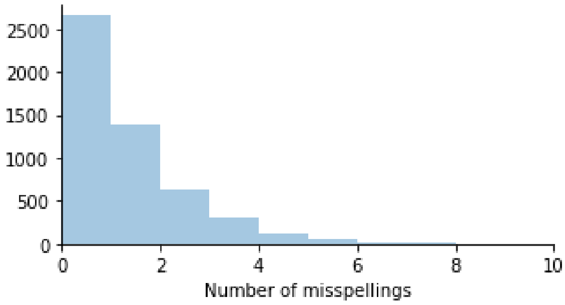

Figure 1: ASAP-SAS spelling errors.

	Mean	SD
Per response	0.414	0.997
Per word	0.026	0.093

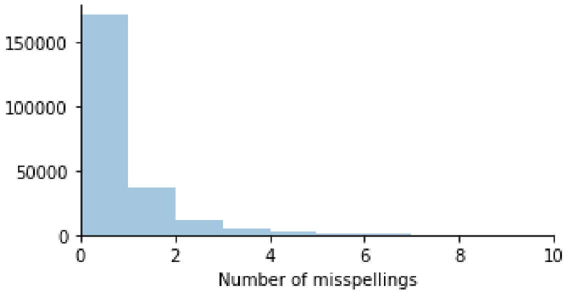

Figure 2: Formative assessments spelling errors.

3 Datasets

Table 1 shows basic statistics for each dataset.

3.1 ASAP-SAS

One of the most widely-used short answer scoring datasets is the Automated Student Assessment Prize Short Answer Scoring (ASAP-SAS) dataset. The dataset is comprised of 10 individual questions on academic subjects such as science, biology, and English Language Arts. The questions were administered to high school students in the United States on state-level assessments. Responses were often one or a few sentences. The responses were scored by two human annotators on a scale from 0 to 2 or 0 to 3 depending on the question (Shermis, 2015). For this study, we used the official training and test data as is without any filtering of responses or manual correction.[1] Figure 1 shows a histogram of the number of misspellings per response (automatically detected). For analysis of the behavior of the models with respect to different numbers of misspellings, we manually binned the number of misspellings per response into `0`, `1`, and `2+` (cf. Figure 5).

3.2 Formative-K12-SAS

We collected a large sample of content-based short answer questions that have been used in a variety of formative classroom settings with middle and high school students. The questions span the subject areas of science, ELA, and social studies. While the questions used different kinds of scoring rubrics with a variety of score ranges, all questions were content-focused. We manually binned the number of misspellings per response (Figure 2) in the same way as was done for the ASAP-SAS dataset (cf. Figure 6).

3.3 Summative-LAS

This dataset is comprised of 20 questions from a series of high-stakes, large-scale standardized tests. The tests are administered to an adult population in the United States, with individuals having completed high school and at least some post-secondary education. Test takers are typically proficient English speakers. Each test measures content knowledge of academic subject areas or elements of effective institutional leadership. Constructed response scores are assigned on a 0–3 scale. Writing proficiency is *not* part of the scoring rubric. Notably, the mean number of words per response is more than 230, making the length of responses comparable to essay questions. Hence, we dub this dataset *Summative-LAS* for *long answer*

[1] `https://www.kaggle.com/c/asap-sas`. The official test set is `public_leaderboard_rel_2.tsv`.

Dataset	Number of questions	Number of responses	Score ranges	Mean number of training responses	Mean number of words (train)
ASAP-SAS	10	22,267	0/1/2(/3)	1363	48.4
Formative-K12-SAS	118	228,909	(0/)1/2(/3/4/5/6)	989	33.0
Summative-LAS	20	108,658	0/1/2/3	4346	233.9

Table 1: Overview of the datasets used in this work. The number of responses covers both the official train and test splits for ASAP-SAS. The mean number of responses and words were computed over the official training set for ASAP-SAS and over 5-fold splits of each question's data (80% train) for the remaining datasets.

	Mean	SD
Per response	3.342	3.836
Per word	0.017	0.019

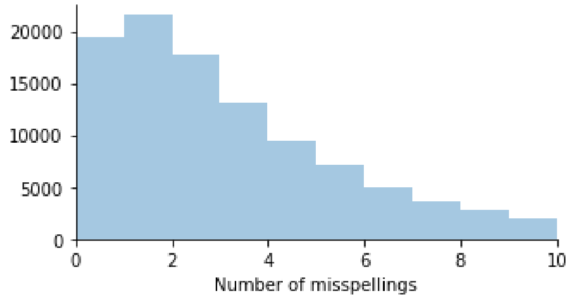

Figure 3: Summative-LAS spelling errors.

scoring. For this dataset, because of the larger spread of spelling errors (Figure 3), we elected to bin the misspellings automatically into relatively equal-sized bins: `0-1`, `2-4`, and `5+` (cf. Figure 7).

4 Method

4.1 Network architecture

The space of network architectures that we explored for this study is depicted in Figure 4. For a word token-only model, pretrained word embeddings are fed to a bidirectional GRU. The hidden states of the GRU are aggregated by a pooling or attention mechanism. Pooling mechanisms included mean and max pooling (Taghipour and Ng, 2016; Shen et al., 2018). The attention mechanism is an MLP-based document-level attention to combined word-character vectors (Yang et al., 2016)[2]. The output of the encoder is aggregated in a fully-connected feedforward layer with sigmoid activation that computes a scalar output for the predicted score.

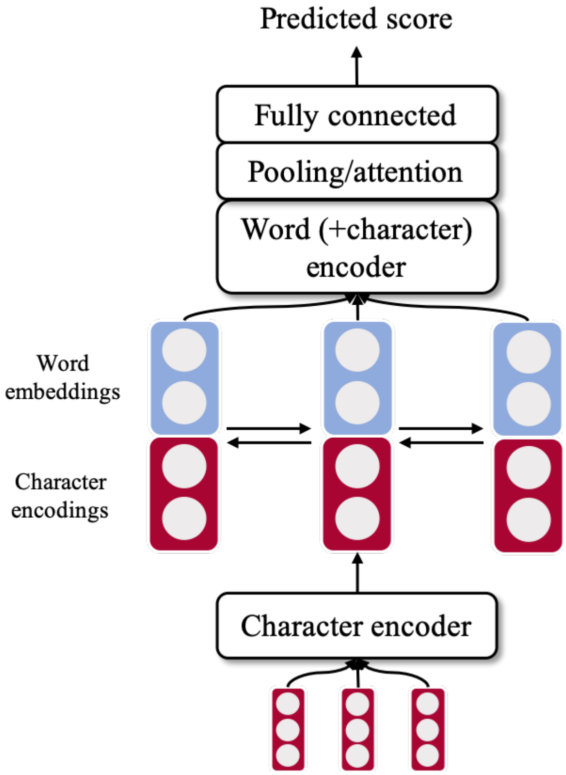

Figure 4: Neural network architectures.

For a model with additional character representations, each word is represented with a sequence of 25-dimensional character embeddings (randomly initialized). These sequences are encoded with a character encoder (see 4.3). The encoded outputs are concatenated with the word embeddings prior to the word-level encoder.

4.2 Data preparation and model training

The text is preprocessed with the spaCy tokenizer with limited custom postprocessing to improve the tokenization outputs. Each response is padded to uniform length, but these padding tokens are masked out during model training. Prior to training, we scale all scores of responses to [0, 1] and use these scaled scores as input to the networks. For evaluation, the scaled scores are converted back to their original range.

[2] A document context vector u is updated at word i with: $u_i = tanh(Wh_i + b)$. The attention is computed with $\alpha_i = exp(u_i^T u)/\sum_t exp(u_i^T u)$ and $d = \sum_t \alpha_i h_i$ for a document d (response) and RNN states h.

For the word tokens, we use GloVe 100 dimension vectors (Pennington et al., 2014) as pretrained embeddings and fine-tune these during training. Word tokens that are not found in the embeddings are mapped to a randomly initialized UNK embedding.

Networks are trained with a mean squared error loss. We carried out extensive preliminary experiments on the ASAP-SAS dev sets to find the highest-performing optimizer (RMSProp with ρ set to 0.9), learning rate (0.001), batch size (32), and gradient clipping setting (10.0).

We employ an exponential moving average of the model's weights during training. These weights w_{EMA} are updated after each batch with

$$w_{EMA} -= (1.0 - d) * (w_{EMA} - w_{current}).$$

d is a decay rate that is updated dynamically at each batch taking into account the number of batches so far:

$$min(decay, \frac{1 + \#batches}{10 + \#batches}).$$

We set *decay*, the maximum decay rate, to 0.999.

For all experiments, we train models with 5-fold cross validation with train/dev/test splits. On the ASAP-SAS dataset, we split the official training data into 5 folds of 80% train and 20% dev. On all other datasets, we split the data into 5 folds of 60% train, 20% dev, and 20% test. For hyperparameter tuning, we evaluate performance only on the dev sets and record the best performance across epochs. For training final models after hyperparameter tuning, we combine the training and dev sets and stop training at the average best epoch across dev folds rounded to the nearest 5th epoch (cf. Johnson & Zhang (2017; 2015)). For ASAP-SAS, final test performance is from the official public test set. For the other datasets, final test performance is the average test performance across folds.

4.3 Hyperparameter tuning

We tuned hyperparameters for both the character and combined word-character encoders. For both encoders, we experimented with several encoder types and hyperparameter configurations on the ASAP-SAS dataset (dev sets only).

For the combined word-character encoder, we varied the encoder type in bidirectional {GRU, LSTM}. Bidirectional GRUs performed better in most cases. We varied the encoder hidden dimensions in {100, 250}, number of layers in {1, 2}, dropout on embeddings in {0.0, 0.3}, pooling/attention in {final state, mean pooling, max pooling, and attention pooling}, and dropout after pooling/attention in {0.0, 0.3}. We obtained the best results on average across the ASAP-SAS questions with 1 layer, 250 dimensions, max pooling, and no dropout.

For the character encoder, we tested a convolutional encoder and three bidirectional recurrent encoders with the same pooling/attention mechanisms: {final state, mean pooling, max pooling, and attention pooling}. For the CNN, we varied the number of filters in {50, 100} and the filter sizes in {3, 5, (3,4,5)}. For the RNNs, we varied the encoder hidden dimensions in {25, 50}. For these experiments, we used a combined word-character encoder with the best hyperparameter settings from the word-character encoder experiments. The best character encoder results were achieved with the CNN with 100 filters and filter sizes of (3,4,5) (i.e. the concatenation of filter sizes 3, 4, and 5) (Johnson and Zhang, 2015).

4.4 Spelling detection and correction

A spelling detection and correction system based on the approach described in Flor (2012) and Flor and Futagi (2012) was used in all experiments. The system employs a set of large-scale dictionaries and language models. The approach demonstrated high spelling correction accuracy on benchmark datasets of essays written on high-stakes summative assessments by both native and non-native English speakers, outperforming comparable industry and open-source spelling correction systems.

For each question in each dataset, we adapted the spelling detection algorithm by incorporating the tokens from the question text. The current work focused on *non-word* misspellings, that is, character sequences that are not valid in standard written English. We leave an examination of real-word (context-sensitive) errors (e.g., confusing *their* and *there*) to future work.

4.5 Evaluation and statistical analysis

To summarize model performance, we report mean squared error (MSE) and quadratic weighted kappa (QWK). For the ASAP-SAS dataset, we also report the Fisher-weighted mean QWK across questions, which was the official metric of the ASAP competition.

To analyze the robustness of performance improvements with character representations, we employ generalized linear mixed-effect models (GLMMs) (Harrison et al., 2018). Mixed-effect models can better capture variation across individual questions by modeling questions as random effects. In contrast with previous work in NLP that analyzes model performance with mixed-effect models, we analyze per-response prediction errors using real-valued regression model predictions. Since the prediction errors are not normally distributed, using standard linear mixed effect models (even with transformation of the dependent variable) can result in Type I errors. Analysis of the prediction error data showed that gamma distributions provided the best fit. Hence we employ gamma GLMMs with a log link function.

We investigated the interaction predicted by Hypothesis 1 with the following GLMM:

$$error \sim feat * missp + \#words + score + (1|question) \quad (1)$$

`feat` is the representation type (*w* vs. *w+c*), `missp` is the misspelling bin, and `feat*missp` is their interaction. `#words` is the number of words in the response, and `score` is the response's human-assigned score. `(1|question)` represents a random intercept for each question. This model estimates the effect of the representation type and the number of misspellings and their interaction, while controlling for the effect of number of words and assigned score.

Hypothesis 2 was examined with a GLMM model of the form:

$$error \sim feat * sp + (1|question) \quad (2)$$

where `sp` is the presence or absence of spelling correction.

For each dataset, we address Hypothesis 2 first, since the evidence relating to this hypothesis is the relative performance of the different models. Then, looking at model predictions by bins of responses for numbers of misspellings, we examine evidence for Hypothesis 1.

Condition	**Mean MSE**	**Mean QWK**	**Mean$_{\text{Fisher}}$ QWK**
w -sp	0.2286	0.7562	0.7652
w+c -sp	0.2218	0.7602	0.7691
w +sp	0.2236	0.7660	0.7748
w+c +sp	**0.2200**	**0.7705**	**0.7788**

Table 2: Human-machine agreement across models on ASAP-SAS. *w* = word representations, *w+c* = word and character representations, *-sp* = no spelling correction, *+sp* = spelling correction.

	Estimate	**SE**	**Pr(>|z|)**
(Intercept)	-1.132	0.119	<2e-16
feature set (*w+c*)	-0.028	0.020	0.168
spelling (+sp)	-0.017	0.020	0.389
feature set : spelling	0.023	0.029	0.423

Table 3: GLMM parameter estimates, standard errors, and *p*-values for model prediction error across all models on ASAP-SAS. *Feature set* is *w* vs. *w+c*. *Spelling* is +/- spelling correction.

5 Results

5.1 ASAP-SAS

Table 2 shows the mean MSE, mean QWK, and mean Fisher-transformed QWK across the 10 questions in the ASAP-SAS dataset. First, we see that the models with character representations outperform their word-only counterparts (*w+c* vs. *w*; lower MSE and higher QWK). Second, the spell-corrected models outperform the corresponding uncorrected models (+sp vs. -sp) with the same representations. The spell-corrected model with character representations achieves the highest performance. The Fisher-transformed mean QWK of 0.7788 represents a new state of the art for the ASAP-SAS dataset for the official test set for single models without data augmentation.[3]

[3] Ramachandran et al. (2015) report a QWK of 0.78 on the ASAP-SAS dataset, but we conclude that their actual unrounded Fisher-transformed mean QWK score was 0.77696. As they note, "The mean QW Kappa achieved by our patterns is 0.78 and that achieved by Tandalla's manual regular expressions is 0.77. Although the QW Kappas are very close... their unrounded difference of 0.00530 is noteworthy." According to the Kaggle public leaderboard (https://www.kaggle.com/c/asap-sas/leaderboard), the Tandalla system's unrounded score was 0.77166. Combining this information: 0.77166 + 0.00530 = 0.77696. Moreover, elsewhere in their paper Ramachandran et al. note "The human benchmark for the dataset was 0.90. The best team achieved a score of 0.77." Because these scores match the Fisher-transformed QWK scores on the Kaggle leaderboard, we conclude that they used the Fisher-transformed mean QWK as opposed to the untransformed mean QWK.

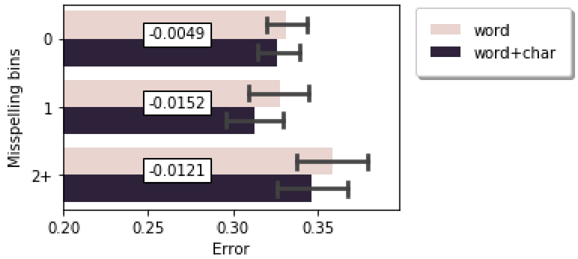

Figure 5: Mean prediction error by models without spell correction on ASAP-SAS. Numbers on the bars represent the difference between *w+c* and *w*.

	Estimate	SE	Pr(>|z|)
(Intercept)	-1.681	0.111	<2e-16
feature set (*w+c*)	-0.018	0.028	0.506
missp 1	0.050	0.034	0.143
missp 2+	0.138	0.037	0.0002
# words	1.947	0.142	<2e-16
score	0.651	0.037	<2e-16
feat (*w+c*) : missp 1	-0.036	0.048	0.454
feat (*w+c*) : missp 2+	-0.022	0.051	0.656

Table 4: GLMM summary for model prediction error on ASAP-SAS for the models without spelling correction. *Feature set* is *w* vs. *w+c*. *Missp* $\{1,2+\}$ are bins of number of misspellings. *Score* is human-assigned response score.

With regard to Hypothesis 2.1, that character representations should improve performance as much as spell correction, the results demonstrate that adding character representations (*w+c*, -sp: mean MSE = 0.2218) can outperform spell correction of a word-only model (*w*, +sp: mean MSE = 0.2236) (although this is not reflected in the QWK results).

To test the strength of these results, we used the GLMM from equation (2). The model parameter estimates are shown in Table 3. Neither the effect of adding character representations (*w+c*) nor the effect of spelling correction (*+sp*) are statistically significant. Notably, there is no evidence for an interaction between character representations and spelling correction, suggesting relatively independent effects.

Next, we examine Hypothesis 1, that character representations should aid performance more on responses with more spelling errors. Figure 5 shows the mean error across all responses in ASAP-SAS by number of spelling errors in bins of `0`, `1`, and `2+` for the models without spelling correction (*w* -sp and *w+c* -sp).

The mixed effect model parameter estimates are presented in Table 4. The main result for our investigation is that there is no significant interaction between model type and number of spelling bins. In other words, the *w+c* models' performance did not significantly improve as the number of misspellings increased [4].

Condition	Mean MSE	Mean QWK
w -sp	0.3220	0.7759
w+c -sp	0.3190	0.7799
w +sp	0.3176	0.7815
w+c +sp	**0.3140**	**0.7828**

Table 5: Human-machine agreement across models on Formative-K12-SAS.

	Estimate	SE	Pr(>|z|)
(Intercept)	-0.962	0.024	<2e-16
feature set (*w+c*)	-0.010	0.002	0.0005
spelling (+sp)	-0.011	0.002	6.58e-05
feature set : spelling	0.001	0.004	0.643

Table 6: GLMM parameter estimates, standard errors, and *p*-values for model prediction error across all models on Formative-K12-SAS.

5.2 Formative-K12-SAS

The performance of the neural models on the Formative-K12-SAS dataset are shown in Table 5. The same trends that were observed for ASAP-SAS are observed here: (1) character and word representations outperform word representations alone (*w+c* vs. *w*); (2) spell-corrected models outperform models without spell correction (+sp vs. -sp); (3) the spell-corrected model with character and word representations performs best. Moreover, on this dataset, the mean MSE and mean QWK trends are consistent.

Applying the statistical model from equation (2) to the prediction errors on all responses in this large dataset (Table 6), both model representations and spelling correction achieve statistical significance. No interaction was observed between representation type and spelling correction.

To analyze the differences between model representations by number of misspellings, we specified 3 bins: `0`, `1`, and `2+`. This was because of

[4] The data for question 10 in the ASAP-SAS dataset suffers from preprocessing issues such that random spaces are introduced between many words. As a result, a much higher number of misspellings are detected for this question. However, refitting the GLMM model excluding the data for this question produced nearly identical trends.

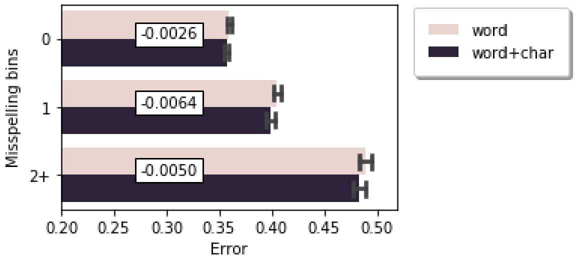

Figure 6: Mean prediction error by models without spell correction on the Formative-K12-SAS dataset. Numbers on the bars represent the difference between *w+c* and *w*.

	Estimate	SE	Pr(>|z|)
(Intercept)	-1.108	0.023	<2e-16
feature set (*w+c*)	-0.008	0.003	0.017
missp 1	0.077	0.005	<2e-16
missp 2+	0.158	0.007	<2e-16
# words	1.996	0.157	<2e-16
score	0.230	0.008	<2e-16
feat (*w+c*) : missp 1	-0.009	0.008	0.242
feat (*w+c*) : missp 2+	-0.004	0.010	0.703

Table 7: GLMM summary for model prediction error on Formative-K12-SAS for the models without spelling correction.

the extreme skew in the misspellings counts – the large majority of responses actually had no misspellings – which precluded specifying bins with a similar number of responses. The model mean prediction error increased across misspellings bins for both w and *w+c* models (Figure 6). Unlike ASAP-SAS, both the difference between feature sets and between misspellings bins *was* significant even when controlling for score and number of words (Table 7). As before, however, there was not a significant interaction between misspelling bins and representation type.

5.3 Summative-LAS

Table 8 provides the MSE and QWK for the dataset of content-based questions on the summative assessment dataset. As in the other two datasets, character and word representations (*w+c*) perform best, and the best models are the models based on spell-corrected text. On this dataset, however, what is striking is the degree to which spelling correction improves model performance: QWK scores increase about 15 points.

Condition	Mean MSE	Mean QWK
w -sp	0.4768	0.5082
w+c -sp	0.4766	0.5115
w +sp	0.3457	0.6590
w+c +sp	**0.3441**	**0.6609**

Table 8: Human-machine agreement across models on Summative-LAS.

	Estimate	SE	Pr(>|z|)
(Intercept)	-0.763	0.031	<2e-16
feature set (*w+c*)	-0.002	0.003	0.479
spelling (+sp)	-0.006	0.003	0.051
feature set : spelling	0.003	0.004	0.508

Table 9: GLMM parameter estimates, standard errors, and *p*-values for model prediction error across all models on Summative-LAS.

	Estimate	SE	Pr(>|z|)
(Intercept)	-0.544	0.032	<2e-16
feature set (*w+c*)	-0.002	0.005	0.706
missp 2-4	0.027	0.005	6.24e-07
missp 5+	0.054	0.006	<2e-16
# words	0.228	0.022	<2e-16
score	-0.478	0.005	<2e-16
feat (*w+c*) : missp 2-4	9.09e-05	0.007	0.991
feat (*w+c*) : missp 5+	-0.001	0.008	0.856

Table 10: GLMM summary for model prediction error on Summative-LAS for the models without spelling correction.

The GLMM parameter estimates (Table 9) show that the difference between models with and without spell correction nearly reaches the 0.05 threshold of significance, underlining the strength of the effect of building models on spell corrected text on this dataset. The addition of character representations, on the other hand, shows a negligible effect on model performance.

The mean prediction error for the w and *w+c* models is shown in Figure 7. The results of mod-

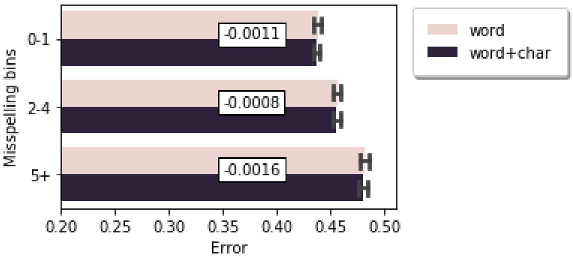

Figure 7: Mean prediction error by models without spell correction on the Summative-LAS dataset. Numbers on the bars represent the difference between *w+c* and *w*.

eling the prediction errors with the model from equation (1) with these bins are given in Table 10. The mean prediction error increases significantly from bin `0-1` to `5+`, but there is little difference between representation types and there is no interaction between representation type and misspelling bin.

6 Discussion

This study is the first large-scale examination of the contribution of character representations in neural network models for automated content scoring. We formulated three hypotheses about the effects of adding character representations to neural models and tested these hypotheses with three diverse datasets, including two large-scale real-world datasets. The results provide several new insights into the capabilities of character representations for content scoring.

First, we examined whether the addition of character representations improves scoring model performance as the number of spelling errors increases. If a model were to effectively learn character-to-score correspondences, we might expect the model to show solid gains on responses with more misspellings. While there was a small trend toward an improvement in word+character models over word-only models on such responses, this trend was not strong enough to produce a statistically significant difference between model representation types. Hence, we cannot conclude that character representations readily account for spelling variation in the training data.

Second, we showed that spelling correction can increase word-only model performance beyond what is achieved with only the addition of character representations (without spelling correction). This trend was strongest in the data with the most spelling errors (Section 5.3). Moreover, we showed that spelling correction can boost the performance of models with character representations. In fact, leveraging spelling correction and character representations contributed to establishing a new state-of-the-art result on the ASAP-SAS official test set. While both trends were not statistically significant given the variability in the prediction error data, neither of these trends are predicted by common ideas about the effectiveness of character representations in automatically learning how spelling variants correlate with scores.

We note that our results do not establish that models with character representations do not learn about some associations between spelling variation and scores. It may be the case that larger training data would lead to more effective learning of the association between character sequence variants and scores. However, large datasets are generally not typical in training data for educational applications. Different kinds of character (or subword) representations may also prove more effective than the space of character representations considered here.

Our results show that character representations, when added to word-based neural models, consistently provide small gains in performance. Therefore, we conclude that character representations may provide some benefit in practice in neural models for content scoring, but that they are unlikely to serve as a replacement for spelling correction of the training data.

Acknowledgements

We thank Aoife Cahill, Anastassia Loukina, Beata Beigman Klebanov, Nitin Madnani and three anonymous reviewers for their helpful comments.

References

Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The Eras and Trends of Automatic Short Answer Grading. *International Journal of Artificial Intelligence in Education*, 25(1):60–117.

Huadong Chen, Shujian Huang, David Chiang, Xinyu Dai, and Jiajun Chen. 2018. Combining Character and Word Information in Neural Machine Translation Using a Multi-Level Attention. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Madalina Cozma, Andrei Madalin Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In *Conference on Natural Language Learning (CoNLL)*.

Michael Flor. 2012. Four types of context for automatic spelling correction. *Traitement Automatique des Langues (TAL)*, 53(3):61–99.

Michael Flor and Yoko Futagi. 2012. On using context for automatic correction of non-word misspellings in student essays. In *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Xavier A. Harrison, Lynda Donaldson, Maria Eugenia Correa-Cano, Julian Evans, David N. Fisher, Cecily E. D. Goodwin, Beth S. Robinson, David J. Hodgson, and Richard Inger. 2018. A brief introduction to mixed effects modelling and multi-model inference in ecology. *PeerJ*.

Andrea Horbach, Yuning Ding, and Torsten Zesch. 2017. The Influence of Spelling Errors on Content Scoring Performance. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA)*.

Andrea Horbach and Torsten Zesch. 2019. The Influence of Variance in Learner Answers on Automatic Content Scoring. *Frontiers in Education*, 4:28.

Rie Johnson and Tong Zhang. 2015. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Rie Johnson and Tong Zhang. 2017. Deep Pyramid Convolutional Neural Networks for Text Categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sachin Kumar, Soumen Chakrabarti, and Shourya Roy. 2017. Earth Movers Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading. In *International Joint Conferences on Artificial Intelligence (IJCAI)*.

Y. Anoop Kumar, Swati Aggarwal, Debanjan Mahata, Rajiv Ratn Shah, Ponnurangam Kumaraguru, and Roger Zimmermann. 2019. Get IT Scored using AutoSAS-An Automated System for Scoring Short Answers. In *9th Symposium on Educational Advances in Artificial Intelligence (EAAI-19)*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nitin Madnani, Anastassia Loukina, and Aoife Cahill. 2017. A Large Scale Quantitative Exploration of Modeling Strategies for Content Scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Smit Marvaniya, Swarnadeep Saha, Tejas I. Dhamecha, Peter Foltz, Renuka Sindhgatta, and Bikram Sengupta. 2018. Creating Scoring Rubric from Representative Student Answers for Improved Short Answer Grading. In *Conference on Information and Knowledge Management (CIKM)*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lakshmi Ramachandran, Jian Cheng, and Peter Foltz. 2015. Identifying Patterns For Short Answer Scoring Using Graph-based Lexico-Semantic Text Matching. In *Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chong Min Lee. 2017. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Swarnadeep Saha, Tejas I Dhamecha, Smit Marvaniya, Renuka Sindhgatta, and Bikram Sengupta. 2018. Sentence Level or Token Level Features for Automatic Short Answer Grading?: Use Both. In *19th International Conference of Artificial Intelligence in Education (AIEd)*.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mark D Shermis. 2015. Contrasting state-of-the-art in the machine scoring of short-form constructed responses. *Educational Assessment*, 20(1).

Kaveh Taghipour and Hwee Tou Ng. 2016. A Neural Approach to Automated Essay Scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Luis Tandalla. 2012. Scoring short answer essays. Technical report.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Nasser Zalmout and Nizar Habash. 2017. Don't Throw Those Morphological Analyzers Away Just

Yet: Neural Morphological Disambiguation for Arabic. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

A Appendix: ASAP-SAS detailed results

	Dataset-general tuning			
Prompt	*w* -sp	*w+c* -sp	*w* +sp	*w+c* +sp
1	0.8222	0.8310	**0.8339**	0.8301
2	0.7802	**0.8017**	0.7857	0.7913
3	0.6443	0.6311	0.6577	**0.6620**
4	0.7044	0.6934	0.7120	**0.7310**
5	0.8285	0.8272	0.8355	**0.8441**
6	0.8562	0.8477	**0.8625**	0.8610
7	0.7060	0.7250	0.7115	**0.7362**
8	0.6510	0.6662	**0.6778**	0.6641
9	0.8045	0.7942	**0.8178**	0.8087
10	0.7645	**0.7847**	0.7650	0.7766
Mean QWK	0.7561	0.7602	0.7659	**0.7705**
Mean $\text{QWK}_{\text{Fisher}}$	0.7652	0.7691	0.7748	**0.7788**

Table 11: Human-machine agreement on ASAP-SAS by prompt.

In Table 11 we report the performance of each prompt's model on ASAP-SAS. We used *dataset-general tuning* of hyperparameters by considering the average best performance across all prompts.