Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5603–5613
Marseille, 11–16 May 2020

Reproduction and Replication: A Case Study with Automatic Essay Scoring

Eva Huber, Çağrı Çöltekin
University of Tübingen
eva.huber@student.uni-tuebingen.de, ccoltekin@sfs.uni-tuebingen.de

Abstract

As in many experimental sciences, reproducibility of experiments has gained ever more attention in the NLP community. This paper presents our reproduction efforts of an earlier study of automatic essay scoring (AES) for determining the proficiency of second language learners in a multilingual setting. We present three sets of experiments with different objectives. First, as prescribed by the LREC 2020 REPROLANG shared task, we rerun the original AES system using the code published by the original authors on the same dataset. Second, we repeat the same experiments on the same data with a different implementation. And third, we test the original system on a different dataset and a different language. Most of our findings are in line with the findings of the original paper. Nevertheless, there are some discrepancies between our results and the results presented in the original paper. We report and discuss these differences in detail. We further go into some points related to confirmation of research findings through reproduction, including *the choice of the dataset*, *reporting and accounting for variability*, *use of appropriate evaluation metrics*, and *making code and data available*. We also discuss the varying uses and differences between the terms *reproduction* and *replication*, and we argue that *reproduction*, the confirmation of conclusions through independent experiments in varied settings is more valuable than exact *replication* of the published values.

1. Introduction

Confirmation of results through independent replication is a well-established practice in experimental sciences. However, a number of recent negative reproduction results in medical and behavioural sciences indicate that a significant number of published results are not verifiable through independent reproduction efforts (Open Science Collaboration, 2015; Freedman et al., 2015, for example). The issue, often titled *reproducibility crisis*, has become influential both in scientific communities and popular media. Similar concerns have been raised in the fields of machine learning and artificial intelligence (Kitzes et al., 2017; Hutson, 2018; Raff, 2019). As a field that heavily relies on experimental work, the same concerns apply to computational linguistics and natural language processing (NLP), and there have been recent efforts to understand the extent of the problem and identify potential solutions. The present work is conducted in the context of such an effort, REPROLANG 2020 shared task on reproducibility of results in computational studies of language.[1]

Our study, in particular, is concerned with the reproduction of a study of *automatic assay scoring* (AES) for determining language proficiency levels of second language learners (Vajjala and Rama, 2018). In AES, the aim is to assign a score, mark or level to a text by means of an automatic system. The motivation behind the development of such systems lies in the time, cost and reliability that are involved in manual essay correction marking (Dikli, 2006). AES is one of the NLP applications that could find its way into real life applications, and hence, have a high potential impact on the society. For example, some of these systems are currently being used in high stake examinations.[2] Since the quality of such high-impact applications is an important concern, reproducibility is especially desirable and needed. Another motivation for the choice of the present task has to do with the fact that AES is a text classification (or regression) task, which is a well-studied and straightforward NLP task. This allows us to focus less on the specifics of the systems and the original task, and more on issues regarding reproducibility.

The target paper we attempt to reproduce by Vajjala and Rama (2018) presents a series of models for predicting the Common European Framework of Reference (CEFR) levels of essays written by German, Czech and Italian learners. The authors use the MERLIN corpus (Boyd et al., 2014) for the task, and present multilingual and cross-lingual models along with monolingual models.

We present three sets of experiments in this paper. First, we use their publicly-available code to replicate their results. The second set of experiments are performed using the same dataset, the same machine learning models and the same features, but we use our own implementation, and perform additional tuning of the model hyperparameters. And third, we test whether their features can be used to successfully predict essay scores of a fourth language, namely English. We use the Cambridge Learner Corpus (CLC) which includes texts that were written as part of a First Cambridge exam (Yannakoudakis et al., 2011). The first two sets of experiments verify the reproducibility of the original results, while the additional dataset allows for testing whether the model is general enough to be extended to another language with a different label granularity.

This paper is structured as follows. In Section 2., we give some relevant pointers to existing literature on reproducibility in NLP. Additionally, a brief overview of previous work on AES, and a brief summary of the target paper is given. Section 3. includes our reproduction experiments of Vajjala and Rama (2018) where we discuss our results in comparison to the original ones. Section 4. presents the results from experiments where we use the same data but a different implementation. The results of experiments where we use their system on a different language are presented in Section 5.. We summarise and discuss the findings of this paper in Section 6., with a brief conclusion in Section 7..

[1] https://www.clarin.eu/event/2020/reprolang-2020.

[2] For instance, Pearson use their in-house *Intelligent Essay Assessor* to grade the written part of the PTE academic test (https://pearsonpte.com/wp-content/uploads/2015/05/7.-PTEA_Automated_Scoring.pdf).

2. Background

2.1. Rep(roduc|licat)tion

Along with the *reproducibility crisis* in the wider scientific context (Fidler and Wilcox, 2018), the issue of reproduction in NLP and related fields has also attracted recent attention. Besides the present shared task, there has been a number of recent workshops and campaigns in the field,[3] as well as in the closely related field of machine learning.[4]

Despite an increasing interest in studies that aim to verify earlier results, it is often unclear what is exactly being verified, and the terms *reproduction* and *replication* are used for referring to different activities in different studies (Cohen et al., 2018). Some use *reproduction* and *replication* interchangeably, whereas others distinguish between the two or use only one of them. The issue may become even more confusing as the terms are sometimes used with opposite meanings in different studies.

So far, we have also used these two terms without a clear definition. Before providing a brief overview of relevant work in NLP, we first clarify our use of the terms. In the rest of this paper, we adopt the usage of the terms as defined by Drummond (2009). We use the term *replication* to refer to the activity of running the same code on the same dataset with the aim of producing the same (or sufficiently similar) measurements presented in the original paper. We use the term *reproduction* to refer to the activity of verifying the claims with experimental settings that are different from the ones in the original paper. For NLP experiments, this typically means re-implementation of the method(s) and the use of different datasets and/or languages. However, for example, Branco et al. (2017) uses the terms in the opposite way. Cohen et al. (2018), on the other hand, defines *replication* (or *repetition*) as running the experiment as implemented by the original study without reference to the aims of the repetition, while calling *reproducibility* the activity of verifying an outcome of the experiment. In their definition, reproduction is associated with one of three levels: a *value*, a *finding* or a *conclusion*. The present REPROLANG shared task is also rather vague about its aims in this respect. Given a clear recipe to produce the values in the selected figures and tables, we assume that the aim is closer to *replication* according to our definition. However, varying the experimental conditions (e.g., using different languages, corpora) is also encouraged in the task description.

Benefits of verifying the conclusions of an experiment with independent reproduction experiments is hardly disputable. However, the same is not true for replication of experiments. The position in earlier work often ranges from a strong emphasis on the value of replication (Pedersen, 2008; Wieling et al., 2018) to strong arguments against any utility of it (Drummond, 2009). Our position on the subject is similar to Drummond (2009). We believe a scientific result gains more support when its overall conclusions are verified with a different experimental setting. However, there are some cases where replication experiments are useful, especially when it means that one shares the code and data used in the experiments.

Although it is not always clear what is exactly being reproduced (or replicated), there is clearly an increasing interest in the subject in the NLP community. Most early reproduction or replication studies in NLP is concerned with subfields whose applications may have high potential impact, such as biomedical NLP (Névéol et al., 2016; Cohen et al., 2018). A number of studies include surveys of papers in prominent NLP conferences for quantifying the properties related to reproduction and replication, such as release of the data and code, actual availability of the resources used, authors' willingness to share their code if it is not already available, or whether the reported results are statistically sound (Mieskes, 2017; Dror et al., 2017; Wieling et al., 2018).

There has also been a number of interesting papers with case studies. Fokkens et al. (2013) focus on (a lack of) system descriptions (e.g., preprocessing, versions of the resources used) that cause reproduction attempts to fail. They further note that replication attempts are also useful for gaining further insight into the original problem and the solution. Cohen et al. (2018) also includes three case studies, which are analyzed carefully according to dimensions summarized above. Moore and Rayson (2018) perform reproduction experiments of over 10 sentiment analysis methods. Their focus is on the reproduction of results on multiple datasets, emphasizing the use of as many datasets as possible in evaluation of NLP systems, which is in line with the other studies where careful, less biased data is shown to change the conclusions of earlier reports (Pirina and Çöltekin, 2018). 10 case studies reported by Wieling et al. (2018) involve only replication. The authors tested whether the code released by earlier studies could be run in a limited 'human time' and, if so, whether the values output by the software are the same as the ones reported in the original papers.

The replication or reproduction attempts listed above report mostly negative results, mirroring the results reported in other fields (Open Science Collaboration, 2015, for example). As a result, it is clear that there is need for mechanisms or guidelines to increase the confirmability of the studies in the field, as well as more reproduction studies and further discussion on fruitful ways to perform replication and reproduction studies.

2.2. Automated Essay Scoring

Automated essay scoring, also called automatic text scoring, goes back to the 60s when Ellis Page developed Project Essay Grader (Page, 1967), A number of AES schemes have emerged since then, some of the most prominent in the field being e-rater (Attali and Burstein, February 2006) and Intelligent Essay Assessor (Foltz et al., 1999) based on Latent Semantic Analysis (Deerwester et al., 1990).

A wide range of features have been developed to analyse essays, from as simple as document length to more complex ones involving, for instance, discourse cohesion (see Zesch et al. (2015) for an overview of features in the litera-

[3] The Workshop on Replicability and Reproducibility in Natural Language Processing: adaptive methods, resources and software at IJCAI 2015 (https://sites.google.com/site/adaptivenlp2015/), 4REAL Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language (http://4real.di.fc.ul.pt/), and CLEF lab on reproducibility (http://www.centre-eval.org/).

[4] For example, reproducibility challenges at NeurIPS (https://reproducibility-challenge.github.io/neurips2019/) and ICLR (https://reproducibility-challenge.github.io/iclr_2019/).

ture). The emergence of Neural Networks and Deep Learning has also prompted an appearance of a body of work that uses deep learning to automatically score essays (Alikaniotis et al., 2019; Taghipour and Tou Ng, 2016; Nadeem et al., 2019). For instance, Alikaniotis et al. (2019) use *score-specific* word embeddings for which they use pretrained embeddings and further train them on predicting essay scores.

A common question raised in the literature is whether to treat AES as a regression or a classification task. Berggren et al. (2019) tackle this issue by experimenting with both regression and classification and also non-neural and neural models for Norwegian Essay Scoring.

Both corpora used in this paper have previously been applied for AES. Weiß (2017) demonstrated the power of complexity features to predict CEFR levels of German essays using the MERLIN corpus. Yannakoudakis et al. (2011), in the paper introducing the CLC corpus, also present AES experiments in which they consider the task as a rank preference learning problem. They use features such as phrase structure rules and error rate to predict essay scores.

2.3. Target study: CEFR scoring

Our target paper by Vajjala and Rama (2018) reports three sets of AES experiments on the MERLIN corpus, which consists of essays written by learners of three different languages, namely Czech, German and Italian. The first set of results includes a comparison of feature sets on all three languages, where the models are trained and tested on the monolingual data. In the second set of experiments, the authors train a single model using data from all languages. The final results are from cross-lingual experiments, where the authors train a model on German data, and test the model on the other languages, investigating cross-lingual transfer on the AES task. The authors present F1-scores (weighted by the support of each class) for each setting. In this section we briefly describe the dataset, and the models used in the original study.

Data and preprocessing Vajjala and Rama (2018) use the MERLIN corpus (Boyd et al., 2014) for their experiments which contains 2286 essays by L2 speakers of German, Italian and Czech. These essays were written as part of written examinations and manually marked with the correspondent CEFR level. CEFR categorises language proficiency into three groups: *basic user*, *independent user* and *proficient user*. These three levels - A, B and C - are again divided into two subclasses, resulting in a total of six levels (A1 < A2 < B1 < B2 < C1 < C2). The essays are annotated with levels on different linguistic dimensions, such as *sociolinguistic appropriateness* and *vocabulary control*. However, in their experiments, Vajjala and Rama (2018) only predict the overall score. They remove any essay that is rated with a level occurring less than 10 times in the subcorpus of one of the languages, and they further remove any unmarked essays. The final distribution of the essays can be seen in Table 1.

CEFR level	DE	IT	CZ
A1	57	29	0
A2	306	381	188
B1	331	393	165
B2	293	0	81
C1	42	0	0

Table 1: This table shows the distribution of essays in the MERLIN corpus. An identical table can be found in Vajjala and Rama (2018, Table 1).

Features and classifiers Vajjala and Rama (2018) use features that are common across AES systems, and most of them often appear in other text classification tasks:

1. word n-grams and POS n-grams where n is 1 to 5;
2. dependency n-grams where n is 1 to 3;
3. domain features which consist of features that are specific to AES research. These include document length, lexical richness features and error features (see their paper for a more detailed description);
4. combined features where domain features are concatenated with each of the three n-gram features.

The texts were POS-tagged and automatically annotated with dependency relations using UDPipe (Straka et al., 2016). The annotated files are readily available in their git repository. They combine the dense domain features and the sparse n-gram features by first training a classifier on the sparse features to get the probability distribution of CEFR classes for each essay. As a second step, they use those probability distributions as features together with the dense features to train the final classifier.

The authors train and test three 'traditional' classifiers, *random forests*, linear *support vector machines* (SVMs) and *logistic regression*, as well as a *multi-layer perceptron* (MLP) with word embeddings trained on the task. The traditional classifiers are trained in different combinations of the features introduced above. The results are compared to a trivial baseline with document length as the only feature. The authors present F1-scores for each setting, where the scores are weighted by the support (the number of positive samples in the gold standard data). In all settings, the models are evaluated by doing 10-fold cross validation. Vajjala and Rama (2018) use sci-kit learn (Pedregosa et al., 2011) to implement the traditional classifiers, and Keras (Chollet, 2015) with Tensor Flow as the back-end to implement the neural networks.

What to reproduce or replicate The task formulated by the REPROLANG organizers is to replicate the values from three tables presenting the results of three sets of experiments including monolingual, cross-lingual and multilingual models. We present the results concerning the replication experiments in Section 3..

The more interesting undertaking, however, is to further explore the results and conclusions of our replication and the original paper by reproducing the experiments. The straightforward contribution of the original paper is demonstrating the success of the method on three different languages. However, Vajjala and Rama (2018) do not state any clear, explicit conclusions. Although it is not easy

to draw general conclusions because the expected level of success differs from application to application, the readers are likely to form a general opinion of the success of the method based on the scores presented in the paper. An interesting aspect of the study is the inclusion of multilingual and cross-lingual experiments, which may show whether cross-lingual transfer helps for this task. The results presented, however, do not support any clear conclusions on this matter, as we discuss it further in Section 3.. One last potential message a reader may get from the paper is related to the comparison between the classification methods. Since the authors performed experiments with multiple classification methods, one may be inclined to draw conclusions about the best classification method for the task.

3. Replication: Same Data, Same Code

The aim of the experiments presented in this section is to replicate the values presented in the original paper for all three settings: monolingual, cross-lingual and multilingual. In our replication attempts, we used the code published by the authors[5] with minor modifications.[6] Our first modification is to let the random seed vary as opposed to the original code where the random seed was fixed. Although fixing the random seed may help replicating the exact same values, it hides an important aspect of most machine learning systems: the variation due to random initialization and different training–test splits. As a result, we repeat each of their experiments 10 times, and allow the random seed to differ in each run.

Our second modification pertains to the reported evaluation metric. The original paper reports F1 scores weighted by support, which promotes models that make fewer mistakes on majority class(es), while errors made on minority classes are not heavily penalized. For example, confusing the level C1 with another level is considered less severe than confusing the level B1 with another level, merely because B1 contains more data points than C1. Since we are not aware of a reason for such a preference, we believe that reporting macro-averaged F1 scores is more appropriate. For the required replication scores, we report both weighted and macro-averaged F1 scores. For the additional reproduction experiments, we report only macro-averaged F1 scores.

3.1. Monolingual Classification

The first set of results presented in Vajjala and Rama (2018) scores on monolingual models. Table 2 presents scores we obtained in our replication attempt alongside the difference from the results reported in the original paper. Note that we do not attempt to replicate the single-value results in the original paper. The values we report are averages of multiple experiments with varying cross-validation splits and random initializations. Most results in Table 2 are within a reasonable range of the results reported in the original article. However, some large discrepancies occur in some of the experiments. The large differences in baseline scores and domain-only scores are due to a bug in the original software where 'macro-averaged' F1 scores were calculated, while the paper reports them as 'weighted' F1 scores. The discrepancy regarding embedding scores is less clear. A plausible explanation is differences in the libraries. Particularly the libraries used for training neural networks are in active development, and there may be important changes in short time periods. However, our attempts to unify the software and libraries that may have obvious effects did not reduce the discrepancy.

Besides some large discrepancies, other observations from the replication results in Table 2 include the fact that macro-averaged F1 scores are lower in all settings. Weighted averaging inflates the scores, potentially giving an impression of a higher success rate.

The variability of scores is, in general, low, rarely exceeding $1\,\%$ difference in F1 scores. Yet, this variation already invalidates some conclusions one may get from the results reported in the original article. For example, although the paper is careful about not making strong claims, the authors mark combination of word n-gram and domain features as the best overall solution for all three languages. According to the weighted F1 scores reported in Table 2, this is no more true for Italian and Czech. Our findings for German are almost in line with the original result. The combined features of domain and word n-grams yield the best results, but they are within a standard deviation of the scores of the other two combined features. Hence, this may indicate that some of the differences here are by chance.

3.2. Multilingual Classification

The second replication target in the REPROLANG shared task is the results from the multilingual model reported in Vajjala and Rama (2018, Table 3). This experiment compares the feature sets used in the monolingual experiments in a single big model trained on the data from all languages. The authors present two settings, where the difference is whether the model is informed about the language of the training and test instances. For the traditional classifiers, the language id is given to the system as another symbolic feature. The neural model, on the other hand, predicts the language as an additional objective.

Similar to the monolingual replication experiments, we use the code published by the authors with minor modifications to report macro-averaged F1 scores alongside the weighted F1 scores. Table 3 reports our replication results similar to the way our monolingual replication results were presented.

The replication of the multilingual model also comes with few surprises. The discrepancies occurring in the differences in baseline and domain features are again due to the fact that in the original paper, the macro-averaged F1 score was reported instead of the weighted F1 score. Our present guess for the cause of the other large differences is again based on the software/library configuration.

A general difference here is that none of the multilingual models in our replication study outperforms the best performing monolingual models, while their multilingual results are better than monolingual scores for German. The utility of this model, especially the fact that it was tested on the complete data, is not entirely clear to us. Perhaps, a

[5] https://github.com/nishkalavallabhi/UniversalCEFRScoring; commit: 86d60de.

[6] The software that we used to get the replication results are available as a docker container at https://gitlab.com/coltekin/cefr-reproduction; commit hash: 94d5a7700e2c39aab8cb1fc7f6a8182a0c076c2c; commit tag: v1.0.

	DE					IT					CZ				
Features	$F1_w$	$\sigma F1_w$	$\Delta F1_w$	$F1_m$	$\sigma F1_m$	$F1_w$	$\sigma F1_w$	$\Delta F1_w$	$F1_m$	$\sigma F1_m$	$F1_w$	$\sigma F1_w$	$\Delta F1_w$	$F1_m$	$\sigma F1_m$
base	61.6	0.00	13.7	48.9	0.00	80.0	0.00	22.2	57.3	0.00	59.6	0.00	0.9	55.3	0.00
word	59.8	0.74	−6.8	46.1	1.05	80.8	0.31	−1.9	59.8	1.42	71.6	0.96	−0.5	68.4	1.10
pos	65.2	0.35	−1.1	50.3	0.32	80.5	0.28	−2.0	59.5	0.19	69.2	0.68	−0.7	65.0	0.85
dep	63.6	0.68	−2.7	49.1	0.68	79.7	0.37	−1.6	59.1	0.72	71.5	0.75	1.1	68.5	0.83
dom	62.7	0.29	9.4	48.9	0.62	81.1	0.00	15.8	65.5	0.00	67.0	0.52	0.7	63.6	0.68
word+dom	63.4	0.53	−5.2	52.7	1.06	79.7	0.44	−4.0	58.1	1.53	72.4	0.89	−1.0	69.3	1.02
pos+dom	64.8	0.75	−3.8	53.5	1.23	79.2	0.42	−2.4	55.0	0.66	70.5	0.97	−0.4	67.6	1.12
dep+dom	63.6	0.49	−4.6	52.4	1.10	78.8	0.50	−1.8	54.9	0.99	71.5	1.55	0.3	69.0	1.73
emb	46.7	0.72	−17.9	37.8	0.46	64.7	0.91	−14.7	53.3	1.21	48.2	0.82	−14.3	42.2	0.94

Table 2: Replication results for monolingual experiments Vajjala and Rama (2018, Table 2). Besides weighted F1 score ($F1_w$), we report macro-averaged F1 scores ($F1_m$) in all settings. The values reported are averages of scores of 10 runs. The columns with prefix σ indicate the standard deviation of the scores obtained. ΔF_w report the difference between the mean score presented in this table and the value reported in the original paper. All numbers are percentages.

	Without language information					With language information				
Features	$F1_w$	$\sigma F1_w$	$\Delta F1_w$	$F1_m$	$\sigma F1_m$	$F1_w$	$\sigma F1_w$	$\Delta F1_w$	$F1_m$	$\sigma F1_m$
base	49.3	1.69	6.5	42.7	0.00	-	-	-	-	-
word	60.3	0.28	−11.8	42.9	0.24	60.4	0.16	−11.5	42.9	0.29
pos	68.1	0.51	−4.5	48.2	0.33	68.0	0.23	−4.4	48.3	0.27
dep	66.0	0.46	−4.3	47.2	0.42	66.1	0.34	−3.2	47.2	0.44
dom	60.0	0.15	15.1	39.0	0.43	64.7	0.31	17.6	43.7	0.53
emb	65.8	0.75	−3.5	46.2	0.66	66.2	0.77	−2.7	46.7	0.38

Table 3: Replication results for multilingual model of Vajjala and Rama (2018, Table 3). Besides weighted F1 score ($F1_w$), we report macro-averaged F1 scores ($F1_m$) in all settings. The values reported are averages of scores of 10 runs. The columns with prefix σ indicate the standard deviation of the scores obtained. ΔF_w report the difference between the mean score presented in this table and the value reported in the original paper. All numbers are percentages.

different test setup, for instance, testing the model on individual languages, may be more insightful.

3.3. Cross-lingual Classification

Our final set of replication experiments consists of the cross-lingual model of Vajjala and Rama (2018, Table 4). Table 4 reports our replication results in the same manner as our earlier results for monolingual and multilingual models were presented.

The scores in Table 4 do not present any surprises. The scores are reasonably close to the reported values. Similar to the original findings, the transfer seems to work better for Italian than Czech, and as expected the scores are lower than the corresponding monolingual models.

4. Reproduction: Same Data, Different Code

In this section, we report our first *reproduction* results, where our experiments differ from theirs in the text classification software employed. The software we use is based on our earlier studies (Rama and Çöltekin, 2017; Çöltekin and Rama, 2018; Wu et al., 2019). Here, we only use traditional classifiers used by Vajjala and Rama (2018), namely, *support vector machines* (SVMs), *logistic regression*, and *random forests*. Since we use the same underlying library (Pedregosa et al., 2011), our experiments should arguably not deviate strongly from theirs.

The main difference in our implementation is that we optimise parameters. Although Vajjala and Rama (2018) try a few alternative classification methods, each model's parameters are set to library defaults. For the results presented here, we tune each model using random search. The hyperparamters tuned for all methods are the maximum number of n-grams (0 to 5), the feature weighting algorithm (tf-idf or BM25), and the minimum document frequency for a feature to be included in the model. For the random forest classifier, we tune the number of estimators (300, 400, 500), and for SVM and logistic regression classifiers we tune the (L2) regularization constant (0.01 to 10.0). For each classification experiment, we repeat the classification 2000 times with parameters chosen randomly from the above values. The optimum values found are reported in Table 9 in Appendix A. As in Vajjala and Rama (2018), we report the average of 10-fold cross-validation results for the monolingual and multilingual experiments, and we report a single-best value for cross-lingual experiments.

Tables 5, 6 and 7, report reproduction results corresponding to Tables 2, 3 and 4 in the original paper, respectively. In all cases, we report macro-averaged F1-scores, and the difference of the present result from *our replication results* in Section 3.. The standard deviation reported in Tables 5 and 6 are the deviation of the scores obtained during a single 10-fold cross validation experiment. Note that this is different from the ones reported in section 3., where the standard deviation measures the variability of average

	Italian					Czech				
Features	$F1_w$	$\sigma F1_w$	$\Delta F1_w$	$F1_m$	$\sigma F1_m$	$F1_w$	$\sigma F1_w$	$\Delta F1_w$	$F1_m$	$\sigma F1_m$
base	56.1	2.38	0.8	37.1	2.17	49.3	1.69	0.6	47.2	4.68
pos	74.4	1.14	−1.4	42.0	0.86	69.4	1.60	4.5	67.4	1.69
dep	62.1	0.97	−0.3	35.4	0.66	65.9	1.61	0.6	64.8	1.56
dom	64.0	4.31	1.0	35.4	9.52	48.9	0.58	1.4	37.2	0.43

Table 4: Replication results for cross-lingual model of Vajjala and Rama (2018, Table 4). Besides weighted F1 score ($F1_w$), we report macro-averaged F1 scores ($F1_m$) in all settings. The values reported are averages of scores of 10 runs. The columns with prefix σ indicate the standard deviation of the scores obtained. ΔF_w report the difference between the mean score presented in this table and the value reported in the original paper. All numbers are percentages.

	German			Italian			Czech		
Features	$F1_m$	σ	Δ	$F1_m$	σ	Δ	$F1_m$	σ	Δ
word	54.1	5.49	8.0	68.4	7.18	8.6	75.1	5.11	6.7
pos	56.4	3.76	6.1	70.3	6.68	10.8	70.5	4.63	5.5
dep	52.3	4.22	3.2	65.5	7.33	6.4	70.5	3.86	2.0
dom	56.1	8.40	7.2	73.3	6.18	7.8	67.4	7.17	3.8
word+dom	57.0	6.75	4.3	74.5	9.02	16.4	75.8	5.38	6.5
pos+dom	58.8	4.78	5.3	73.1	10.27	18.1	68.3	9.65	0.7
dep+dom	57.9	7.43	5.5	74.5	9.57	19.6	71.0	5.83	2.0

Table 5: Reproduction of Vajjala and Rama (2018, Table 2) with a different software. We only report macro-averaged F1 score ($F1_m$) The columns with title σ indicate the standard deviation of the score within 10-fold cross validation. Δ columns report the difference between the mean score presented in this table and macro-averaged F1 score reported in Table 2. All numbers are percentages.

Features	$F1_m$	σ	Δ
word	58.1	3.58	15.2
pos	57.5	4.43	9.3
dep	53.6	2.47	6.4
dom	48.5	4.89	9.5

Table 6: Reproduction of Vajjala and Rama (2018, Table 3) with a different software. We only report macro-averaged F1 score ($F1_m$) The columns with title σ indicate the standard deviation of the score within 10-fold cross validation. Δ column reports the difference between the mean score presented in this table and macro-averaged F1 score reported in Table 3. All numbers are percentages.

	Italian		Czech	
Features	$F1_m$	Δ	$F1_m$	Δ
pos	59.0	10.8	63.4	15.2
dep	58.4	11.2	62.9	15.7
dom	64.4	25.4	60.0	21.0

Table 7: Reproduction of Vajjala and Rama (2018, Table 4) with a different software. We only report macro-averaged F1 score ($F1_m$) Δ columns report the difference between the mean score presented in this table and macro-averaged F1 score reported in Table 4. All numbers are percentages.

scores over multiple cross-validation experiments.

In all cases, unsurprisingly, the tuned models yield better scores than the replicated results, varying from almost identical results (monolingual model for Czech with POS and domain features) to around 20 % (cross-lingual models with domain features). Another finding in line with the original study is that the addition of hand-crafted domain features seems to provide a modest, but consistent boost to the monolingual models in most cases.

Nevertheless, tuning the models changes some of the conclusions one may derive from the original study. For example, the ordering of the most powerful features often differs from the ones in Vajjala and Rama (2018) and our replication study. More importantly, unlike the original study, where the numbers indicate substantially better transfer from German to Italian in comparison to Czech, our cross-lingual experiments suggest that transfer to Czech from German is almost in line with transfer from Italian to German.

5. Reproduction: Different Data, Same Code

In our second *reproduction* study, we apply their software on a different language, namely English, and on a different scoring system. To do so, the Cambridge Learner Corpus (CLC) is used. The CLC comprises essays which are answers to some prompts as part of a Cambridge First Exam in 2000 and 2001. Each file consists of two essays written by the same student. We have argued that using the MERLIN corpus for AES may lead to results that show a

	all scores		collapsed to 5		collapsed to 3	
Features	$F1_m$	σ	$F1_m$	σ	$F1_m$	σ
base	7.55	1.49	23.14	2.31	32.44	3.44
word	10.58	<0.01	31.50	<0.01	43.36	<0.01
pos	11.46	<0.01	28.28	<0.10	36.96	0.07
dep	9.74	<0.01	29.65	<0.01	37.35	<0.01
dom	9.66	0.16	28.96	<0.01	35.20	0.17
word+dom1	9.32	0.53	30.89	1.25	37.28	1.78
word+pos	9.32	0.57	28.70	1.12	35.35	0.73
word+dep	9.18	0.40	28.78	0.90	35.24	0.51

Table 8: Reproduction of Vajjala and Rama (2018, Table 3) with a different dataset. We only report macro-averaged F1 score ($F1_m$) The columns with title σ indicate the standard deviation of the scores obtained when running the system 10 times, allowing the random seed to vary. All numbers are percentages.

task-effect rather than a difference in proficiency levels. We shall, therefore, only use the first essay of each student to train and test the software to avoid mixing different text genres. The files are annotated with an overall score (0-40) that the student achieved in the exam as well as scores for each essay individually (0-5.5). Essays that were annotated with a score appearing less than 10 times in the corpus have been removed. The total number of 1223 is comparable to the German part of the MERLIN corpus in Vajjala and Rama (2018) in terms of size. An explanation that was given by the authors of why the classifier predictions were worse for the German texts than for the Italian and Czech was that for German it was a five-class problem, whereas for Italian and Czech only a three-class problem. Thus, in our experiment, three experiment settings are carried out. We first want to see how well the classifiers do on all data points as categorical values. Then, we want to see how it does in a setting similar to the German one, namely collapsing the labels into five classes. Lastly, we conflate the labels into three bins, thus the same number of classes as in the Italian and Czech setting.

1. all scores: all scores as categorical data points
2. scores collapsed to five bins:
   - scores of first essay: <1; 1-2; 2-3; 4-5; 5+
3. scores collapsed to 3 bins:
   - scores of first essay: <3; 3-4.2; 4.2+

The numbers of essays for each group can be found in table 10 in Appendix A.

For the reproduction, we try to follow the preprocessing steps of Vajjala and Rama (2018) as closely as possible. The essays from the CLC come with manual error annotation. For the sake of consistency and comparability, we use the functions provided in Vajjala and Rama (2018) to extract error features. As in Vajjala and Rama (2018), POS tags and Universal Dependencies are extracted with the UDPipe parser (Straka et al., 2016). We, however, do not reproduce all of their experiments. Since the experiments including embeddings do not yield promising results, we only apply the three traditional classifiers. In order to evaluate our models, we do 10-fold cross-validation. Further, we only reproduce their monolingual experiments since the scores in the CLC are not comparable to the CEFR levels in the MERLIN corpus. As in the first reproduction study, we only report macro-averaged F1 scores. The trivial baseline to which the results are compared is again using document length as the only feature.

The best results for the 12-class model are achieved by using POS n-grams only. However, the results are only around $4\,\%$ better than the baseline results. Attaining good results in a 12-class classification task is generally difficult. If one wants to undertake such a fine-grained analysis, treating AES as a regression task might be more suitable. The results for the five-class and three-class classification experiments are better, and for both tasks, the best models use word features only. However, compared to the results of the original paper, the results are still far below the ones achieved for German, Italian and Czech in the original study.

The result of the model that predicts labels collapsed into five classes is $22\,\%$ below the best macro-averaged F1 score for the German data. Similarly, the model that only predicts between three labels achieves results that are $22.14\,\%$ below the Italian and $25.94\,\%$ below the Czech macro-averaged F1 scores.

The reason why the results for English diverge from the ones using the MERLIN corpus may be explained by the nature of the texts of the CLC corpus. Since all texts are written by students taking the Cambridge First exam, the proficiency levels of the authors are less heterogeneous than in the MERLIN corpus, where texts of all CEFR levels are included. Therefore, the models and features may not be able to predict the fine-grained scores of a more homogeneous dataset with regard to the levels of the writers. It is possible that the models do not pick up on the more subtle differences between texts written by language learners whose levels are closer to each other. Note that we do not tune our models as in our first reproduction task. Tuning the models might increase the performance of the models on the English dataset, which is worth pursuing in the future. The fact that the genres of the essays for German, Italian and Czech differ between the CEFR levels might be an additional reason why the model performed much worse on the English data. While the English data is more homo-

geneous regarding the levels and the genres, the essays of the MERLIN corpus are the opposite.

6. General Observations and Discussion

In this paper, we reported our reproduction and replication efforts of a study on automatic essay scoring by Vajjala and Rama (2018), in three different settings: (1) using the same dataset and the same code, (2) using the same dataset with an alternative implementation, and (3) testing the same code on another dataset. This section discusses some interesting and, in our opinion, important points that emerged from the experiments reported above.

Availability of code and data Some of the experiments above were possible because of the fact that both code and data was available. Relatively readable and straightforward code allowed us to re-run their experiments with minor changes to the code (e.g., changing hard-coded paths). The discrepancies we observed with the published results and our replication are difficult to explain, except the changes in the software environment, or potential (unintentional) errors in transferring the values from the software output to the paper. Regardless of the source of the problem, a mismatch between the results from the software and the publications is not desirable, but there is little we can solve with exact replications. On the other hand, code and data availability is also crucial for reproduction. The reproduction experiments we report in Section 4. relies on availability of the data, and experiments we report in Section 5. relies on availability of the code. And, as noted by Wieling et al. (2018), the availability of code and data has further advantages for the community and the authors, such as facilitating comparisons in later studies, and increasing the visibility of the study.

Use of appropriate evaluation metrics We noted that the authors reported F1 scores weighted by the number of positive classes in the gold standard without any clear motivation. This inflates the scores presented, and favours systems that do well on majority classes. Although this is rather an issue with the review process, another benefit of reproduction studies is to catch similar issues missed during peer review.

Reporting variation Overemphasizing replicability of exact values encourages researchers to eliminate some of the natural sources of variation from the output of the released code due to, for example, random initializations or random splits of the data into training and test sets. A method that is commonplace, namely fixing the random seed, was also used in the code released by Vajjala and Rama (2018). In any data-driven experiment, the results are expected to vary with changes in the data. Fixing our software to generate exactly the same numbers may give us a false sense of absolute results. We believe that reporting the variability in such results, and drawing our conclusions with appropriate levels of caution depending on the variability of the results is more important than generating exact replication results. In our experiments, we have shown that paying attention to natural variation in the experiment can prevent making wrong conclusions.

The inevitable bugs Good software engineering practices help reducing bugs, but a zero-bug code is close to impossible for any non-trivial software. And, as reported by Cohen et al. (2018), they may be discovered at any point in the lifetime of a publication. We also discovered a few bugs, notably the reporting of unintended type of score in the publication due to wrong calculation in the software. Note, however, that finding bugs is enabled by the reproduction experiments presented in Section 4.. If we were interested in replication of the sort promoted by Pedersen (2008) and Wieling et al. (2018), we would not have discovered these bugs.

Tuning the machine learning models We obtained quite different results than the originally reported results in Section 4., mainly due to tuning the model parameters.
Although one should pay attention not to 'overtune', there is nothing special about the default hyperparameters coded in the machine learning libraries. In fact, not tuning the hyperparmeters of a machine learning method hides the actual potential of the method, but more crucially other aspects of the systems, such as the feature set. For example, it is likely that a model with a large feature set, and hence more parameters, overfits and performs worse than a simpler model if models were trained with the same settings. However, tuning the regularization parameter may allow the complex model to utilize the signal in the additional features without overfitting.

Choice of data The choice of dataset is crucial for creating reliable systems. Some of the low scores we obtained in Section 5. may be explained by the fact that the English dataset presented us with a different, somewhat more difficult task. Even with the same number of classes, the differences in the CLC corpus is expected to be more fine-grained than the CEFR levels. However, since the original study uses the MERLIN corpus, where topic of the essays correlate with the CEFR levels, it is also likely that part of the success of the original system comes from detecting the topic of the text, rather than the proficiency of the author.

7. Conclusions

We presented replication and reproduction experiments of an automatic essay scoring system for determining proficiency levels of second language learners (Vajjala and Rama, 2018). Although our results mostly agrees with the original report, failures in the present replication effort raises a number of issues, including *the choice of data*, *tuning the machine learning models*, *the use of appropriate evaluation metrics*, *availability of code and data*, *reporting variation*, *the use of appropriate evaluation metrics*, and *availability of code and data* as discussed in Section 6.. Some of these issues are not immediately related to reproduction, but important for the final aim of such an effort: verifying the validity of claims made in a publication.
In closing, we want to reiterate the difference between replication and reproduction. We believe that reproduction, confirming the results under varied settings, is a more useful activity. The overemphasis on exact replication of published values may even have unintended effects, such as encouraging researchers to 'hide' the natural variation that is expected in the task.

8. References

Alikaniotis, D., Yannakoudakis, H., and Rei, M. (2019). Automatic Text Scoring Using Neural Networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 715–725. Association for Computational Linguistics.

Attali, Y. and Burstein, J. (February, 2006). Automated Essay Scoring With e-rater® V.2. *The Journal of Technology, Learning, and Assessment*, 4:3:3–30.

Berggren, S. J., Rama, T., and Øvrelid, L. (2019). Regression or classification? Automated Essay Scoring for Norwegian. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 92–102. Association for Computational Linguistics.

Boyd, A., Hana, J., Nicolas, L., Meurers, D., Wisniewski, K., Abel, A., Schöne, K., Štindlová, B., and Vettori, C. (2014). The MERLIN corpus: Learner language and the CEFR. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1281–1288, Reykjavik, Iceland. European Language Resources Association (ELRA).

Branco, A., Cohen, K. B., Vossen, P., Ide, N., Calzolari, N., et al. (2017). Replicability and reproducibility of research results for human language technology: introducing an LRE special section. *Language Resources and Evaluation*, 51(1):221–247.

Chollet, F. (2015). Keras. https://github.com/fchollet/keras.

Cohen, K. B., Xia, J., Zweigenbaum, P., Callahan, T., Hargraves, O., Goss, F., Ide, N., Névéol, A., Grouin, C., and Hunter, L. E. (2018). Three dimensions of reproducibility in natural language processing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Çöltekin, Ç. and Rama, T. (2018). Tübingen-Oslo at SemEval-2018 task 2: SVMs perform better than RNNs at emoji prediction. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 34—38, New Orleans, LA, United States.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Dikli, S. (2006). An Overview of Automated Scoring of Essays. *The Journal of Technology, Learning, and Assessment*, 5:1:4–36.

Dror, R., Baumer, G., Bogomolov, M., and Reichart, R. (2017). Replicability analysis for natural language processing: Testing significance with multiple datasets. *Transactions of the Association for Computational Linguistics*, 5:471–486.

Drummond, C. (2009). Replicability is not reproducibility: nor is it good science. In *Proceedings of the Evaluation Methods for Machine Learning Workshop at the 26th ICML*, pages 14–18.

Fidler, F. and Wilcox, J. (2018). Reproducibility of scientific results. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2018 edition.

Fokkens, A., van Erp, M., Postma, M., Pedersen, T., Vossen, P., and Freire, N. (2013). Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1691–1701, Sofia, Bulgaria, August. Association for Computational Linguistics.

Foltz, P., Laham, D., and Landauer, T. (1999). The Intelligent Essay Assessor: Applications to Educational Technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 1:2.

Freedman, L. P., Cockburn, I. M., and Simcoe, T. S. (2015). The economics of reproducibility in preclinical research. *PLOS Biology*, 13(6):1–9, 06.

Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science (New York, NY)*, 359(6377):725.

Kitzes, J., Turek, D., and Deniz, F. (2017). *The practice of reproducible research: case studies and lessons from the data-intensive sciences*. Univ of California Press.

Mieskes, M. (2017). A quantitative study of data in the NLP community. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 23–29, Valencia, Spain, April. Association for Computational Linguistics.

Moore, A. and Rayson, P. (2018). Bringing replication and reproduction together with generalisability in NLP: Three reproduction studies for target dependent sentiment analysis. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1132–1144, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Nadeem, F., Nguyen, H., Liu, Y., and Ostendorf, M. (2019). Automated Essay Scoring with Discourse-Aware Neural Models. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 484–493. Association for Computational Linguistics.

Névéol, A., Cohen, K., Grouin, C., and Robert, A. (2016). Replicability of research in biomedical natural language processing: a pilot evaluation for a coding task. In *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*, pages 78–84, Auxtin, TX, November. Association for Computational Linguistics.

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, (349):943–951.

Page, E. B. (1967). Statistical and linguistic strategies in the computer grading of essays. In *COLING 1967 Volume 1: Conference Internationale Sur Le Traitement Automatique Des Langues*.

Pedersen, T. (2008). Empiricism is not a matter of faith. *Computational Linguistics*, 34(3):465–470.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P.,

Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pirina, I. and Çöltekin, Ç. (2018). Identifying depression on Reddit: The effect of training data. In *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task*, pages 9–12, Brussels, Belgium.

Raff, E. (2019). A Step Toward Quantifying Independently Reproducible Machine Learning Research. In *Advances in Neural Information Processing Systems*, pages 5486–5496.

Rama, T. and Çöltekin, Ç. (2017). Fewer features perform well at native language identification task. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 255–260, Copenhagen, Denmark.

Straka, M., Hajic, J., and Straková, J. (2016). UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In *LREC*.

Taghipour, K. and Tou Ng, H. (2016). A Neural Approach to Automated Essay Scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891. Association for Computational Linguistics.

Vajjala, S. and Rama, T. (2018). Experiments with universal CEFR classification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 147–153, New Orleans, Louisiana, June. Association for Computational Linguistics.

Weiß, Z. (2017). Using Measures of Linguistic Complexity to Assess German L2 Proficiency in Learner Corpora under Consideration of Task-Effects. Master Thesis.

Wieling, M., Rawee, J., and van Noord, G. (2018). Reproducibility in computational linguistics: Are we willing to share? *Computational Linguistics*, 44(4):641–649, December.

Wu, N., DeMattos, E., So, K., Chen, P.-z., and Çöltekin, Ç. (2019). Language Discrimination and Transfer Learning for Similar Languages: Experiments with Feature Combinations and Adaptation. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 54–63, TOBEFILLED-Ann Arbor, Michigan.

Yannakoudakis, H., Briscoe, T., and Medlock, B. (2011). A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 180–189. Association for Computational Linguistics.

Zesch, T., Wojatzki, M., and Scholten-Akoun, D. (2015). Task-Independent Features for Automated Essay Grading. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 224—232. Association for Computational Linguistics.

A Appendix

Model		C	classifier	min_df	n_estimators	num_mix	vectorizer	w_ngmax
Monolingual								
word	de	0.87	svm	5	-	-	bm25	4
	it	0.12	svm	5	-	-	bm25	5
	cz	2.74	svm	1	-	-	bm25	1
pos	de	3.42	svm	5	-	-	bm25	1
	it	1.26	svm	5	-	-	bm25	1
	cz	0.04	svm	1	-	-	bm25	4
dep	de	0.15	svm	5	-	-	bm25	5
	it	0.04	svm	5	-	-	bm25	1
	cz	0.02	svm	1	-	-	bm25	4
dom	de	0.21	svm	1	-	-	tfidf	0
	it	3.31	svm	1	-	-	tfidf	0
	cz	0.11	svm	1	-	-	tfidf	0
word+dom	de	6.91	lr	5	-	0.81	tfidf	0
	it	7.33	svm	2	-	1.41	bm25	0
	cz	1.82	svm	1	-	0.11	bm25	1
pos+dom	de	3.51	lr	1	-	1.21	tfidf	1
	it	5.41	lr	1	-	1.91	tfidf	2
	cz	-	rf	1	400	1.91	tfidf	1
dep+dom	de	6.63	lr	1	-	0.81	bm25	0
	it	6.69	svm	2	-	0.51	tfidf	0
	cz	6.18	lr	5	-	0.61	tfidf	2
Multilingual								
word		4.31	lr	5	-	-	bm25	2
pos		1.61	lr	2	-	-	bm25	3
dep		4.29	lr	1	-	-	bm25	2
dom		0.09	svm	1	-	-	bm25	0
Crosslingual								
pos	de	3.42	svm	5	-	-	bm25	1
	it	1.26	svm	5	-	-	bm25	1
	cz	0.04	svm	1	-	-	bm25	4
dep	de	0.15	svm	5	-	-	bm25	5
	it	0.04	svm	5	-	-	bm25	1
	cz	0.02	svm	1	-	-	bm25	4
dom	de	0.21	svm	1	-	-	tfidf	0
	it	3.31	svm	1	-	-	tfidf	0
	cz	0.11	svm	1	-	-	tfidf	0

Table 9: The parameter values used for models reported in Section 4.. We present the best values from the random search which was stopped after 2000 iterations or when the search space was exhausted.

scores	0	2.1	2.2	2.3	3.1	3.2	3.3	4.1	4.2	4.3	5.1	5.2	5.3
number of essays		21	40	184	148	168	174	134	99	111	86	39	19
collapsed to 5	1	2		3			4				5		
number of essays	0	245		490			344				144		
collapsed to 3	1			2			3						
number of essays	245			624			354						

Table 10: This table shows how the scores are grouped into more coarse-grained classes (non-collapsed (all scores), collapsed to five classes, collapsed to three classes). The numbers of essays in each class appear in this table as well.