Regression or classification? Automated Essay Scoring for Norwegian

Stig Johan Berggren
Department of Informatics
University of Oslo
stigjb@gmail.com

Taraka Rama
Department of Informatics
University of Oslo
taraka.kasi@gmail.com

Lilja Øvrelid
Department of Informatics
University of Oslo
liljao@ifi.uio.no

Abstract

In this paper we present first results for the task of Automated Essay Scoring for Norwegian learner language. We analyze a number of properties of this task experimentally and assess (i) the formulation of the task as either regression or classification, (ii) the use of various non-neural and neural machine learning architectures with various types of input representations, and (iii) applying multi-task learning for joint prediction of essay scoring and native language identification. We find that a GRU-based attention model trained in a single-task setting performs best at the AES task.

1 Introduction

Automated essay scoring (AES), in the literature also referred to as Assessment of Proficiency or Automated Text Scoring (ATS), considers the task of assigning a grade to a free form text, often responding to a specific prompt. Automation of this assessment task has clear applications in language education, where second language learners can receive feedback as to which proficiency level they might be on, for instance in relation to the Common European Framework of Reference (CEFR) level. This may help learners who want to take language examination to find the appropriate timing and level of testing, since an examination can be both an economical and logistical inconvenience. Automation also allows students to receive feedback quicker and more frequently.[1]

Recent work on the AES task has used both non-neural, feature-rich approaches that make use of a variety of linguistic features (Briscoe et al., 2010; Yannakoudakis et al., 2011; Vajjala, 2017), and neural end-to-end architectures (Taghipour and Ng, 2016; Alikaniotis et al., 2016). Previous work has furthermore adopted different formulations of the task, either as a regression problem (Phandi et al., 2015; Taghipour and Ng, 2016) or a classification task (Rudner and Liang, 2002; Briscoe et al., 2010; Vajjala and Rama, 2018). Most previous work however, with a few noteworty exceptions (Hancke, 2013; Vajjala and Loo, 2014; Pilán et al., 2016), has been focused on English learner language.

In this paper we present first results for automated essay scoring of Norwegian learner language. We make use of the ASK corpus of learner language (Tenfjord et al., 2006), with added CEFR labels (Carlsen, 2012), and compare and contrast different formulations of the task, a number of different machine learning architectures and different input representations and further experiment with a multi-task setting with Native Language Identification as auxiliary task.

The rest of the paper is structured as follows. We present related work in section 2 and go on to describe the Norwegian learner corpus (ASK) in section 3. We describe the aims of this paper in section 4.1 and describe the data preprocessing and the creation of training, development, and test datasets in section 4.2. We present the results of non-neural linear models in section 5 and CNNs and Gated-RNNs on the development dataset in section 6. Then, we briefly describe the results of our experiments with native language identification in section 7 and the subsequent results of multitask experiments in section 8. Finally, we report the results of the best linear and neural models on the held-out test data in section 9. We summarize and conclude the paper in section 10.

[1] This work was performed when the first author was a Masters student with the Language Technology Group at University of Oslo. Similarly, the second author took part in the BigMed project https://bigmed.no/ hosted at University of Oslo.

2 Related work

Yannakoudakis et al. (2011) present the CLC First Certificate of English (FCE) corpus as well as a system that makes use of deep linguistic features, such as PoS-tags and syntactic information and further employ a discriminative ranker that is shown to outperform a regression approach on the FCE corpus.

Vajjala (2017) trains linear classifiers over the TOEFL11 corpus of non-native English (Blanchard et al., 2013) and the FCE corpus and makes use of a number of linguistic features for the task, including several different measures for lexical diversity, distribution of POS tags, and syntactic complexity, as well as features capturing discourse properties. Several of these features were based on previous work on measuring syntactic complexity in L2 writing by (Lu, 2010).

Alikaniotis et al. (2016) and Taghipour and Ng (2016) both present neural systems trained and evaluated on the ASAP Kaggle dataset of student essays. Both formulate the AES task as a regression task and experiment with several types of neural architectures applied to the same dataset, showing the best results using a bidirectional LSTM with pre-trained embeddings.

Whereas much previous work has been focused on English learner language, there has also been some work on AES using the CEFR scale for languages other than English, viz. Hancke (2013) for German, Vajjala and Loo (2014) for Estonian and Pilán et al. (2016) for Swedish learner texts. All these papers take a very similar approach to the task, modeling essay scoring as SVM classification using a large number of lexical, morphological, syntactic and semantic features. Finally, Vajjala and Rama (2018) present results for mono-lingual, cross-lingual and multi-lingual CEFR classification using the MERLIN corpus (Boyd et al., 2014).

The ASK corpus (Tenfjord et al., 2006) employed in the current study, and further described in section 3, has been used in several linguistic studies on features of Norwegian learner language and transfer effects from different L1 (Pepper, 2012; Golden, 2016; Vigrestad, 2016). The ASK corpus has also been used in previous work to train machine learning systems for Native Language Identification (NLI). The task of NLI for English language learners has been the subject of several shared tasks (Tetreault et al., 2013; Schuller et al., 2016; Malmasi et al., 2017). Norwegian NLI has been studied by Malmasi et al. (2015), using the ASK corpus. In their methodology, they create artificial documents to train on by segmenting the learner texts into sentences, then putting all the sentences from learners with the same L1 into a bag and sampling sentences from the bag to create the new documents. Their rationale for the methodology is that all the resulting documents are of similar length, and that they eliminate the variation between individual writers that otherwise might present a stronger signal than the writer's L1 alone.

In a later study, Malmasi and Dras (2017) perform an NLI experiment on several corpora, namely TOEFL11, the Norwegian ASK corpus and the Jinan Chinese Learner corpus. For Norwegian, they use the features such as function word uni-/bigrams and part-of-speech n-grams. By combining a selection of base classifiers using a LDA meta-classifier trained with bootstrap aggregation (bagging), they achieve an accuracy of 0.818 on the artificially Norwegian essay corpus. The authors' methodology also involves generating artificial essays which discard the discourse properties of a text that could help in improving the system performance at NLI task. Therefore, we do not replicate their experiments but chose to test our best models tuned on development split and then report the best model on a separate test split.

In this paper, we test several RNN models were implemented based on the architecture described in Taghipour and Ng (2016). We made necessary changes to the architectures in order to accommodate our data. For instance, Taghipour and Ng (2016) modelled the task as a regression problem, where the output layer consists of a single node with a value constrained to $(0, 1)$ by the sigmoid function. This layer was replaced with a softmax layer which is described further in section 6.

3 Dataset

The ASK corpus (*AndreSpråksKorpus*; Tenfjord et al., 2006) contains Norwegian learner essays from two different language tests: *Språkprøven i norsk for voksne innvandrere* and *Test i norsk – høyere nivå*,[2] which test proficiency at the B1 and B2 levels, respectively. Following the naming in

[2] Translated to as "Language testing in Norwegian for adult immigrants" and "Test in upper Norwegian levels".

Carlsen (2012), we will refer to these tests as the *IL test* (Intermediate Level, "Språkprøven") and the *AL test* (Advanced Level, "Høyere nivå").

First language	AL test	IL test	Total
English	100	100	200
Polish	100	100	200
Russian	100	100	200
Somali	7	100	107
Spanish	100	100	200
German	100	100	200
Vietnamese	5	100	105
Subtotal (included languages)	512	700	1212
(Albanian)	24	100	124
(Bosnian-Croatian-Serbian)	100	100	200
(Dutch)	100	100	200
(Norwegian nynorsk)	21	11	32
(Norwegian bokmål)	79	89	168
Subtotal (excluded languages)	324	400	724
Total (all languages)	836	1100	1936

Table 1: Distributions of first languages for each test level in ASK. Texts in each test level for all L1. Languages which are not included in our CEFR-labeled dataset are listed in parentheses.

The corpus contains 1736 texts and each document includes metadata such as the writer's L1: one of German, Dutch, English, Spanish, Russian, Polish, Bosnian-Croatian-Serbian, Albanian, Vietnamese, and Somali. All texts from seven of these language backgrounds, 1212 in total, have been assigned a *CEFR* score at a later stage (Carlsen, 2012), and these texts comprise the subcorpus we will be working with. In particular, all texts except those written by learners with Dutch, Bosnian-Croatian-Serbian or Albanian as L1 have a CEFR score. The CEFR labels are available since work by Carlsen (2012), and were not included in the initial release of the corpus. Table 1 shows the number of texts in the corpus for each native language and at each test level.

Restricting the corpus size to only 1212 documents with CEFR scores, the number of tokens amounted to approximately 487,000 in total. Other types of metadata, apart from L1 and CEFR score, include, but are not limited to: the test level the essay was written for, what topic the essay is about, and the learner's country of origin, age, and gender. The CEFR scores in the ASK corpus range between A2 and C1, and also include intermediate labels between the canonical proficiency scores, such as A2/B1 and B1/B2. Thus, the total number of distinct CEFR scores is seven, which is more fine-grained than the TOEFL11 corpus (Blanchard et al., 2013), which only uses three distinct proficiency categories, or the corpus used in Vajjala and Rama (2018), the MERLIN corpus, where the CEFR scores range between A1 and C1, but without any intermediate levels.

When examining the correlation between various types of metadata and proficiency scores in the corpus, there are several noteworthy properties. First of all, we observe that test levels have different distributions of proficiency. We find that the distribution of CEFR scores corresponds to the similarity of the various L1 to Norwegian. The Germanic languages—German and English—have the fewest number of essays below B1 level in the IL test. Two non-Indo-European languages—Vietnamese and Somali—rarely score above B1 level in the IL test, and their mode is A2/B1 compared to B1 for all the Indo-European languages.

4 Methods

4.1 Aims

In this section, we describe the objectives of the experiments reported in the paper. Apart from an extensive analysis of the ASK corpus using linear and neural models, we investigate whether AES based on ASK should be modelled as a classification task or a regression task (given that both the approaches are common in the literature, see section 2). We approach this question by testing three different models namely: Logistic regression, Support Vector Regression, and SVM classification for a wide range of linguistic representation combinations. We then go on to assess the level of performance at the AES task using neural methods? Finally, we combine both the AES task and the NLI task under a single multi-tasking model to check if joint training of a neural model with two different objectives can improve the performance at the AES task.

4.2 Preprocessing

The data files in the ASK corpus are in XML format, and contain information about tags, mistakes and corrections, paragraphs, sentences and more. First, the files were converted to plain text files where all the tags or correction labels were stripped of. The text files have one sentence per line, consisting of space-separated tokens, and an empty line separating paragraphs. These raw text files were then processed using the

UDPipe pipeline (Straka and Straková, 2017) for PoS-tagging and dependency parsing, with pre-trained models trained on the Norwegian UD tree-bank (Øvrelid and Hohle, 2016).

Two different sets of output labels are used in the experiments: The original seven CEFR labels, and a collapsed set where the intermediate classes, such as 'A2/B1', are rounded up to the nearest canonical class, i.e., the CEFR label after the slash. This step yields only four different labels in the *collapsed* set: 'A2', 'B1', 'B2' and 'C1'.

4.3 Reported metrics

We report both the macro and micro F_1-scores for all experiments. The metrics are reported for two different modes: The first utilizing the full set of classes, and the second mode involved training and evaluating on the collapsed classes. A third option, namely to train on the full set of classes and reduce the predictions to the collapsed set of classes, was also attempted, however, in practice the best performers on the collapsed labels turned out to be the models that were also trained on the collapsed tags.

4.4 Data split

Since this paper reports the first results for AES on the ASK corpus, care was taken to create well-defined splits of the data for training, development and final held-out testing. In an ideal scenario, the training and testing datasets would typically have the same distribution of classes, but the limited amount of data makes this difficult; and, as many as 15 combinations of language and proficiency label have fewer than three documents. Moreover, we also took care to create splits in such a manner such that there is no overlap of topics between the splits. The reason for this split is to perform our experiments in a real-world setting where the model needs to be tested on topics not seen in the training data. The final data splits follow a 8:1:1 distribution and we tried to ensure that the joint distribution of proficiency and native language is similar across the splits and that topics do not recur across training, development, and test splits.

5 Linear models

In this section, we train and evaluate the performance of three different linear models on the development dataset. We use the results of this experiment to establish if AES is best modelled as a regression or classification task for the ASK dataset.

Classification vs. Regression As mentioned earlier, AES can be modelled both as a classification task and as a regression task. A disadvantage with using regression for the AES task is that while we know the correct order of classes, it is not obvious if the distance between the adjacent classes is always the same. For instance, we do not know if the distance from CEFR score 'A2/B1' to 'B1' is just as 'B1' to 'B1/B2'. However, we need to be aware of this when we transform the labels into numeric values for regression. This challenge of quantifying distance between classes does not apply to the classification approach, but it does come with another problem. The multi-class approach does not take the order of classes into consideration which is an intrinsic property of the class label in AES tasks.

Implementation All models in this section were implemented using the scikit-learn (Pedregosa et al., 2011) library. The logistic regression model was trained with the 'lgbfs' solver to minimize multinomial loss. The linear regression model was a support vector regressor also chosen from scikit-learn with default parameters. We experiment with different types of input and use both tokens, character ngrams, mixed POS and function word ngrams, and POS ngrams as inputs.

Results In order to report classification based metrics for a regression model, we transformed the predicted scores, which are continuous, into discrete scores equivalent to the given classes. This was done by rounding the raw regression scores to the nearest integer. The output from the support vector regression model is not constrained to any interval, making it necessary to additionally clip the output values to the range of scores: $[0, 6]$ in our full set of labels and $[0, 3]$ in the collapsed set. All the macro and micro F_1 scores for the linear models, for both the full and collapsed sets of classes, are reported in table 2. The Support Vector Regression model is the best performing model at both all labels and collapsed labels tasks. The best model turns to the one trained using UPOS n-grams. The results suggest that treating the AES as a regression problem is a better approach than the classification approach, at least for ASK dataset.

	All labels		Collapsed labels	
Model	Macro F_1	Micro F_1	Macro F_1	Micro F_1
Majority	0.040	0.163	0.127	0.341
LogReg BOW	0.199	0.317	0.384	0.626
LogReg Char	0.221	0.317	0.399	0.602
LogReg POS	0.190	0.301	0.312	0.569
LogReg Mix	0.213	0.341	0.337	0.577
SVC BOW	0.210	0.317	0.391	0.610
SVC Char	0.189	0.293	0.347	0.537
SVC POS	0.157	0.244	0.336	0.618
SVC Mix	0.215	0.350	0.319	0.585
SVR BOW	**0.444**	0.415	0.429	0.659
SVR Char	0.252	0.317	0.440	0.602
SVR POS	0.334	0.358	**0.476**	0.593
SVR Mix	0.312	0.350	0.441	0.659

Table 2: F_1-scores of various classifiers. LogReg is logistic regression, SVC is support vector classification, and SVR is support vector regression.

6 Neural models

In this section, we train and evaluate a wide range of convolutional networks and gated RNNs for the AES task. We further experiment with the use of pre-trained word embeddings which are fine-tuned for the task. The embedding models have been trained on a large Norwegian corpus, the combination of Norsk aviskorpus (The Norwegian Newspaper Corpus) and NoWaC (Norwegian Web As Corpus; Stadsnes, 2018) using the FastText software (Bojanowski et al., 2017) and are available from the NLPL vector repository (Fares et al., 2017).[3]

6.1 Convolutional Networks

We train a number of models which are variants of the convolutional architecture described in Kim (2014). Here, documents are represented as sequences of token IDs and fed into an embedding lookup layer. We pad short documents (length < 700) and use a word frequency cutoff to model out-of-vocabulary words. All models were implemented using Keras (Chollet et al., 2015) and TensorFlow (Abadi et al., 2015) as backend.

The central part of the architecture is a set of convolutional filter banks that are applied to sequences of embeddings. The default architecture from (Kim, 2014) uses 300 convolutional filters: 100 each of window size 3, 4 and 5. After applying the convolutions, the output is max pooled along the time axis. This selects the highest output each filter computed across all windows in the document. In practice, three pooling operations are included in the computational graph, one for each filter bank. This is a technical consideration, necessary because of the different window sizes.

The pooled vectors for each of the filter banks are concatenated into a single vector, representing the document as a whole. This vector has as many elements as there are filters in all the filter banks combined. This representation vector is fed to a final softmax layer to produce a classification output. During training, we apply dropout to the final softmax layer.

	All labels		Collapsed labels	
Model	Macro F_1	Micro F_1	Macro F_1	Micro F_1
Randomly initialized embeddings				
CNN	0.168	**0.398**	0.388	0.732
CNN+POS	0.146	0.374	0.398	0.748
CNN Mix	0.201	**0.398**	0.383	0.724
CNN Reg	0.230	0.382	0.439	0.724
CNN Reg+POS	0.236	0.341	0.383	0.724
CNN Reg Mix	**0.258**	**0.398**	0.412	0.642
CNN Rank	0.177	0.374	0.392	0.740
CNN Rank+POS	0.187	0.382	0.397	0.748
CNN Rank Mix	0.231	0.382	0.379	0.715
Pre-trained, fine tuned embeddings				
CNN	0.208	0.382	0.384	0.724
CNN+POS	0.161	0.366	0.402	**0.756**
CNN Reg	0.242	0.341	**0.463**	0.724
CNN Reg+POS	0.232	0.366	0.411	0.715
CNN Rank	0.198	0.350	0.384	0.724
CNN Rank+POS	0.181	0.325	0.401	**0.756**

Table 3: F_1 scores of CNN classifiers on AES. +POS: Multi-channel input with both words and UPOS tags. Reg: Regression model. Rank: Ordinal regression.

We employ both pre-trained and randomly initialized embeddings and fine-tune the embeddings in the training step. The results of this experiment are given in table 3. In the case of all label prediction, the best results are obtained with a CNN regression model with mixed POS tags as input. The best model in the case of collapsed label prediction is a CNN ordinal rank regression with POS tags as input.

6.2 Recurrent Networks

In this section, we tested a wide range of recurrent models for the AES task by modeling the problem as a regression problem. Taghipour and Ng (2016) test multiple recurrent architectures such as LSTM, GRU, and attention model in their paper. We made some changes to the architectures and added more experiments which are described in the following. The embedding layer in Taghipour and Ng (2016) was of 50 dimensions and randomly initialized. We increased the embedding di-

[3] http://vectors.nlpl.eu/repository/

mensions size to 100 and experimented with randomly initialized embeddings and with pre-trained ones as in the experiments with CNN. Due to the long essay length, we chose to work with gated RNNs since they are known to capture long-distance dependencies. We experimented with the following settings in the case of gated RNNs:

- Long Short Term Memory network (LSTM) vs. Gated Recurrent Unit (GRU)
- Bidirectional vs. unidirectional
- Attention mechanisms: Mean, maximum of the hidden states over all the time steps, and a weighted version of attention involving a feed-forward network (Pappas and Popescu-Belis, 2017).

	All labels		Collapsed labels	
Model	Macro F_1	Micro F_1	Macro F_1	Micro F_1
Random init, unidirectional GRU				
Mean	0.264	0.374	0.455	0.675
Max	0.219	0.325	0.487	0.683
Attn	0.434	0.431	**0.806**	0.805
+POS Mean	0.348	0.398	0.450	0.642
+POS Max	0.230	0.374	0.500	0.748
+POS Attn	0.434	0.423	0.718	0.813
Mix Mean	0.225	0.333	0.388	0.634
Mix Max	0.200	0.398	0.398	0.756
Mix Attn	0.302	**0.455**	0.509	0.780
Random init, BiGRU				
Mean	0.314	0.333	0.444	0.667
Max	0.160	0.325	0.460	0.691
Attn	0.459	0.447	0.805	0.805
+POS Mean	0.373	0.333	0.425	0.683
+POS Max	0.175	0.309	0.503	0.748
+POS Attn	**0.460**	0.447	0.687	**0.821**
Mix Mean	0.231	0.350	0.395	0.642
Mix Max	0.200	0.382	0.405	0.764
Mix Attn	0.275	**0.455**	0.617	0.707
Pre-trained, unidirectional GRU				
Mean	0.274	0.366	0.463	0.715
Max	0.185	0.350	0.401	0.756
Attn	0.414	0.431	0.678	0.797
+POS Mean	0.282	0.382	0.477	0.699
+POS Max	0.193	0.382	0.405	0.764
+POS Attn	0.409	0.423	0.746	0.789
Pre-trained, BiGRU				
Mean	0.266	0.390	0.435	0.707
Max	0.187	0.398	0.393	0.740
Attn	0.454	0.447	0.773	0.797
+POS Mean	0.281	0.382	0.480	0.724
+POS Max	0.183	0.341	0.397	0.748
+POS Attn	0.433	0.439	0.758	0.805

Table 4: F_1 scores of GRU classifiers on AES. BiGRU is birectional GRU; Attn is attention model.

The results of our experiments are given in table 4. Although we experimented with both LSTM and GRUs we found that the GRU architectures performed better than the LSTM architectures at different metrics and label sets with wide variety of settings. Therefore, we report only the results from our GRU classifier. The results show that a combination of embeddings trained separately on words and POS tags give better results at both the full label set and the collapsed label set. Throughout the experiments the document attention model performed the best with randomly initialized embeddings.

7 Native language identification

In contrast to the proficiency labels in ASK where the number of higher level classes are so few, the distribution of L1 labels is more even across the documents. Regression as such is not applicable to NLI since there is no natural ordering among the L1 languages. Therefore, we model the NLI task as a classification problem. We train both CNN, LSTM, and GRU architectures to predict the native language of the writers of the essays. We found that the RNN models performed better than CNN models and among the RNN models, GRU architectures performed better than their LSTM counterparts. Therefore, we report only the results of our best GRU model in table 5. The best model is a GRU model which employs pretrained embeddings and takes mean of the hidden states over the time steps to perform classification using a softmax layer. This model achieves a best accuracy of 0.537 which is lower than the score of 0.542 reported by Malmasi and Dras (2018) on the original essays.

Model	Macro F_1	Micro F_1
Mean	**0.520**	**0.537**
Max	0.401	0.390
Attn	0.447	0.480
+POS Mean	0.467	0.480
+POS Max	0.406	0.431
+POS Attn	0.454	0.463

Table 5: F_1 scores of Pre-trained, BiGRU classifier at the NLI task consisting of 7 classes.

Note however that we cannot compare our results to the previous work for the following reasons. First, our subset of the ASK dataset corresponds to the seven L1 languages which has been assigned CEFR scores, as compared to the full set of ten L1 languages used by Malmasi et al. (2015). Second, Malmasi et al. (2015) used simulated data sets and not the actual raw essays to perform NLI experiments. Third, unlike the previous studies we do not evaluate our results in a cross-validation fashion.

8 Multi-task learning

Until now, we treated AES and NLI as independent tasks and performed experiments on both the tasks separately. We will now attempt to train a joint model to predict both the tasks jointly.

We selected four models for the multi-task experiments—two convolutional and two recurrent neural networks—based on the macro F_1 results on the development set. The top two models (two each for CNN and RNN) had the highest macro F_1 on the full set of labels on the development set. The multi-task model that we use in this paper has two outputs with different loss functions: one for CEFR prediction and the other for NLI task. The loss function for CEFR output is mean squared loss and for NLI is categorical cross-entropy loss. The losses from both the models are linearly weighted with weights summing up to 1. We searched for the best loss weight by searching over the range of $[0, 1]$ where each weight is separated by an interval of 0.1.

Hyperparameter	CNN1	CNN2
Word embeddings	Dynamic	
Embedding size	100	
L_2 constraint	3	
Windows	3,4,5	
Embedding init	Random	Pre-trained
Input representation	Mixed UPOS	Tokens
Hyperparameter	**RNN1**	**RNN2**
Word embeddings	Dynamic	
Embedding size	100	
RNN cell	GRU	
Pooling method	Attention	
Bidirectional	Yes	
Embedding init	Random	Pre-trained
Input representation	Tokens+UPOS	Tokens

Table 6: Descriptions of the CNN and RNN models showing different settings.

The hyperparameters for the four models are summarized in table 6. All our word embeddings are based on the FastText model. We also tested the variability of the models by training and evaluating each model five times with different random seeds to estimate the variance of the results.

Results We show the results of all the models with all the auxiliary weight combinations in figure 1. When the auxiliary task loss weight is zero, the model reduces to the original single task model of CEFR prediction. The RNN models show the highest macro F_1 scores on the full label set. There is some variablity in the performance of the RNN models in accordance with the loss weight. In each panel, we show the F_1 scores for the five models trained with different random seeds. The CNN models do not benefit from including the NLI task as additional task. The five highest macro-F_1 scores are in the range of 0.468 and 0.483 and were achieved using auxiliary loss weights in the range from 0 to 0.5. Although, the highest score was achieved in single-task mode the auxiliary task results are also competitive as shown in the figure. In fact, the variability due to the initial random seed allows us to conclude that the macro-F_1 scores for a multi-task model is almost the same as the single task mode (auxiliary loss weight set to zero).

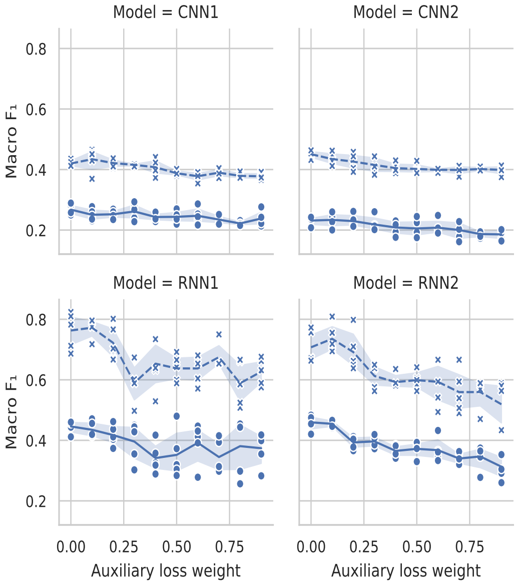


Figure 1: Lines follow the mean of macro F_1 scores. Shaded areas show 95% confidence interval for the mean. Results for the collapsed set of classes are plotted with cross symbols and dashed lines.

The same trend can be observed with RNN2 model also which shows a decreasing trend in the macro-F_1 scores. The macro-F_1 scores' trends for both the RNN models are not similar at the task of collapsed label set classification. The RNN1 model shows a high variation with changes in the auxiliary loss weights. Similar to the full label set classification, the collapsed label set classification does not show worse performance but shows competitive results when the auxiliary loss weight is set to 0.1. We conclude from these experiments that including the NLI task as auxiliary task, at least, does not hurt the performance at AES task.

9 Results

Until now, we evaluated the performance of our models on the development dataset. We now go on to report the performance of our best models from development on the held-out test dataset in table 7. In the case of the neural architectures, we employed the model showing the best macro-F_1 during development.

	All labels		Collapsed labels	
Model	Macro	Micro	Macro	Micro
Majority	0.045	0.187	0.127	0.341
SVR BOW	0.231	0.285	0.420	0.602
SVR POS	0.271	0.350	0.422	0.602
RNN1	0.291	0.439	0.478	**0.724**
RNN2	**0.388**	**0.480**	**0.511**	**0.724**
Multi-RNN1	0.266	0.398	0.509	0.707
Multi-RNN2	0.356	0.447	0.443	**0.724**

Table 7: Results from evaluation on the held-out test set. SVR is support vector regression. Hyperparameters for RNN1 and RNN2 are found in table 6. Multi-task models use an auxiliary task weight of 0.1.

We report the results for SVR, RNNs, and Multi-task RNN models. In terms of micro-F_1 scores, the RNN models are the best across both the collapsed labels and full label sets. Across all the models, both the micro- and macro-F_1 scores are lower than the scores reported on the development split. The Multi-RNN2 model performs the best in terms of micro-F_1 score at collapsed labels. The multi-tasking model shows poor performance in terms of macro-F_1 score across all the tasks when compared to the single task model. The linear models show worse performance than the neural models across all the label sets and evaluation measures. We further observe that a multi-task setup with NLI as auxiliary task does not show competitive results. In conclusion, a fine-tuned embedding BiGRU model augmented with attention and initiated with FastText word embeddings performs the best.

We further examine the confusion matrices for our best multi-task model, 'RNN2 Multi', on the test set. The plot is given in figure 2. We see in the confusion matrix that we can identify a diagonal running from the top-left to the bottom-right, with zeros in the top-right and bottom-left corners. Furthermore, mis-classifications are mostly close to the true value, with no predictions being more than two classes away from the gold label.

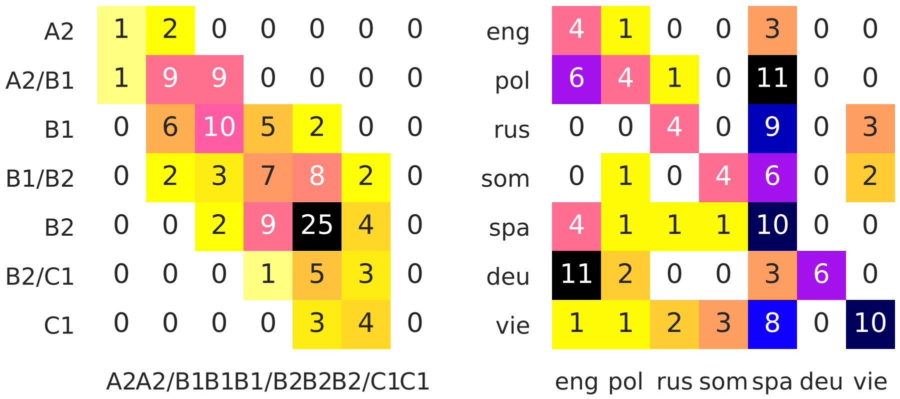
In the confusion matrix for L1, we see that all languages are predicted to be Spanish at some point. German is predicted with 100% precision, however a lot of German texts are wrongly classified as English. This seems reasonable since English and German are similar languages in the same language family. However, the Slavic languages are not confused for each other, as one might expect, but rather Russian and Polish are both commonly mistaken for Spanish.

10 Conclusion

In this paper, we analyzed the ASK corpus for the first time at the AES task with neural and nonlinear models. We addressed the question of modeling AES as regression vs. classification task using three different non-neural models. We find that the AES task is best modeled as regression, at least in the case of the ASK corpus. We tested different input representations such as word, character, and POS n-grams for training the non-neural models. We find that the best results are obtained when using Support Vector Regression algorithm.

In the case of neural models, we tested both convolutional networks and recurrent neural networks (LSTM and GRU) both at AES and NLI tasks. We augmented the neural models with different models of attention such as mean-over-time, max-over-time, and attention learned through a feed-forward network. We find in our experiments that attention augmented BiGRU perform the best at AES task. In contrast, the simpler mean-over-time BiGRU model performed the best at NLI task.

We performed an extensive evaluation of four multitasking models where NLI is an auxiliary task. We tuned the auxiliary loss weight over the development split and found that the weight of 0.1 is best suited for joint modeling of AES and NLI tasks. Although the joint model does not yield results that are better than the single task AES models, we conclude that the joint model yields results that are competitive with the single task model setting. We conclude that multi-task models do not help improve the performance of AES task. We



	A2	A2/B1	B1	B1/B2	B2	B2/C1	C1
A2	1	2	0	0	0	0	0
A2/B1	1	9	9	0	0	0	0
B1	0	6	10	5	2	0	0
B1/B2	0	2	3	7	8	2	0
B2	0	0	2	9	25	4	0
B2/C1	0	0	0	1	5	3	0
C1	0	0	0	0	3	4	0

	eng	pol	rus	som	spa	deu	vie
eng	4	1	0	0	3	0	0
pol	6	4	1	0	11	0	0
rus	0	0	4	0	9	0	3
som	0	1	0	4	6	0	2
spa	4	1	1	1	10	0	0
deu	11	2	0	0	3	6	0
vie	1	1	2	3	8	0	10

Figure 2: Confusion matrices for RNN2 Multi on the held-out test set. AES task on the left, NLI task on the right. The numbers are counts.

also tested if initializing the embeddings with the FastText model improves the AES results. Although the results on development dataset are ambiguous about the choice of pretrained vs. random initialized embeddings, the results on the test set show that the RNN2 model (fine-tuned) works best at AES classification. Therefore, we suggest that any future neural system for AES should use pretrained embeddings to achieve the best results.

As future work, we intend to analyze the errors made by the model and compare them with the errors marked by the human annotators which are available in the ASK corpus. We believe that such an analysis would be the first that would be useful for designing better models and understanding where the neural models make mistakes. Another direction of future work is to use a hierarchical RNN where each sentence within an essay is encoded by a RNN that would be stacked with an additional RNN layer to handle possible loss of signal in extremely long documents.

Acknowledgments

The authors thank the anonymous reviewers for the comments that helped improve the paper. This work is supported by different projects awarded to the Language Technology Group with the University of Oslo.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. *arXiv preprint arXiv:1606.04289*.

Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. Toefl11: A corpus of non-native english. *ETS Research Report Series*, 2013(2):i–15.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schone, Barbora Stindlova, and Chiara Vettori. 2014. The MERLIN corpus: Learner language and the CEFR. In *Proceedings of LREC*.

Ted Briscoe, B. Medlock, and O. Andersen. 2010. Automated assessment of ESOL free text examinations. Technical report, University of Cambridge Computer Laboratory.

Cecilie Carlsen. 2012. Proficiency level—a fuzzy variable in computer learner corpora. *Applied Linguistics*, 33(2):161–183.

François Chollet et al. 2015. Keras. `https://keras.io`.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, 131, pages 271–276, Linköping, Sweden. Linköping University Electronic Press, Linköpings universitet.

Anne Golden. 2016. Ask-korpuset gjør andrespråksforskningen enda morsommere. men hva gjør voksne innlærere med verbet gjøre i ask? *NOA-Norsk som andrespråk*, 30(1-2):77–120.

Julia Hancke. 2013. Automatic prediction of CEFR proficiency levels based on linguistic features of learner language. Master's thesis, University of Tubingen, Germany.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International journal of corpus linguistics*, 15(4):474–496.

Shervin Malmasi and Mark Dras. 2017. Native language identification using stacked generalization. *arXiv preprint arXiv:1703.06541*.

Shervin Malmasi and Mark Dras. 2018. Native language identification with classifier stacking and ensembles. *Computational Linguistics*, 44(3):403–446.

Shervin Malmasi, Mark Dras, and Irina Temnikova. 2015. Norwegian native language identification. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 404–412, Hissar, Bulgaria.

Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano, and Yao Qian. 2017. A report on the 2017 native language identification shared task. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 62–75, Copenhagen, Denmark. Association for Computational Linguistics.

Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, Portorož, Slovenia.

Nikolaos Pappas and Andrei Popescu-Belis. 2017. Multilingual hierarchical attention networks for document classification. *arXiv preprint arXiv:1707.00896*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Steve Pepper. 2012. Lexical transfer in norwegian interlanguage. Master's thesis, Oslo, Norway.

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*, Lisbon, Portugal.

Ildiko Pilán, David Alfter, and Elena Volodina. 2016. Coursebook texts as a helping hand for classifying linguistic complexity in language learners' writings. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity*, Osaka, Japan.

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

Björn W Schuller, Stefan Steidl, Anton Batliner, Julia Hirschberg, Judee K Burgoon, Alice Baird, Aaron C Elkins, Yue Zhang, Eduardo Coutinho, and Keelan Evanini. 2016. The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity & Native Language. In *INTERSPEECH 2016*, pages 2001–2005, San Francisco, CA, USA.

Cathrine Stadsnes. 2018. Evaluating Semantic Vectors for Norwegian. Master's thesis, Oslo, Norway.

Milan Straka and Jana Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

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*, pages 1882–1891, Austin, TX, USA.

Kari Tenfjord, Paul Meurer, and Knut Hofland. 2006. The ASK corpus: A language learner corpus of Norwegian as a second language. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1821–1824, Genoa, Italy.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A report on the first native language identification shared task. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pages 48–57, Atlanta, GA, USA.

Sowmya Vajjala. 2017. Automated assessment of non-native learner essays: Investigating the role of linguistic features. *International Journal of Artificial Intelligence in Education*, pages 1–27.

Sowmya Vajjala and Kaidi Loo. 2014. Automatic CEFR level prediction for Estonian learner text. In *Proceedings of the third workshop on NLP* *for computer-assisted language learning*, Uppsala, Sweden.

Sowmya Vajjala and Taraka Rama. 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, LA, USA. Association for Computational Linguistics.

Tone Vigrestad. 2016. Ortografi i norsk som andrespråk. Master's thesis, Oslo, Norway.

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-Volume 1*, pages 180–189, Portland, OR, USA. Association for Computational Linguistics.

A Supplemental Material

The data and code for these experiments is available here: https://github.com/PhyloStar/NorskASK