Language Proficiency Scoring

Cristina Arhiliuc
arhili01@ads.uni-passau.de
University Of Passau

Yuliia Barabash
baraba04@ads.uni-passau.de
University Of Passau

Jean Pierre Char
Jean.Char@uni-passau.de
University Of Passau

Stepan Gagin
gagin01@ads.uni-passau.de
University Of Passau

ABSTRACT

The number of individuals seeking language certificates is increasing, which provides a load of work for text level assessment. Our article evaluates and extends the results of an approach of Automatic Essay Scoring proposed as a part of the REPROLANG 2020 challenge. We provide a comparison between our results and the ones from the published article and we include a new corpus for the English language for further experiments. Our results are lower than the expected ones using the same approach and it does not scale well with the added English corpus.

1 INTRODUCTION

In the world of globalization and internationalization, the knowledge of multiple languages offers more opportunities. This drives more individuals to learn additional languages which increases the number of language exams taken for exams such as TOEFL and IELTS for the English language, TCF, DELF and DALF for the French language, telc, TestDaF and Goethe-Institut for the German language and many other language tests.

The Common European Framework of Reference (CEFR) offers a generalized scoring system of language proficiency of learners that consists of 6 levels independent on the language: A1, A2, B1, B2, C1 and C2.

Automated Essay Scoring (AES) represents the task of automatically assessing texts written by learners using natural language processing tools. The verification and validation of a new AES approach are part of the REPROLANG 2020 challenge[1] along with many other research topics in the area of natural language processing.

The goal of this article is reproducing the results published in the candidate article [6] and extending their approach with a new corpus. Section 2 gives a short overview on the state-of-the-art research on AES approaches. A short description of the used corpora is presented in section 3 followed by the methodology applied in this paper in section 4. Section 5 shows the results of reproducing their experiments. Furthermore, section 6 completes their cross-lingual experiments. Additionally, the dataset is augmented and experimented with in section 7. Lastly, our research is concluded in section 8.

[1]REPROLANG 2020, https://www.clarin.eu/event/2020/reprolang-2020, last accessed on July 21, 2019

Text Mining Project, 2019, University Of Passau
.

2 STATE OF THE ART

Common approaches to building AES systems are based on a monolingual evaluation[1, 7]. However, new approaches that construct and evaluate models on multiple languages are emerging[6]. The multilingual approaches are our main interest in this paper.

[6] analyses the possibility of building a universal CEFR classifier. For that, 3 different categories of classifications are analyzed:

- Monolingual classification: Training and evaluating classifiers on texts written in the same language.
- Multilingual classification: Training and evaluating classifiers on texts written in multiple languages.
- Cross-lingual classification: Training classifiers on one language and evaluating them on other languages.

Their experiments were conducted on a multilingual corpus called MERLIN[2], especially on 3 languages: German, Czech and Italian. Each text is described with metadata, such as information about author, information about the text and CEFR levels of rating criteria.

The original corpus was transformed into text files, which contain only the texts without any metadata. The names of these files contain information about the language and CEFR level. In order to tag parts of the speech from the corpus texts researchers used UDPipe parser[5] with universal dependencies treebanks[4]. With these tools, parsed files were created in separate directories.

The researchers emphasized several AES specific features to evaluate a text independent on the language[6]:

- Word and POS n-grams, which are common in AES classifiers[7]
- Embeddings of task-specific words and characters trained through a softmax layer. The authors pointed out that their paper is the first to explore character embeddings as a cross-linguistic feature for AES classifiers[6].
- Dependency n-grams where each unigram consists of 3 elements: The dependency relation, the POS tag of the dependent and the POS tag of the head. The authors pointed out that this kind of features were never used in previous work on AES systems[6].
- Linguistic features such as:
  - Document length: The number of words in a text.
  - Lexical richness features: Lexical density, lexical variation and lexical diversity features.
  - Error features: These are obtained by using LanguageTool[2] for spelling and grammar checking. These features were only collected for the German and Italian languages.

[2]LanguageTool, https://languagetool.org/, last accessed on July 21, 2019

Features	DE	IT	CZ	Avg. Dev.
Baseline	0.477 (-0.020)^{RF}	0.573 (-0.005)^{LR}	0.613 (+0.026)^{LR}	0.017
Word n-grams(1)	0.589 **(-0.077)**^{RF}	0.799 (-0.028)^{RF}	0.727 (+0.006)^{RF}	0.037
POS n-grams(2)	**0.658** (-0.005)^{RF}	0.801 (-0.024)^{RF}	0.678(-0.021)RF	0.016
Dep. n-grams(3)	0.637 (-0.026)^{RF}	**0.800** (-0.006)^{RF}	0.706 (+0.002)^{RF}	0.011
Domain features	0.520 (-0.013) ^{LR}	0.654 (+0.001)^{LR}	0.629 **(-0.034)**^{RF}	0.016
(1)+Domain	0.644 (-0.042)^{RF}	0.793**(-0.044)**^{RF}	0.720 (-0.014)^{RF}	0.033
(2)+Domain	0.646 (-0.040)^{RF}	0.796 (-0.020)^{RF}	0.687 (-0.022)^{RF}	0.027
(3)+Domain	0.639 (-0.043)^{RF}	0.784 (-0.022)^{RF}	**0.730** (+0.018)^{RF}	0.027
Word embeddings	0.604 (-0.042)	0.777 (-0.017)	0.609 (-0.016)	0.025
Avg. Dev.	0.034	0.018	0.017	

Table 1: Weighted F1 scores for monolingual Classification compared to the results from [6] (in parenthesis, $value_{reproduced} - value_{original}$).

which are also called domain features.

Logistic regression, random forests, multi-layer perceptrons and SVMs are compared on experiments with non-embedding features. For the embedded features, neural networks models are trained specifically for that task. They use categorical cross-entropy loss and Adadelta algorithm to train the algorithm. For classification with word embeddings, they used a softmax layer.

They considered 2 different categories of features when experimented with classifiers[6]:

- Non-embedding features - used for logistic regression, random forests, multi-layer perceptron and support vector machines implemented using scikit-learn[3].
- Embedding features - neural network models are used implemented using Keras[4] with TensorFlow[5] as backend.

The results of their experiments were measured using a weighted F1 score. The purpose is to compute the weighted average of the F1 score taking class distribution into account[6].

3 DATASETS PRESENTATION

In the original paper[6], the MERLIN dataset[2] was used. It contains 2,286 texts, which were taken from written examinations of acknowledged test institutions. This dataset contains texts in 3 languages: Czech, German and Italian. Every text is overall graded according to CEFR.

For the purposes of preprocessing the data, the text files from levels, where there was less than 10 instances, were removed from the dataset. Furthermore, unlabeled files were also removed. The final version of the corpora consisted of 2267 texts, the distribution of corpora is shown in Table 2.

International Corpus Network of Asian Learners of English (ICNALE)[6] offers freely available text corpora graded according to the CEFR levels. They contain several collections of different kinds of texts and speeches collected from learners of the English language in 10 Asian countries and regions, as well as from native speakers. For the purpose of this project, the ICNALE Written Essays module[3], containing 5600 essays (200-300 words long) about two topics, is used. For the experiments, only 5200 essays are used and 400 were removed due to missing labels.

The distribution of labels in the new dataset is shown in the last column of table 2. The English corpus contains files labeled as A2, B1 and B2 only. The issue of not having texts of all labels is existent also in the MERLIN dataset.

CEFR level	CZ	DE	IT	EN
A1	0	57	29	0
A2	188	306	381	960
B1	165	331	394	3776
B2	81	293	0	464
C1	0	42	0	0
Total	434	1029	804	5200

Table 2: Distribution of labels in corpora

4 METHODOLOGY

The authors of [6] approached the topic of AES systems differently than previous work:

- They use CEFR system to study AES systems.
- They explore the possibility of an Universal AES, given that the CEFR guidelines are not language specific. They call it Universal CEFR classifier.
- They are exploring cross-lingual AES.

The goal of our research is to verify the results published in [6] and experiment with an additional language. Therefore, three tasks are required:

[3] scikit-learn, https://scikit-learn.org/stable/, last accessed on July 21, 2019
[4] Keras, https://keras.io/, last accessed on July 21, 2019
[5] TensorFlow, https://www.tensorflow.org/, last accessed on July 21, 2019
[6] ICNALE: The International Corpus Network of Asian Learners of English, http://language.sakura.ne.jp/icnale/, last accessed on July 21, 2019

(1) The mentioned experiments will be reproduced, monitored and documented and the results will be compared using the provided code[7].
(2) An English corpus from [3] will be added to the dataset, the experiments will be executed again and the results will be reported in this paper.
(3) Experiment with cross-lingual classifiers using inter-family and intra-family languages.

Through all the article, the notations RF, LinSVC and LR are indicating the used classifiers: Random Forest, Linear Support Vector Classifier or Logistic Regression respectively.

4.1 Execution Environment

All our tests were done on a machine with a processor Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHZ 3.80GHZ. The RAM of the machine is 16 GB (15.9 GB usable) and the operating system is a 64-bit Windows 10 Home Edition, x64-based processor. We have no information about the hardware used for the execution of the experiments whose results are reported in [6].

The language used for the experiments mentioned in this paper is Python with the version 3.7. We have no information about the version of Python and the libraries used for the experiments from [6]. The environment file for our execution is put in the appendix, section B, of this article.

5 ORIGINAL PAPER RESULTS ANALYSIS

In this section, we are going to analyse the implementation of the approach proposed in [6] along with its results. For the purpose of reproducing their experiments, we use the environment described in 4.1. The main goal is to check the validity of the results presented in their article and to explore some improvement possibilities.

Through all this section, the average deviation is defined as follows:

$$Average\ Deviation = \frac{\sum |value_{reproduced} - value_{original}|}{n}$$

where n is the number of values in the column or in the row.

We are going to compare their results with ours for monolingual, multilingual and cross-lingual classification with German as training language.

Table 1 presents the results that we obtained during monolingual classification. The results that we get are quite different from the original paper. The biggest difference in terms of worse results we can notice at classification based on word n-grams for German language (-0.077) and the biggest difference in terms of better results is for Czech language (+0.026). The word n-grams based classification seems to have the furthest result from the original paper with an average deviation of 0.037. As in the original results, the n-grams seem to perform better than syntactic features. Although, here, they seem to be better than or have close results with the combination between n-grams and domain. In our experiments, this combination doesn't seem to have a considerable improvement on the results (e.g. German with dependency n-grams) in some situations and impacts them negatively in some other scenarios (e.g. Italian with dependency n-grams).

[7] GitHub repository, https://github.com/nishkalavallabhi/UniversalCEFRScoring, last accessed on July 21, 2019

Features	Lang (-)	Lang (+)	Avg. Dev.
Baseline	0.426 (-0.002)^{LR}	-	0.002
Word n-grams	0.605 **(-0.116)**^{RF}	0.607 **(-0.112)**^{RF}	0.114
POS n-grams	**0.680** (-0.046)^{RF}	**0.680** (-0.044)^{RF}	0.045
Dep. n-grams	0.650 (-0.053)^{RF}	0.652 (-0.041)^{RF}	0.047
Domain features	0.433 (-0.016)^{LR}	0.447 (-0.024)^{LR}	0.020
Word embeddings	0.652 (-0.041)	0.645 (-0.044)	0.042
Avg. Dev.	0.045	0.053	

Table 3: Weighted F1 scores for multilingual classification with models trained on combined datasets compared to the results from [6] (in parenthesis, $value_{reproduced} - value_{original}$).

Table 3 presents the results obtained during multilingual classification. We notice a huge difference between our results and the ones from the paper for word n-grams both with and without language features. We compared their results published in the paper with the ones available in the "Results" directory of the source code and they were identical. We could not find the source of this problem, especially given that the same function for treating word n-grams was used for both multilingual classification and monolingual classification and this function gives close results for Italian and Czech in monolingual classification. Our results concerning the best features for the multilingual classification are the same as in the original paper: POS n-grams, although our F1 score is lower. For all multilingual experiments, our F1 score was lower than in the original paper. The closest results we got are for the baseline and domain features.

Features	Test: IT	Test: CZ	Avg. Dev.
Baseline	0.553 (=)^{LR}	0.48 (=)^{LR}	0.000
POS n-grams	**0.752** (-0.006)^{RF}	**0.679 (+0.030)**^{RF}	0.018
Dep. n-grams	0.60 **(-0.023)**^{RF}	0.66 (-0.012)^{RF}	0.017
Domain features	0.62 (-0.001)^{LR}	0.46 (-0.009)^{RF}	0.005
Avg. Dev.	0.007	0.017	

Table 4: Weighted F1 scores for cross-lingual classification model trained on German compared to the results from [6] (in parenthesis, $value_{reproduced} - value_{original}$).

→Pred	A1	A2	B1	B2	C1
A1	3 (-2)	26 (+2)	0	0	0
A2	9 (=)	330 (+19)	39 (-17)	3 (-2)	0
B1	2 (-1)	89 (+19)	260 (-19)	43 (-1)	0

Table 5: DE-Train:IT-Test setup with POS n-gram features compared to the results from [6] (in parenthesis, $value_{reproduced} - value_{original}$).

→Pred	A1	A2	B1	B2	C1
A2	0	134 (+5)	54 (-3)	0 (-2)	0
B1	0	30 (+7)	98 (-3)	37 (-4)	0
B2	0	2 (-3)	24 (-1)	55 (+4)	0

Table 6: DE-Train:CZ-Test setup with dependency features compared to the results from [6] (in parenthesis, $value_{reproduced} - value_{original}$).

Tables 4, 5, 6 present the results of cross-lingual classification. We can notice in table 4 that the results of our experiments are close to the ones from the original paper. The only notable difference could be seen in tables 5 and 6 for the predicted A2, where on our environment the classifiers seem to predict more A2 for texts with the true labels A1 and B1. Nevertheless, our experiments showed lower misclassfications of A2 as B2 and of B2 as A2.

Overall, our experiments have shown worse results compared to the results in the original paper. We assume that one possible reason for this is that the results published in the original paper were obtained by repeating the experiments multiple times and choosing the best results.

6 CROSS-LINGUAL EXTENSION

Given that the authors of [6] published the results only for cross-lingual classification with German as training language, we decided to extend their experiments for the other languages to check if the results of the system are similar.

Features	Test: DE	Test: CZ
Baseline	**0.711$^{\text{LinSVC}}$**	**0.770$^{\text{LR}}$**
POS n-grams	0.508$^{\text{RF}}$	0.657$^{\text{RF}}$
Dep. n-grams	0.549$^{\text{LinSVC}}$	0.602$^{\text{LinSVC}}$
Domain features	0.706 $^{\text{LinSVC}}$	0.756$^{\text{RF}}$

Table 7: Weighted F1 scores for cross-lingual classification model trained on Italian.

Features	Test: DE	Test: IT
Baseline	**0.528$^{\text{RF}}$**	0.697$^{\text{LR}}$
POS n-grams	0.444$^{\text{LR}}$	0.587$^{\text{RF}}$
Dep. n-grams	0.363$^{\text{LR}}$	0.531$^{\text{RF}}$
Domain features	0.478 $^{\text{LR}}$	**0.796$^{\text{LinSVC}}$**

Table 8: Weighted F1 scores for cross-lingual classification model trained on Czech.

In the paper, the authors explain using only the German texts corpus as training by the fact that it is the only one containing samples of all the labels. In order to make the cross-language validation correct, we made sure that the prediction is made only on the segment of the data that has the same labels as the training language.

We could notice in the tables 7 and 8 that the results of using the other languages for training are quite different than the ones for German as training language. Baseline and domain features seem to perform better in these two cases.

In tables 9 and 10 where the test data had more than two labels, we can notice the same tendency of the classifier to predict more the label A2 for the texts with the true labels A1 and B1.

→Pred	A1	A2	B1
A1	2	55	0
A2	0	227	79
B1	0	47	284

Table 9: IT-Train:DE-Test setup with baseline

→Pred	A2	B1	B2
A2	283	12	3
B1	186	106	39
B2	23	146	124

Table 10: CZ-Train:DE-Test setup with Domain features

7 EXPERIMENTS WITH AUGMENTED DATASET

One of the aims of the paper was to check the validity of the approach proposed in [6] for other languages and to investigate if building classifiers on languages belonging to the same family improves the results for cross-lingual classification. For this, a new corpus for the English language is added to the corpora list on which experiments are executed.

We also had to rename files similar to how it was done in the original project. So after this step, every file name contains label of language and CEFR level. Next, using an English treebank[8] and the same UDPipe tool, we obtained parsed files for our new corpus.

[8]UDPipe model for English, https://github.com/jwijffels/udpipe.models.ud.2.0/tree/master/inst/udpipe-ud-2.0-170801, last accessed on July 21, 2019

Features	English
Baseline	0.333^{LinSVC}
Word n-grams(1)	0.617^{RF}
POS n-grams (2)	0.615^{RF}
Dep. n-grams(3)	0.616^{RF}
Domain features	0.335^{LinSVC}
(1) + domain	$\mathbf{0.629^{RF}}$
(2) + domain	0.620^{RF}
(3) + domain	0.620^{RF}
Word embeddings	0.619

Table 11: Weighted F1 scores for monolingual classification

Table 11 shows results for the monolingual classification based on the English dataset. We can see that performance for features such as Word n-grams, POS n-grams and dependency n-grams is doubled compared to the baseline. Additionally, the variation among the mentioned features is minor because of the size of new dataset that is at least 4 times bigger than old datasets. The baseline performed expectedly bad given that it is based on the lengths of the documents. According to the table 12, for the English corpus, the variation of text lengths for different labels is insignificant and therefore a bad criterion for classification.

CEFR level	CZ	DE	IT	EN
A1	-	32.23	39.86	-
A2	93.68	56.89	69.04	214.28
B1	169.81	112.48	145.61	224.54
B2	205.91	187.96	-	232.92
C1	-	220.95	-	-

Table 12: Average document length per level

Table 13 presents the result of multilingual classification that was extended with the English language. We compared new results with obtained results from section 5 both with and without language features. In parenthesis is indicated the difference between the current value and the one obtained in section 5.

We observed that adding the English corpus had a negative effect on all F1 scores. However, we want to emphasize that dependency n-grams features and POS n-grams features have the smallest average deviation (0.029 and 0.046) given that the pattern of sentence structure is similar for the majority of the European languages. Simultaneously, baseline and domain features have the biggest average deviations (0.118 and 0.098) which could be explained by the fact that the majority of data samples come from English corpus and English text files classification perform poorly for baseline and domain features.

Features	Lang (-)	Lang (+)	Avg. Dev.
Baseline	0.308 $\mathbf{(-0.118)^{LR}}$	-	0.118
Word n-grams	0.563 $(-0.042)^{\text{RF}}$	0.559 $(-0.048)^{\text{RF}}$	0.045
POS n-grams	**0.634** $(-0.046)^{\text{RF}}$	**0.634** $(-0.046)^{\text{RF}}$	0.046
Dep. n-grams	0.623 $(-0.027)^{\text{RF}}$	0.620 $(-0.032)^{\text{RF}}$	0.029
Domain features	0.318 $(-0.115)^{\text{LR}}$	0.365 $\mathbf{(-0.082)^{LR}}$	0.098
Word embeddings	0.571 (-0.081)	0.572 (-0.073)	0.077
Avg. Dev.	0.071	0.056	

Table 13: Weighted F1 scores for multilingual classification with models trained on Italian, Czech, German and English corpora, compared to the ones trained on Italian, Czech and German corpora

Table 14 presents F1 scores for cross-lingual classification model trained on English corpus. We did this to check if a language from the same family would improve performance. The authors of [6] mentioned that word n-grams and word embeddings are not suitable for cross-language classification. Therefore, the considered features are: baseline, domain features, POS n-grams and dependency n-grams. We noticed that results for a language from the same family (tested on German) have lower F1 scores especially for dependency n-grams and domain features compared to inter-family (tested on Italian and Czech).

Features	Test: DE	Test: IT	Test: CZ
Baseline	0.272^{LR}	$\mathbf{0.726^{LR}}$	0.536^{LR}
POS n-grams	0.431^{RF}	$\mathbf{0.821^{RF}}$	0.570^{RF}
Dep. n-grams	0.299^{LinSVC}	$\mathbf{0.580^{LinSVC}}$	0.351^{RF}
Domain features	0.289^{LR}	$\mathbf{0.363^{LR}}$	0.242^{LR}

Table 14: Weighted F1 scores for cross-lingual classification model trained on English

Table 15 shows a confusion matrix based on POS n-grams features. Moreover, the results from multilingual classification have better scores than the ones from cross-lingual classification.

→Pred	A2	B1	B2
A2	226	40	0
B1	112	219	0
B2	4	289	0

Table 15: EN-Train:DE-Test setup with POS n-gram features

However, results of testing on Italian demonstrate the best performance among other languages and POS n-grams show great performance with a F1 score of 0.821. The confusion matricx 16 shows the misclassification only for adjoining levels of proficiency. Additionally, results of testing on Italian also demonstrate better

performance for baseline and POS n-grams in comparison to multilingual classification.

→Pred	A2	B1	B2
A2	328	53	0
B1	85	309	0

Table 16: EN-Train:IT-Test setup with POS n-gram features

Now we consider table 17 consisting of F1 scores for cross-lingual classification tested on English texts. As in the previous description, we did not consider all features but only baseline, POS n-grams, dependency n-grams and domain features. We observed that baseline and domain features have the smallest values (0.075 and 0.107) in training on German. This means that the model is poorly performing. The reason for this is that as mentioned before, the lengths of essays in English and in German vary dramatically according to table 12. Results of training on Italian (Table 17) show great performance as well as results of testing on Italian (Table 14). F1 scores of using dependency n-grams and domain features have better effectiveness (20% and 45%).

Features	Train: DE	Train: IT	Train: CZ
Baseline	0.075^{RF}	$\mathbf{0.707}^{\mathbf{LinSVC}}$	0.400^{RF}
POS n-grams	0.362^{RF}	$\mathbf{0.716}^{\mathbf{RF}}$	0.567^{RF}
Dep. n-grams	0.449^{LR}	$\mathbf{0.718}^{\mathbf{RF}}$	0.619^{RF}
Domain features	0.107^{RF}	$\mathbf{0.708}^{\mathbf{RF}}$	0.614^{RF}

Table 17: Weighted F1 scores for cross-lingual classification model tested on English

Furthermore, we discovered an interesting case that the efficiency of domain features is different for testing on Czech (Table 14) and training on Czech (Table 17). The performance of domain features of training on Czech is almost 3 times better that of testing on Czech.

The confusion matrices 18 and 19 demonstrate that cross-lingual classification on texts with the true label B1 between English and Czech performs poorly: especially for classifiers trained on English that have an accuracy of at most 25%.

→Pred	A2	B1	B2
A2	5	183	0
B1	0	164	1
B2	0	79	2

Table 18: EN-Train:CZ-Test setup with domain features

→Pred	A2	B1	B2
A2	12	935	13
B1	22	3630	124
B2	0	432	32

Table 19: CZ-Train:EN-Test setup with domain features

8 CONCLUSION

Following the execution of the same experiments as the ones presented in [6], our results showed a remarkable difference with the published ones, especially on the multilingual models.

Furthermore, our paper proved that the presented approach doesn't scale well when English is added. Although, this could have been caused by the different properties of the English corpus: text length, lexical diversity and sentence structure.

Our experiments show that for these corpora of English and German there are no better results in intra-family classification, as we would have expected. The results have nevertheless proved a good correlation between English and Italian.

The model could be possibly improved by exploring other relevant features and see the improvement that they could have on the generic model. Certain new features related to the semantic and syntactic analysis of the texts - correctness of the sentence structure and sentence sense score (check if the sentences make sense from a semantic point of view) - could prove advantageous for further approaches.

REFERENCES

[1] Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic Text Scoring Using Neural Networks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2016). https://doi.org/10.18653/v1/p16-1068

[2] Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Stindlová, and Chiara Vettori. 2014. The MERLIN corpus: Learner language and the CEFR.. In *LREC*. 1281–1288.

[3] Shin'ichiro Ishikawa. 2013. The ICNALE and sophisticated contrastive interlanguage analysis of Asian learners of English. *Learner corpus studies in Asia and the world* 1 (2013), 91–118.

[4] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Portorož, Slovenia, 1659–1666. https://www.aclweb.org/anthology/L16-1262

[5] Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Portorož, Slovenia, 4290–4297. https://www.aclweb.org/anthology/L16-1680

[6] Sowmya Vajjala and Taraka Rama. 2018. Experiments with Universal CEFR Classification. *CoRR* abs/1804.06636 (2018). arXiv:1804.06636 http://arxiv.org/abs/1804.06636

[7] 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 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 180–189. http://dl.acm.org/citation.cfm?id=2002472.2002496

A ADDITIONAL TABLES

→Pred	A1	A2	B1
A2	0	122	66
B1	0	14	151

Table 20: IT-Train:CZ-Test setup with Domain features

→Pred	A2	B1	B2
A2	334	36	11
B1	98	164	132

Table 21: CZ-Train:IT-Test setup with Domain features

→Pred	A1	A2	B1	B2	C1
A2	0	174	419	329	38
B1	0	421	1646	1503	206
B2	0	24	203	189	48

Table 22: DE-Train:EN-Test setup with dependency n-gram features

→Pred	A1	A2	B1
A2	0	84	876
B1	0	212	3564

Table 23: IT-Train:EN-Test setup with dependency n-gram features

B PYTHON ENVIRONMENT

```
    name: tmp
channels:
  - anaconda
  - defaults
dependencies:
  - _py-xgboost-mutex=2.0=cpu_0
  - _tflow_select=2.3.0=mkl
  - absl-py=0.7.1=py37_0
  - anaconda-client=1.7.2=py37_0
  - anaconda-navigator=1.9.7=py37_0
  - asn1crypto=0.24.0=py37_0
  - astor=0.7.1=py37_0
  - attrs=19.1.0=py37_1
  - blas=1.0=mkl
  - boto=2.49.0=py37_0
  - boto3=1.9.162=py_0
  - botocore=1.12.163=py_0
  - bz2file=0.98=py37_1
  - ca-certificates=2019.5.15=0
  - certifi=2019.6.16=py37_0
  - cffi=1.12.3=py37h7a1dbc1_0
  - chardet=3.0.4=py37_1
  - clyent=1.2.2=py37_1
  - cryptography=2.7=py37h7a1dbc1_0
  - cycler=0.10.0=py37_0
  - decorator=4.4.0=py37_1
  - docutils=0.14=py37_0
  - freetype=2.9.1=ha9979f8_1
  - gast=0.2.2=py37_0
  - gensim=3.4.0=py37hfa6e2cd_0
  - grpcio=1.16.1=py37h351948d_1
  - h5py=2.8.0=py37hf7173ca_0
  - hdf5=1.8.18=vc14h7a021fe_0
  - icc_rt=2019.0.0=h0cc432a_1
  - icu=58.2=ha66f8fd_1
  - idna=2.8=py37_0
  - intel-openmp=2019.4=245
  - ipython_genutils=0.2.0=py37_0
  - jmespath=0.9.4=py_0
  - joblib=0.13.2=py37_0
  - jpeg=9b=hb83a4c4_2
  - jsonschema=3.0.1=py37_0
  - jupyter_core=4.5.0=py_0
  - keras=2.2.4=0
  - keras-applications=1.0.8=py_0
  - keras-base=2.2.4=py37_0
  - keras-preprocessing=1.1.0=py_1
  - kiwisolver=1.1.0=py37ha925a31_0
  - libmklml=2019.0.3=0
  - libpng=1.6.37=h2a8f88b_0
  - libprotobuf=3.8.0=h7bd577a_0
  - libtiff=4.0.10=hb898794_2
  - libxgboost=0.90=0
  - m2w64-gcc-libgfortran=5.3.0=6
  - m2w64-gcc-libs=5.3.0=7
  - m2w64-gcc-libs-core=5.3.0=7
  - m2w64-gmp=6.1.0=2
  - m2w64-libwinpthread-git=5.0.0.4634.697f757=2
  - markdown=3.1.1=py37_0
  - matplotlib=3.1.0=py37hc8f65d3_0
  - mkl=2019.4=245
  - mkl-service=2.0.2=py37he774522_0
  - mkl_fft=1.0.12=py37h14836fe_0
  - mkl_random=1.0.2=py37h343c172_0
  - mock=3.0.5=py37_0
  - msys2-conda-epoch=20160418=1
  - nbformat=4.4.0=py37_0
  - nltk=3.4.3=py37_0
  - numpy=1.16.4=py37h19fb1c0_0
  - numpy-base=1.16.4=py37hc3f5095_0
  - olefile=0.46=py37_0
  - openssl=1.1.1=he774522_0
  - pandas=0.24.2=py37ha925a31_0
  - pillow=6.0.0=py37hdc69c19_0
  - pip=19.1.1=py37_0
```

```
  - protobuf=3.8.0=py37h33f27b4_0
  - psutil=5.6.3=py37he774522_0
  - py-xgboost=0.90=py37_0
  - pycparser=2.19=py37_0
  - pyopenssl=19.0.0=py37_0
  - pyparsing=2.4.0=py_0
  - pyqt=5.9.2=py37h6538335_2
  - pyrsistent=0.14.11=py37he774522_0
  - pysocks=1.7.0=py37_0
  - python=3.7.3=h8c8aaf0_1
  - python-dateutil=2.8.0=py37_0
  - pytz=2019.1=py_0
  - pyyaml=5.1=py37he774522_0
  - qt=5.9.7=vc14h73c81de_0
  - qtpy=1.8.0=py_0
  - requests=2.22.0=py37_0
  - s3transfer=0.2.0=py37_0
  - scikit-learn=0.21.2=py37h6288b17_0
  - scipy=1.2.1=py37h29ff71c_0
  - setuptools=41.0.1=py37_0
  - sip=4.19.8=py37h6538335_0
  - six=1.12.0=py37_0
  - smart_open=1.8.4=py_0
  - sqlite=3.28.0=he774522_0
  - tensorboard=1.13.1=py37h33f27b4_0
  - tensorflow=1.13.1=mkl_py37h9463c59_0
  - tensorflow-base=1.13.1=mkl_py37hcaf7020_0
  - tensorflow-estimator=1.13.0=py_0
  - termcolor=1.1.0=py37_1
  - tk=8.6.8=hfa6e2cd_0
  - tornado=6.0.3=py37he774522_0
  - traitlets=4.3.2=py37_0
  - urllib3=1.24.2=py37_0
  - vc=14.1=h0510ff6_4
  - vs2015_runtime=14.15.26706=h3a45250_4
  - werkzeug=0.15.4=py_0
  - wheel=0.33.4=py37_0
  - win_inet_pton=1.1.0=py37_0
  - wincertstore=0.2=py37_0
  - xz=5.2.4=h2fa13f4_4
  - yaml=0.1.7=vc14h4cb57cf_1
  - zlib=1.2.11=h62dcd97_3
  - zstd=1.3.7=h508b16e_0
  - pip:
    - language-check==1.1
prefix: ***
```

C EXECUTION WARNINGS

The execution raises 3 warnings of type FutureWarning on multiple occasions:

- Default solver will be changed to 'lbfgs' in 0.22.
- Default multi_class will be changed to 'auto' in 0.22.
- The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.