The Effect of Adding Authorship Knowledge in Automated Text Scoring

Meng Zhang[1], Xie Chen[2], Ronan Cummins[3], Øistein Andersen[1], and Ted Briscoe[1]

[1,3]ALTA Institute, Department of Computer Science and Technology, University of Cambridge, UK
[1]{*mz342, oa223, ejb1*}*@cam.ac.uk*
[3]*Ron.Cummins@gmail.com*
[2]Department of Engineering, University of Cambridge, UK
[2]*xc257@cam.ac.uk*

Abstract

Some language exams have multiple writing tasks. When a learner writes multiple texts in a language exam, it is not surprising that the quality of these texts tends to be similar, and the existing automated text scoring (ATS) systems do not explicitly model this similarity. In this paper, we suggest that it could be useful to include the other texts written by this learner in the same exam as extra references in an ATS system. We propose various approaches of fusing information from multiple tasks and pass this authorship knowledge into our ATS model on six different datasets. We show that this can positively affect the model performance in most cases.

1 Introduction

The existence of various English exam products provides a useful and fair way for language learners to measure their English skills accurately. It also offers a well-accepted standard to help schools and companies to quantitatively judge whether their non-native English applicants meet the compulsory language requirements they set up. Many learners have taken different English exams to get the qualifications required by different organisations. For example, more than two million International English Language Testing System (IELTS) exam sessions have been taken in 2012-2013[1], and more than 30 million people have taken the Test of English as a Foreign Language (TOEFL) exam[2].

English exams like IELTS and TOEFL have free-text writing tasks to evaluate a learner's writing ability. For a writing task, each learner needs to write a text to answer the prompt in the task. Appropriately assessing the quality of free-text writings requires highly proficient human examiners, and the lack of professional and qualified examiners makes it hard for learners to get accurate feedback on the quality of their writings in a timely fashion. Consequently, it is hoped that an ATS system can possibly act as a kind of examiner to mitigate this problem, which offers an assistance to both learners and educators. The goal of ATS is to improve consistency and reduce human resource overheads. ATS usually utilises machine learning techniques to build a model to learn the underlying relationship between texts and scores. ATS is often used as the second marker in high-stakes exams, the only marker in practice and tutoring software products.

1.1 Multiple Writing Tasks

To evaluate a learner's writing skill more thoroughly, many English exams like IELTS and TOEFL ask them to answer multiple writing tasks. These tasks are drawn from different topics and genres, and each learner is required to write a text for each task. In practice, human judges score each text with an **individual score**, and these scores are aggregated to obtain an **overall score**, which reflects their writing skills. We also define the ATS model predicting the individual score of a text and the overall score of all the texts as the **individual-level** and **overall-level** models, respectively.

When an individual-level ATS model scores texts, previous work has made an implicit assumption that all responses to all tasks are composed independently. This is not true for exams requiring responses to multiple tasks. The writing skill exemplified by a learner during the same exam session will not normally vary greatly, so the texts written by one learner may share some commonalities, such as preferential word usages and common mistakes, and should also approximately

[1]https://www.britishcouncil.org/organisation/press/record-two-million-ielts-tests
[2]https://www.ets.org/toefl/ibt/about

Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 305–314
New Orleans, Louisiana, June 5, 2018.

equally reflect their writing skills. We suggest that when an individual-level model predicts the score of a text written by a learner, it is helpful to use their other texts as a reference and pass it as an extra piece of information to the model. We refer to this information as **authorship knowledge**.

We suggest that the potential benefit of passing this authorship knowledge to an ATS model might come from a reduction of data sparsity and improvement in the robustness and reliability of feature extraction. Normally the text length for each task is limited, and so there may be insufficient features exemplified in a single response to differentiate language proficiency levels. It can be challenging for an ATS model to learn the mapping between texts and scores accurately, and adding authorship knowledge might provide additional salient features to learn the mapping.

In this paper, we test the hypothesis that authorship knowledge can improve individual-level model performance. We pass this authorship knowledge to an individual-level model in two independent ways: feature fusion and score fusion. When the model predicts text scores, both methods pass all the texts written by the same learner to the model as an extra reference. It is shown that adding this knowledge is helpful in an individual-level ATS model in most cases. To the best of our knowledge, this is the first study that studies how authorship knowledge affects ATS system performance.

2 Related Work

In most previous work, text features are defined manually and automatically extracted from each text. A machine learning model is then applied to learn the mapping from features to scores. Many different machine learning models have been used, including regression (Page, 2003; Attali and Burstein, 2006; Phandi et al., 2015), classification (Larkey, 1998; Rudner and Liang, 2002) and ranking (Chen and He, 2013; Cummins et al., 2016b). The features used in previous work range from shallow textual features to discourse structure and semantic coherence (Higgins et al., 2004; Yannakoudakis and Briscoe, 2012; Somasundaran et al., 2014), and from prompt independent to dependent features (Cummins et al., 2016a). Some recent models have dispensed with feature engineering and utilised word embeddings and neural networks (Alikaniotis et al., 2016; Dong and Zhang, 2016; Taghipour and Ng, 2016).

However, no previous work has investigated the utility of authorship knowledge in ATS. One possible reason is that most datasets only have one text written by each learner. The First Certificate in English (FCE) dataset released by Yannakoudakis et al. (2011), on the other hand, contains two texts per learner. We primarily focus on the FCE dataset in this paper, but also utilise other datasets to corroborate our results. Yannakoudakis et al. defined all the texts written by a learner as a **script**. They extracted features from each text and then combined the features of the two texts within the same script together. A support vector machine (SVM) ranking model was trained to learn the relationship between features and overall scores.

3 Datasets

In this paper, we require a dataset that includes more than one text written by each learner, where each text is scored with an individual-level score. We finally get six datasets in total for our experiments. Each dataset is a set of texts collected from a real exam, and each exam is targeted at one or more Common European Framework of Reference for Languages (CEFR) [3] levels in English. There are six CEFR levels in total: A1, A2, B1, B2, C1 and C2 arranged from lowest to highest.

In each dataset, each script consists of the answers to two tasks. The answers to both tasks were scored on the same grading scale. Each script was written on the same day so we can safely assume no dramatic variation in the writing skill for each learner. The FCE dataset discussed in Section 2 was collected from the FCE exam. The other five datasets were provided by Cambridge Assessment collected from different years.

We need to choose the score for each text for an ATS model to learn. As the original score for each text in the FCE is not reported on a numerical scale, Cambridge Assessment helped us convert the grades to integers between 0 and 20. This mapping is available in Table 2. All the texts from the B2-U, B2-S, C1-U and C1-S datasets are evaluated in terms of four aspects: content, communicative achievement, language quality and organisation. Each aspect is scored as an integer in the range 0-5. We add the scores of these four aspects of a text together to obtain a total score in the range 0-20, and we use this total score as the score for

[3] http://www.coe.int/t/dg4/linguistic/Cadre1_en.asp

Exam	CEFR	Score Range	MEAN	STD	# prompts	# scripts	# train	# dev	# test
FCE	B2	0-20	13.92	2.92	31	1212	822	293	97
B2-U	B2	0-20	14.51	2.18	37	2047	1447	300	300
C1-U	C1	0-20	13.20	2.69	50	2088	1488	300	300
AL-U	A1-C2	0-9	5.78	0.96	58	1604	1004	300	300
B2-S	B2	0-20	13.72	2.41	67	6584	5984	300	300
C1-S	C1	0-20	12.77	2.73	35	1910	1310	300	300

Table 1: The details of the six datasets. FCE is the dataset released by Yannakoudakis et al.. For the other five datasets, the name of each dataset encodes its target CEFR level learners with whether it is **unshuffled** or **shuffled**. B2-U means that it aims at **B2** level learners and is **unshuffled**. MEAN and STD describe the mean and standard deviation of the scores. All the datasets have two writing tasks, and for each writing task, each learner is required to write an answer to one prompt. # prompts describes how many prompts exist in each dataset.

this text for our study. In contrast, AL-U is marked on a scale of 0-9 at 0.5 mark intervals, where each text also receives a score for each of four aspects including task achievement, coherence, word usage and grammar. The total score is aggregated from the scores on all four aspects by Cambridge Assessment, and it is still normalised to 0-9 at 0.5 mark intervals. In this case, we directly use the existing total score as the individual score for a text in AL-U for our study.

Original → New	Original → New
0,0 → 0	3,2 → 13
1,1 → 1	3,3 → 14
1,2 → 4	4,1 → 15
1,3 → 7	4,2 → 16
2,1 → 9	4,3 → 17
2,2 → 10	5,1 → 18
2,3 → 11	5,2 → 19
3,1 → 12	5,3 → 20

Table 2: The score mapping in the FCE dataset

We summarise the six datasets in Table 1. The difference between the shuffled and unshuffled datasets in Table 1 is how texts are presented to human judges to score. For the four unshuffled datasets, each human judge marks the first and second text written by a learner in sequence, so the score of the second text might be affected by the first marked text. In comparison, the texts in B2-S and C1-S are shuffled and randomly displayed to human judges. Hence, this removes any grading bias due to knowing the authorship.

Due to transcription errors, we only kept scripts which do not contain any invalid individual score. After we cleaned the text scores, each dataset was then split into training, development and test sets. The total number of scripts in each dataset, and the number of scripts in the training, development and test sets are summarised in Table 1. The test set for FCE is the same in Yannakoudakis et al. (2011).

4 Notations

Let us introduce some notations to facilitate our discussion. Each dataset consists of M tasks for each learner to answer, and there are J learners in one dataset. We assume that each learner only takes any exam once. All the datasets we described in Section 3 require learners to write two texts. Hence, $M = 2$ in each dataset. $t_{m,j}$ denotes the m^{th} text written by learner l_j, which answers the m^{th} task $task_m$ in a dataset. text $t_{m,j}$ can be represented as a sequence of words written by learner l_j. The individual score for text $t_{m,j}$ marked by a human examiner is $s_{m,j}$.

$TL_j = \{t_{m,j}|m = 1, ..., M\}$ denotes the set of all the texts written by l_j in a dataset. In other words, TL_j is equivalent to the script answered by learner l_j.

$TN_{m,j} = TL_j \setminus t_{m,j}$ denotes the neighbouring text set of $t_{m,j}$, which is all the texts written by learner l_j except for $t_{m,j}$. In this section, since each dataset only contains 2 texts per learner, the number of texts in $TN_{m,j}$ is always 1, and the only text in this set is $t_{(M+1-m),j}$, which denotes the neighbouring text of $t_{m,j}$.

$TT_m = \{t_{m,j}|j = 1, ..., J\}$ denotes the sequence of all the texts to the m^{th} task $task_m$ answered by all learners in the same exam.

5 Assumptions

There are two assumptions behind authorship knowledge and ATS we want to validate.

The first assumption is that there is a variable $skill_j$ which can describe the writing skill of each learner l_j, and $skill_j$ is approximately constant during an exam. If we believe the skill of a learner could be measured by the English exam they take, $s_{m,j}$ for any m will be a sample from a distribution constrained by $skill_j$ during the exam. We also assume that no learner will behave totally differently on the two tasks during the same exam. In this case, these individual text scores should be close and correlate well with their skill $skill_j$, and this correlation might be helpful in training an individual-level model.

However, the first assumption is not always correct. In some circumstances, learners will perform really well on some tasks, but fail to finish other tasks to the same quality, and they can get low scores on these tasks. An obvious reason for this is that some learners may have managed their time badly and failed to finish the second task; some may also be better prepared for the topic and genre elicited by one of the prompts.

To verify and measure this assumption, we calculate root-mean-squared error (RMSE), quadratic weighted kappa (κ), Pearson (ρ_{prs}) and Spearman correlation (ρ_{spr}) between all the responses to the first task TT_1, and the second task TT_2 answered by all learners. The results are given in Table 3.

Dataset	RMSE	κ	ρ_{prs}	ρ_{spr}
unshuffled datasets				
FCE	2.264	0.700	0.706	0.704
B2-U	1.902	0.620	0.630	0.607
C1-U	2.148	0.680	0.684	0.670
AL-U	0.716	0.726	0.746	0.735
shuffled datasets				
B2-S	2.566	0.434	0.440	0.416
C1-S	2.984	0.408	0.419	0.394

Table 3: The relation between TT_1 and TT_2 to check how the scores of the first and second text written by each learner are correlated

As we can see, κ, ρ_{prs} and ρ_{spr} are above 0.6 in the four unshuffled datasets, and about 0.4 in the two shuffled datasets. It is suggested by Landis and Koch (1977) that there is a substantial agreement between two sequences if Cohen's Kappa is above 0.6 and a moderate agreement when Cohen's Kappa is between 0.4 to 0.6 [4]. We use their interpretation to describe our results, and there is at least a moderate correlation and agreement between the scores of TT_1 and TT_2. This verifies our first assumption to some degree. Whether this amount of agreement can affect the performance of an ATS model is further investigated in the following sections.

The second assumption concerns whether passing authorship knowledge to an ATS model truly improves the model performance by bringing more reliable features and better understanding about each learner's writing skill. An alternative explanation for the possible improvement, if it exists, is brought by the bias during the marking procedure. When comparing RMSE for the unshuffled and shuffled datasets as shown in Table 3, we can see that RMSE is higher for BS-2 than for B2-U, and higher for C2-S than for C2-U. This suggests that human judges might be biased when marking the second text after the first. Hence, we aim to determine whether authorship knowledge truly improves ATS performance by looking at the shuffled dataset, since any improvement on the unshuffled dataset might be the result of grading bias.

6 Methods

To study how authorship knowledge affects ATS, we first need a baseline model.

6.1 Baseline

In the baseline model, a feature vector $f_{m,j}$, extracted from text $t_{m,j}$ written by learner l_j, is used to train an individual-level model to learn the relationship between feature vector space F and text score space S. The model finally predicts the score of text $t_{m,j}$ as $\hat{s}_{m,j}$. The predicted score $\hat{s}_{m,j}$ might be invalid on the given grading scale. For example, an ATS model might predict a score of 4.3, but the grading scale requires an integer. Hence, we round $\hat{s}_{m,j}$ to the nearest valid score on the given grading scale as $\hat{rs}_{m,j}$, which is 4 in this case.

6.1.1 Features

The features for the baseline model we use are similar to those of Yannakoudakis et al. mentioned in Section 2. More specifically, we use features including word and POS n-grams, script

[4] Although Landis and Koch claimed that this interpretation is clearly arbitrary.

length, the n-gram missing rate estimated on a background corpus, phrase structure rules, and grammatical dependency distances between any two words within the same sentence, though we only use the top parse result for grammatical relation distance measures. The n-gram missing rate is estimated on UKWaC (Ferraresi et al., 2008). Besides that, we also include the number of words misspelt, the count of grammatical relation types, and the minimum, maximum and average sentence and word lengths. The POS tags, grammatical relations and phrase structure rules are derived from the RASP (robust accurate statistical parsing) toolkit (Briscoe et al., 2006). We remove any feature whose frequency in the training set is below 4, and keep the top 25,000 features that have the highest absolute Pearson correlation with text scores. Each feature vector is normalised to $||f_{m,j}|| = 1$.

6.2 Benchmark

Yannakoudakis et al. (2011) only built an overall-level model and evaluated it in terms of ρ_{prs} and ρ_{spr}. As we use more features and also a global feature selection step, we should ensure that our model is relatively optimal and thus a challenging baseline.

We firstly concatenate all the texts in script TL_j together as **concatenated text** ct_j so that

$$ct_j := t_{1,j} \oplus t_{2,j} \oplus ... \oplus t_{M,j}$$

We extract the script feature vector cf_j from the concatenated text ct_j based on the features defined in Section 6.1.1. We define the combined script score cs_j as the sum of the individual text scores to represent the overall score of each script: $cs_j := \sum_{m=1}^{M} s_{m,j}$

The FCE dataset has another overall script score ss_j for script TL_j used by Yannakoudakis et al. (2011). In order to benchmark with Yannakoudakis et al.'s work, we train an overall-level model by means of support vector regression (SVR) and SVM ranking between cf_j and its script score ss_j rather than cs_j together with a linear kernel. In order to get the valid predicted score on given the grading scale for SVM ranking, we train another linear regression model on the training set between the ranking scores and the actual text scores. For both SVR and SVM ranking, we then round the scores predicted from their corresponding regressors to the nearest valid integers on the given grading scale.

Model	RMSE	κ	ρ_{prs}	ρ_{spr}
UKWaC	X	X	0.735	0.758
CLC	X	X	0.741	0.773
DISCOURSE	X	X	0.749	**0.790**
SVR (BASE)	**3.988**	**0.657**	**0.761**	0.787
SVM RANKING	4.123	0.646	0.735	0.766

Table 4: The comparison of the previous work and our baseline models on the FCE test set.

We tune the regularization hyper-parameter on the development set and report the results achieving the lowest RMSE on the development set. The results are included in Table 4. The upper part of the table shows previous results. UKWaC and CLC are the results reported in Yannakoudakis et al. (2011) on SVM ranking models which use the UKWaC and the Cambridge Learner Corpus (CLC) (Nicholls, 2003) as the background corpus for n-gram missing rate estimation respectively. DISCOURSE is the CLC version with extra discourse features. In the DISCOURSE version, Yannakoudakis and Briscoe (2012) investigated different features to measure the coherence of a text and how these features affect the overall score of the texts in the FCE dataset. They showed that the coherence feature based on incremental semantic analysis (Baroni et al., 2007) measuring average adjacent sentence similarity can help their ATS system improve in terms of the Pearson and Spearman correlations.

Table 4 does not include any recent neural model on the FCE dataset, because the neural model developed by Farag et al. (2017) shows that there is still a performance gap between the neural model and the models built on hand-crafted features.

Our models achieve relatively good performance, and we also found that by selecting appropriate features and hyper-parameters, the difference between using ranking and regression to train an ATS model is relatively small. This contrasts with Yannakoudakis et al. (2011)'s finding that ranking is much better than regression on this task. Therefore, we use SVR (BASE) in the following experiments.

6.3 Model Fusion

There are two ways in which we can pass authorship knowledge in our ATS model. We refer to

them as **score fusion** and **feature fusion**.

For **score fusion**, we concatenate all the texts within the same script together as ct_j written by learner l_j. We extract the script feature vector cf_j from ct_j. An overall-level model is trained on cf_j and its combined script score cs_j, which is the sum of all the individual scores of one script. This overall-level model predicts the combined script score of ct_j as $\hat{cs}_j$, and the predicted normalised combined score $\frac{\hat{cs}_j}{M}$ is fused with the predicted individual score $\hat{s}_{m,j}$ by linear interpolation to get the predicted fused score $\hat{fs}_{m,j}$:

$$\hat{fs}_{m,j} := (1-\alpha)\hat{s}_{m,j} + \alpha\frac{\hat{cs}_j}{M}$$

The interpolation hyper-parameter α is tuned on the development set, and $\hat{fs}_{m,j}$ is then rounded to the nearest valid score on the given grading scale as the final predicted individual-level score for $t_{m,j}$.

For **feature fusion**, we still extract the script feature vector cf_j from ct_j. Then, we define the fused feature vector $ff_{m,j}$ of $t_{m,j}$ as the vector concatenated by $f_{m,j}$ and cf_j together as follows:

$$ff_{m,j} := (1-\beta)f_{m,j} \oplus \beta cf_j$$

where β is the concatenation weighting hyper-parameter to be tuned on the development set. We train an individual-level model on the fused feature vector $ff_{m,j}$ and text score $s_{m,j}$, and the predicted score $\hat{s}_{m,j}$ is rounded to the nearest valid score $\hat{rs}_{m,j}$ on the given grading scale.

Another question raised by the discussion here is what to fuse. For text $t_{m,j}$ in score fusion, instead of fusing the individual score $\hat{s}_{m,j}$ with the combined script score $\hat{cs}_j$, we can also fuse $\hat{s}_{m,j}$ with the individual predicted score $\hat{s}_{(M-m+1),j}$ from the other text within the same script, which is the neighbouring text $t_{(M-m+1),j}$.

For feature fusion, when we are augmenting text feature vector $f_{m,j}$ to $ff_{m,j}$, we can concatenate it with the feature vector $f_{(M-m+1),j}$ from the neighbouring text $t_{(M-m+1),j}$ instead of the script feature vector cf_j derived from the concatenated text ct_j. Therefore, we have two different fusion approaches, and each approach also has two different sources to fuse.

It should be noticed that the two questions for each dataset are designed on a similar difficulty level. The fusion approach can easily be made to work even if these questions are not on the same difficulty level. If the difficulty difference between the targeted question and the neighbouring question is too large, we can penalise the neighbouring question so that the ATS model mainly look at the targeted question. This is straightforward to do in our method by adjusting the weight of the neighbouring question. We will investigate questions from different difficulty levels in future work once we have a suitable dataset.

7 Results and Discussion

In this section, we evaluate the baseline model and the fusion approaches to study the influence of authorship knowledge. For each setup, we train an individual-level model for each dataset. The model for each setup is optimised on each development set in terms of RMSE. We tune the SVR regularisation and interpolation hyper-parameters on each development set. We report RMSE, κ, ρ_{prs} and ρ_{spr} in Table 6 for each test set. The optimal interpolation hyper-parameters for each fusion approach are reported as α/β in Table 6.

Some readers might notice that there is a numerical difference between Table 4 and Table 6 for the same baseline model evaluated on the FCE test set. The reason for the difference here is that these two tables correspond to two different tasks. The task in Table 4 is predicting the overall-level score, and Table 6 is the individual-level score of a text. It seems that predicting the individual-level scores is a harder task as there is less text to assess (Section 1.1).

For feature fusion, feature fusion with neighbouring text (FF-NT) and concatenated text (FF-CT) is consistently better than the baseline (BASE) on all the datasets except for the B2-U on κ, ρ_{prs} and ρ_{spr}. For score fusion, score fusion with concatenated text (SF-CT) is better than BASE on all the six datasets except for κ in AL-U. In contrast, score fusion with neighbouring text (SF-NT) is better than BASE on all the datasets regarding RMSE except for FCE, but κ is only better than BASE on C1-S. Both SF-CT and SF-NT are better than BASE in terms of ρ_{prs} and ρ_{spr}. The improvement is also visible on the two shuffled datasets, and we suggest that adding authorship knowledge is not merely the result of modelling grading bias, which answers the second assumption in Section 5.

To give a better global understanding of how each approach performs, we conduct the Wilcoxon

signed-rank test (Wilcoxon, 1945; Demšar, 2006) across the six datasets to see whether any setup is significantly better or worse than BASE at a global level. We use the SciPy implementation to run the test[5], and the p-values of all the metrics across all the six datasets are listed in Table 5. Based on the result in Table 5, there is a significant difference between all the fusion approaches ($p < 0.05$) and BASE on all the metrics except for SF-NT on κ across multiple datasets.

Setup	RMSE	κ	ρ_{prs}	ρ_{spr}
SF-NT	0.046	**0.058**	0.028	0.028
SF-CT	0.028	0.046	0.028	0.028
FF-NT	0.028	0.046	0.046	0.046
FF-CT	0.028	0.046	0.046	0.046

Table 5: p-value for each approach estimated by the Wilcoxon signed-rank test across all the six datasets. The value bigger than 0.05 is in **bold**

7.1 Hyper-parameters

$\alpha, \beta > 0.5$ in each fusion approach tells the ATS model that it should favour the information from the other source over the current individual text $t_{m,j}$ being marked, and vice versa. We also visualise the relation between β and RMSE for the feature fusion approaches in Figure 1 and 2.

For the fusion with concatenated text ct_j, $\alpha > 0.5$ on FCE and C1-S in SF-CT. $\beta > 0.5$ for all the datasets except for B2-U in FF-CT. Furthermore, if we tune β on the test sets, we can find the optimal β for all the six datasets are bigger than 0.5. On the one hand, we are a little bit surprised that the fusion approaches with concatenated text favour ct_j, and it might mean that ct_j is more salient compared to the original text $t_{m,j}$ in ATS. On the other hand, it is still to be expected to observe these results, because ct_j also contains $t_{m,j}$, and the information from $t_{m,j}$ is still dominant in the model even if $\alpha, \beta > 0.5$.

In contrast, we expect that the model fused with neighbouring text achieves the best performance on each dataset when α or β is smaller than 0.5, as the model should focus on the text $t_{m,j}$ being marked. For SF-NT in Table 6, the optimal α is always smaller than 0.5. However, for FF-NT, the optimal $\beta = 0.5$ for AL-U and C1-U in Table 6. Furthermore, if we choose the test sets to tune β instead of the development sets, we can see that $\beta > 0.5$ on the FCE, C1-U and AL-U dataset in Figure 2. Based on these results, we suggest that in some cases, the features from two tasks written by the same learner could be highly similar and shared to some extent in an ATS model.

[5]https://www.scipy.org

7.2 Score Difference

Although positive effects are observed in most cases, no method is significantly better than BASE on every dataset and metric we used. One reason might be that it is not ideal to aggregate the two texts written by the same learner together if the performance difference between these texts is big. For example, some learners might perform well on the first task, but fail to complete the second task. This is what we have discussed in the first assumption in Section 5, and this assumption might be invalid in some cases. So, we conduct another study to see how the score difference between the two texts in each script affects the model performance.

We define the script score difference sd_j as the score difference between two texts $t_{1,j}$ and $t_{2,j}$ within the same script TL_j: $sd_j := |s_{1,j} - s_{2,j}|$.

The text score difference of text $t_{m,j}$ is defined as the score difference of the script to which it belongs: $sd_{m,j} := sd_j$.

The text score error $error_{m,j}$ denotes the difference between the predicted score and the gold score of $t_{m,j}$: $error_{m,j} := |\hat{rs}_{m,j} - s_{m,j}|$.

The text score errors $error_{m,j}$ produced by BASE and any fusion approach on text $t_{m,j}$ denote $error_{m,j}^{\text{BASE}}$ and $error_{m,j}^{\text{FUSION}}$, respectively.

The performance difference $PD_{m,j}$ between BASE and any fusion approach for text $t_{m,j}$ denotes the difference between the errors made by the two setups:

$$PD_{m,j} := error_{m,j}^{\text{BASE}} - error_{m,j}^{\text{FUSION}} \quad (1)$$

$PD_{m,j} > 0$ means that the fusion approach is better than BASE at predicting the score of $t_{m,j}$, and vice versa.

We calculate the Pearson correlation ρ_{prs} between $PD_{m,j}$ and $sd_{m,j}$ for each test set in Table 7. Although we do not find any interesting relation between the correlation here and the performance variation in Table 6, Table 7 does reveal some patterns. On the one hand, most values are negative, and the five positive values in bold tend to be close to 0, and p is always bigger than 0.05 for all the positive values. We suggest that there is a negative correlation between performance dif-

Setup	RMSE	κ	ρ_{prs}	ρ_{spr}	α/β	RMSE	κ	ρ_{prs}	ρ_{spr}	α/β
	FCE					AL-U				
BASE	2.569	0.511	0.662	0.652	X	0.693	0.620	0.684	0.659	X
SF-NT	2.572	0.490	0.693	0.696+	0.35	0.686	0.603	0.704	0.687+	0.34
SF-CT	2.495	0.533	**0.696**	**0.702+**	0.70	0.691	0.610	0.689	0.667	0.33
FF-NT	2.529	**0.554+**	0.688	0.688+	0.30	0.683	0.634	0.698	0.680	0.50
FF-CT	**2.460+**	**0.554+**	0.694	0.695	0.67	**0.664+**	**0.649+**	**0.720+**	**0.710+**	0.70
	B2-U					B2-S				
BASE	1.991	0.246	0.359	0.339	X	2.085	0.386	0.476	0.442	X
SF-NT	1.979	0.241	0.371	0.347	0.18	2.050+	0.384	0.501+	0.463	0.23
SF-CT	**1.954+**	**0.271+**	**0.398+**	**0.377+**	0.32	2.029+	0.400	0.510+	0.476+	0.33
FF-NT	1.982	0.242	0.348	0.324	0.20	**1.983+**	**0.430+**	**0.541+**	**0.511+**	0.33
FF-CT	1.974	0.241	0.354	0.333	0.25	2.017+	0.415	0.506	0.481	0.80
	C1-U					C1-S				
BASE	2.405	0.269	0.411	0.410	X	2.421	0.341	0.504	0.471	X
SF-NT	2.387	0.260	0.438	0.433	0.37	2.413	0.343	0.511	0.480	0.02
SF-CT	2.366+	0.288	0.453+	0.451+	0.37	**2.346+**	0.378+	**0.567+**	**0.523+**	0.78
FF-NT	**2.350+**	**0.304+**	**0.462+**	**0.455+**	0.50	2.370+	**0.389+**	0.529	0.498	0.40
FF-CT	2.378	0.296+	0.428	0.420	0.60	2.361+	0.381+	0.548+	0.513+	0.67

Table 6: The results of different setups on the test sets. The best setup per dataset is in **bold**. GREEN means improvement and RED means degradation over BASE. The optimal interpolation hyper-parameters for each fusion approach are reported as α/β. + means significantly better ($p < 0.05$) than BASE using the permutation randomisation test (Yeh, 2000) with 2,000 samples. No metric is found significantly worse than BASE.

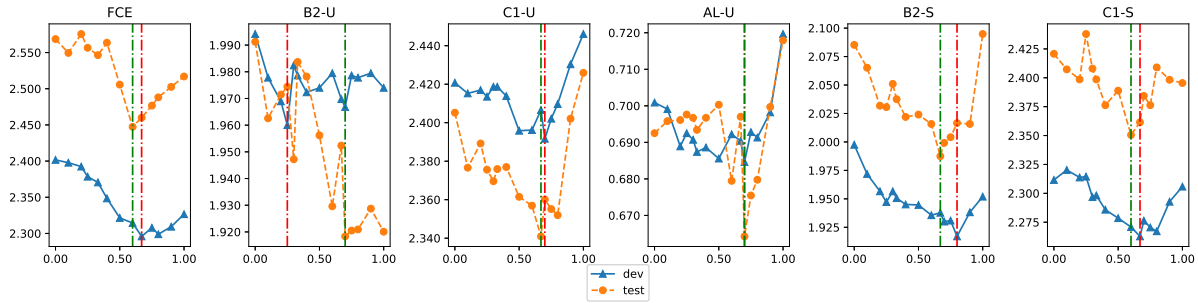

Figure 1: How RMSE (y-axis) changes with β (x-axis) in FF-CT. The vertical RED and GREEN dashed-dot lines in each graph represent that the model achieves the lowest RMSE on the development and test sets at the corresponding β.

ference $PD_{m,j}$ and text score difference $sd_{m,j}$ on some datasets.

On the other hand, only the p-values for six negative values in Table 7 are smaller than 0.05. We think the negative influence brought by the score difference is not huge, because the scores of the two texts written by the same learners are at least moderately correlated in Table 3. This correlation might reduce the negative influence of score difference here.

In some operational settings, it might be considered unfair to use other responses to score a new response, and grading guidelines usually require texts to be marked independently. Nevertheless, we found a clear improvement when making use of such information, and no approach is significantly worse than BASE on any metric or dataset. In other words, the positive influence brought by our fusion approaches is stronger than any possible negative effects.

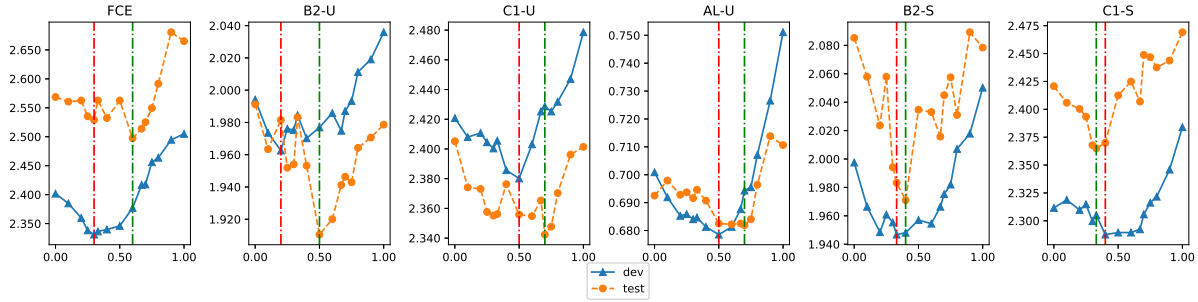

Figure 2: How RMSE (y-axis) changes with β (x-axis) in FF-NT. The vertical RED and GREEN dashed-dot lines in each graph represent that the model achieves the lowest RMSE on the development and test sets at the corresponding β.

Setup	SF-NT	SF-CT	FF-NT	FF-CT
FCE	-0.102	-0.156*	**0.002**	-0.162*
B2-U	-0.034	-0.009	**0.036**	**0.012**
C1-U	-0.060	-0.018	**0.034**	-0.005
AL-U	-0.188*	-0.107*	-0.021	-0.108*
B2-S	-0.039	-0.039	**0.014**	-0.074
C1-S	-0.074	-0.102*	-0.032	-0.048

Table 7: The Pearson correlation between performance difference $PD_{m,j}$ and script score difference $sd_{m,j}$ on the test sets. * denotes p-value < 0.05, and **bold** denotes a positive correlation.

8 Conclusion

In this paper, we studied how authorship knowledge, by means of score fusion and feature fusion, is a useful feature in ATS. We showed that including such information improves model performance at in most datasets, and that improvement is not only from modelling grading bias. One possible topic for future work is to study whether the target CEFR level of each dataset affects the influence of adding authorship knowledge.

9 Acknowledgement

This work is funded by the Institute for Automated Language Teaching and Assessment (ALTA). Special thanks to Christopher Bryant, Yimai Fang, Helen Yannakoudakis, Nanyang Ye and Ann Copestake, as well as the anonymous reviewers for their valuable suggestions at various stages.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 715–725. http://www.aclweb.org/anthology/P16-1068.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment* 4(3).

Marco Baroni, Alessandro Lenci, and Luca Onnis. 2007. Isa meets lara: An incremental word space model for cognitively plausible simulations of semantic learning. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*. Association for Computational Linguistics, pages 49–56.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. Association for Computational Linguistics, Sydney, Australia, pages 77–80. https://doi.org/10.3115/1225403.1225423.

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, pages 1741–1752. http://www.aclweb.org/anthology/D13-1180.

Ronan Cummins, Helen Yannakoudakis, and Ted Briscoe. 2016a. Unsupervised modeling of topical relevance in l2 learner text. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. pages 95–104.

Ronan Cummins, Meng Zhang, and Ted Briscoe. 2016b. Constrained multi-task learning for automated essay scoring. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 789–799. http://www.aclweb.org/anthology/P16-1075.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research* 7(Jan):1–30.

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*. Association for Computational Linguistics, Austin, Texas, pages 1072–1077. https://aclweb.org/anthology/D16-1115.

Youmna Farag, Marek Rei, and Ted Briscoe. 2017. An error-oriented approach to word embedding pre-training. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Copenhagen, Denmark, pages 149–158. http://www.aclweb.org/anthology/W17-5016.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*. pages 47–54.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *HLT-NAACL*. pages 185–192.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* pages 159–174.

Leah S Larkey. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 90–95.

Diane Nicholls. 2003. The cambridge learner corpus: Error coding and analysis for lexicography and elt. In *Proceedings of the Corpus Linguistics 2003 conference*. volume 16, pages 572–581.

Ellis Batten Page. 2003. Project essay grade: Peg. *Automated essay scoring: A cross-disciplinary perspective* pages 43–54.

Peter Phandi, Kian Ming A. Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 431–439. http://aclweb.org/anthology/D15-1049.

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

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 950–961. http://www.aclweb.org/anthology/C14-1090.

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*. Association for Computational Linguistics, Austin, Texas, pages 1882–1891. https://aclweb.org/anthology/D16-1193.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics bulletin* 1(6):80–83.

Helen Yannakoudakis and Ted Briscoe. 2012. Modeling coherence in esol learner texts. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, Montréal, Canada, pages 33–43. http://www.aclweb.org/anthology/W12-2004.

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*. Association for Computational Linguistics, Portland, Oregon, USA, pages 180–189. http://www.aclweb.org/anthology/P11-1019.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, pages 947–953.